INFORMATION SYSTEMS
IN MANAGEMENT VI

Ontologies and Data Base Technologies

Scientific editors
Piotr Jałowiecki
Arkadiusz Orłowski

WULS Press  Warsaw 2010
Effectiveness of modern business processes is strongly depended on abilities of the proper information management. The main basis in these implementations are different database technologies, which enables an information storing in form and structures optimized to effective processing and analysis according to end-users needs. Nowadays, traditional data bases methods and techniques are supported by new tools. One of the most popular and most effective in the present days are ontologies originated from artificial intelligence. There are very popular method of knowledge ordering and formalization with using several methods of categorization and hierarchy implementation.

Current volume consists of 12 papers written by 21 authors coming from different institutions. It provides a rich source of ideas, concepts, solutions, and perspectives. We believe that presented results will be useful for all researchers, experts, and business practitioners, including managers themselves, which are dealing with different categories of information management systems. Chapters are ordered alphabetically, according to the surnames of the first-named authors.

Solutions focused on modeling of user preferences and their database model in MELSQL fuzzy queries system are presented (R. Budziński, M. Krakowiak). Software Measurement Framework (SMF) is one of the basic tools for the software project controlling. Possibilities and methods of SMF extensions to cover the organizational perspective are discussed (D. Dymek). As the coal mining sector employs expensive production tools, the effective management of them significantly influences the production efficiency. General assumptions and structure of a management system for such sector are given (S. Głodzik, J. Jasiewicz, R. Magda, T. Woźni). The role of database applications in supporting decision making processes is enormous and still increasing. Data warehouse and Business Intelligence systems can be effective in such a context. The whole process of creating data warehouse and implementing MicroStrategy business intelligence system (starting from the process of extracting, transforming, and loading source data, and ending with a creation of professional reports) is described (K. Karpio, A. Orłowski, M. Zaroda). Examples of ontologies and XML-based standards for information management in agriculture (description of agriculture-related concepts, identification of their structures, presentation of relationships) are provided (W. Karwowski). The number of on-line shops is constantly increasing and the product diversity caused by market needs is observed. An attempt to build an ontology system for e-commerce applications focused on their functionality is presented (T. Ordysiński). A practical approach to the CMMI (Capability Maturity Model Integration) in the context of building faster and more reliable business management applications is described (A. Poniszewska-Marańda, A. Tworzydło).
Possibilities of using ontology for data and information integration in network structures that require sharing of information resources in quickly changing environment and do not allow traditional integrated systems based on central database are explored (D. Put, J. Sztorc). Advantages of using ADOit software, which supports a variety of application scenarios within IT management tasks, are advocated (D. Strzęciwilk, R. Nafkha). Analysis of the most effective XML compression methods is performed and benefits of evaluation methods are illustrated via some relevant examples (J. Swacha). One of important tasks in designing efficient databases is a wise selection of appropriate indexes. Effective usage of XML indexes in processing data stored in relational databases is discussed (M.J. Śmietanański). Presented strategy of XML indexes selection is based on Microsoft SQL Server 2008 software. Importance of ontologies in integration of websites evaluation methods is suggested and some issues regarding the quality assessment are discussed (P. Ziemb, R. Budziński).

Piotr Jakowiecki
Arkadiusz Orłowski
# TABLE OF CONTENTS

**Budziński R., Krakowiak M.**  
Representation of User Preferences in Interactive Query Language MELSQL ................................................. 7

**Dymek D.**  
Software Measurement Framework as an Integration Platform of Business and Engineering Perspectives of Software Projects .......... 18

**Głodzik S., Jasiewicz J., Magda R., Woźni T.**  
Data Bases Concerning Longwall Faces as a Part of Information System Supporting Management of Technical Means of Production ....... 28

**Karpió K., Orłowski A., Zaroda M.**  
Data Warehouse and Business Intelligence Project Based on Microsoft SQL Server 2005 and Microstrategy Applications .................. 36

**Karwowski W.**  
Ontologies and Agricultural Information Management Standards ...... 47

**Ordyniński T.**  
Ontology of E-commerce Application Functionality ................................... 57

**Poniszewska-Marańda A., Tworzydło A.**  
Process of Software Development Using CMMI Model ......................... 67

**Put D., Sztorc J.**  
Co-Shared Ontology as a Core of Information Resources Integrating Systems ......................................................... 78

**Strzęciwilk D., Nafkha R.**  
IT Architecture and Process Management with ADOIT .......................... 89

**Swacha J.**  
The Economic Substantiation of XML Databases Compression ............... 100

**Śmietański M.J.**  
Indexing XML Data and the Performance of XQuery in Relational Database Based on SQL Server 2008 ................................. 109

**Ziembra P., Budziński R.**  
Ontologies in Integration of Websites Evaluation Methods .................. 121
REPRESENTATION OF USER PREFERENCES IN INTERACTIVE QUERY LANGUAGE MELSQL

Ryszard Budziński, Magdalena Krakowiak
Department of Informatics, West Pomeranian University of Technology, Szczecin (ZUT)

Abstract. The paper presents a solution concerning user preferences modeling and their database model in fuzzy query system MELSQL. In the designed system one made the new look on applying of similarity matrix, membership functions and rough sets. The described in this article data model (model SERM) of user preferences is part of designed data model of interactive query language for relational database using fuzzy logic.

Keywords. User preferences, similarity matrix, membership functions, rough sets.

1. INTRODUCTION

Requirements for modern systems of managements grow permanently. According to expectations, main direction of their development is proficient aiding of the decision making.

One of important conditions of the efficient decision support is storing the information about current preferences of decision-maker. In the destination of analysis of their changeability one should store also previous values.

Purpose of article is presentation of solution concerning modeling, storage and using details about preferences of decision-makers.

There is this author's solution being an element of the designed model of the interactive query language using the fuzzy logic MELSQL.

2. QUERY LANGUAGE MELSQL

Designed model of language MELSQL have connection of functionality of fuzzy query language with rules query languages. Main user interface it extended creator of query, of which functionality is presented in Fig. 1. User has capability of work in one of two mode, but namely searching for information or searching for relations between collected data.
Two types of query are available in first mode:

- Simple and composite juxtaposition being based on parameters of projection (from foundation without use of aggregate functions) and selection (building simple or composite predicate)
- Extended statistic reports using with many aggregate functions and ordering.

**Figure 1.** Functional range of query MELSAL. Source: own preparation.
Above queries can of course accept the form of fuzzy query. Simple foundation accepts for requirements of models of languages MELSQL, that fuzzy query has following form:

\[
\text{SELECT} \quad \text{<list of columns/expressions>}
\]

\[
\text{FROM} \quad \text{<list of table>}
\]

\[
\text{WHERE} \quad \text{<fuzzy conditions>}
\] (1)

Containing in oneself the fuzzy predicate is deciding on the fuzzy condition and the fuzzy predicate is characteristic of containing at least one of elements:

- fuzzy value of attributes (linguistic values, fuzzy numbers),
- fuzzy arithmetical operators,
- fuzzy quantifiers.

The analyser of the predicate, constitutes the main block of the algorithm of the analyser of the condition realized as part of query’s analyser in designed model of query language MELSQL. The algorithm of action of the analyser of the predicate was left divided on three, one by one appearing after oneself, corresponding to the part for elements determining his fuzzy character:

- analysis of the attribute value (W) and identification of value of membership function (Xw) for value of attribute
- analysis of the arithmetical operator (O) and identification of value of similarity (Xo) for the arithmetical operator
- analysis of the quantifier (K) and identification of value of membership function (Xk).

The first and third part of the algorithm of the analyser of the predicate is based on exploiting the function of the membership studying the grade of membership of the linguistic value or the quantifier in the determined fuzzy set. However in the second part table of similarity are used for analysis of arithmetical operators.

Algorithm of creation of membership functions and algorithm of creation of similarity table are presented one by one in Fig. 3 and Fig. 4.

In mode of searching for relations user can create one of three type of query [5]:

- Control of essentiality of assigned feature
- Searching for important features
- Creation of rules.

All three above mentioned types of query are realized in the procedure of rules creator using rough sets presented in Fig. 2.
Figure 2. Procedure of rules creator using rough sets. Source: own preparation [5].
MELSQL query serves two kind of information: linguistic value (fuzzy or precise) and numeric value (fuzzy or precise number) using similarity matrix, membership functions and rough sets.

Above mentioned elements of modeling be as realization of query and registration of user preferences, which describe in chapter 3.

3. USER PREFERENCES

The manner of the registration of the preference of the user results from accepted elements of modeling (membership function, similarity matrix and rough sets) what one can see in Fig. 1. Below forms of the registration of the preference of the user are described.

3.1. Membership function

Functions of the membership are an most often used element of fuzzy modeling. They serve determining the degree of the membership (from the scope from 0 to 1) of element for the fuzzy set. On account of minute requirements as for the amount of data polygonal functions of the membership are applicable to a model of the language MELSQL: triangular and trapezium. For creating triangular functions only three values are required (borders of the range of the support of the set and the core), and in case of the trapezium function four values (borders of the range of the support and borders of the range of the core of the set) what the expressed algorithm is showing (Fig. 3).

3.2. Similarity matrix

Tables of similarities are matrices applied in modeling attributes in case of discreet domains. Degree of their mutual similarity is defined for each steam of value from universe of discourse variable consideration give values of degrees of membership.

The way of creating table of similarity dependent from the kind of the space of deliberations of the variable what one can see in Fig. 4. Filling up the board with set number of the value consists in calculating the distance between them (step):

\[
k = \frac{(b - a)}{(i - 1)} \quad \text{for} \quad a \leq X \leq b,
\]

where:

- \(a, b\) - borders of the space
- \(X\) - numerical space of the variable
- \(i\) - number of an elements of a set
- \(k\) - step.
creating the membership function
determining the support of the set
choice of the type of the polygonal membership function
triangular function?
give the range of the values of the core
symmetrical function?
give the core of the set
Generating of equaling of function
Presentation of graphic form
Record?
Registration in database
END

Figure 3. Algorithm of creating the membership function. Source: own preparation.
Figure 4. Algorithm of creating the table of similarity. Source: own preparation.
In case of the numerical space with the fixed difference between next values, it is possible to enumerate the value of similarity from the following formula:

\[ P(X_m, X_n) = 1 - |m - n| / (i - 1), \]  

(3)

where:

- \( X_m, X_n \) - next \((m, n)\) elements of the space \( X \)
- \( i \) - number of an element of a set
- \( m, n \) - number of the element in the space.

### 3.3. Tables of the discretization

The discretization of variables is one of phases of the procedure of rules creator using rough sets (Fig.2). Chosen method and in her result classes of variables are arising characteristic of every user. For every variable at first a table of the discretization is being inspected. In case of her lack depending on the type of the variable the user has to choose from among others:

- manual defining ranges for the constant variable,
- manual grouping for the discreet variable,
- automatic creating equal ranges for the constant variable according to set parameters concerning their amoun,
- automatic creating ranges for the constant variable according to set parameters concerning the number given in individual periods,
- automatic creating ranges for the discreet variable on the basis of the minimal cardinality [5].

Next, received as a result of the discretization, classes of variables (the scope or the content of ranges and names granted them) are being encoded by assigning to them the subsequent number.

**Table 1.** Model boards of the discretization for the constant variable “ height ”.

<table>
<thead>
<tr>
<th>Input data (height in centimetres)</th>
<th>Name of class</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>fewer than 155</td>
<td>Short</td>
<td>1</td>
</tr>
<tr>
<td>&lt;155; 175&gt;</td>
<td>Medium</td>
<td>2</td>
</tr>
<tr>
<td>more than 175</td>
<td>Tall</td>
<td>3</td>
</tr>
</tbody>
</table>

Source: Own preparation.
3.4. Decision rules

Detecting rules is a last part of the procedure presented in Fig. 3 [5]. Base for her realization is found reduct that is set containing only significant attributes (in the extreme case set of all conditional attributes). As a result of the operation of projecting, for which he is a parameter reduct and decision-making attribute, conducted on the secondary signboard a board of rules, for which every record is treated as the single rule will come into existence. One should tidy received rules up according to the decision-making attribute, and then eliminate contradictory rules that is so which have other his value at the same values of conditional attributes. In incurred into this way set of well defined rules the procedure is rejecting rules about the minute support (numbers of records confirming the rule) or for small strength (attitude of the support to all records). Limitary value of these parameters have the default value in the system, but a possibility of placing them by the user exists. Amongst remaining rules having the same value of the decision, similar rules are sought after and aggregate in one.

4. DATA MODEL OF USER PREFERENCES

The data model of the preference of the user constitutes the part of the database structure for the service of query language MELSQL [1]. Every of higher described forms of the registration of the preference of the user has it headline table connected with detailed tables. The headline table is characteristic by an own number, an username, a date of the introduction and an attribute (variable) which concerns. In case of tables of the membership function, table of similarity and table of discretization, detailed tables are connected with table of the value and are storing information about degrees of the membership, degrees of similarity and encoding classes. A table of rules constitutes the exception, where the information about attributes results from connected predicates. Moreover she has two detailed tables apart from the table of predicates. They are storing the information concerning sides of the rule and logical operators which are linking the left hand for her.

Fig. 5 is moving the database structure of the language MELSQL. He is created according to SERM methodology – tables placed according to the hierarchy of inheriting, but relationships are denoted appropriately PK (primary key) or FK (foreign key).
5. CONCLUSION

This paper presents a research which aims is modeling, storage and using details about preferences of decision-maker. Designed model of language MELSQL is supposed to fill the information gap up integrating possibilities and functions of the rules query language with the service of fuzzy query language. The applied solution to modeling the preference of the user definitely is increasing his effectiveness.

REFERENCES


SOFTWARE MEASUREMENT FRAMEWORK AS AN INTEGRATION PLATFORM OF BUSINESS AND ENGINEERING PERSPECTIVES OF SOFTWARE PROJECTS

Dariusz Dymek

Department of Computational Systems, Cracow University of Economics

Abstract. Software Measurement Framework (SMF) is one of the basic tools for the software project controlling. It gives an opportunity for insight view of processes within the project from management point of view. But it only reflects the software engineering perspective. In presented paper we show how to extend the SMF to cover the organizational perspective, where the project is only one of the parts of IT enterprise. This extension is made by linking the software measures with the primary organizational goals.

Keywords. Software Measurement, Software Project Management.

1. INTRODUCTION

One of the main problems of the IT projects is their effectiveness. The common data shows that only about 40-60% of the IT projects ended with satisfactory results. The main reasons of such situation are the lack of understanding of the business goals by project team and the lack of tools which allow linking the business goals with the activities within the project.

Proposed solution is based on the incorporation of the goals measurement principle with the software measurement process commonly used to control the IT projects. The organizational goals create the hierarchical structure which reflects different levels of detail in the goals definitions and their assignment to the organization units. According to the goals measurement principle for each level of this hierarchy the measures of the goals achievement should be specified. On the other side at the level of the IT project there are also specified the measures which are used for evaluation of the project progress and its results properties. Presented approach shows the general idea how to establish the linkage among the goals and project activities based on the measurement principle. This linkage gives the project team better knowledge about business goals of project and in the same time gives the managers better insight into the project progress.

This paper is divided into three main parts. First part describes the general structure of the IT enterprises and points out the place of the IT project in this
structure (section 2). Second part (section 3 and 4) deals with the basic principles of measurement process respectively in cases of the goals and the IT projects. The last part (section 5 and 6) presents the proposed solution and points out its advantages.

2. GENERAL OVERVIEW OF THE IT ENTERPRISE

IT enterprise is one of the kinds of the organizational and economical enterprises. Its distinguishing feature is the role of the information technology in the overall enterprise. Generally, IT enterprise is linked with the development or the implementations of the new software systems in order to achieve the organizational goals. It can be divided into three phases [1]: preparation, implementation and usage.

The preparation phase covers all activities connected with goals definition and their operationalization. It can be also called the strategic phase. The aim of this phase is to answer the questions about how we can achieve the strategic goals and what we need to do. In case of IT enterprises the results of this phase are the specification of the necessary software and basic parameters of the IT project such as resources, schedule and budget. The implementation phase is responsible for software development and implementation, in accordance with previously defined requirements. In the last usage phase, the new software system is used to achieve the organizational goals. The more detail characteristic of every of the distinguished phases can be found in [2,3]. The general schema of the IT enterprises is presented in Fig. 1.

**Figure 1.** The general schema and properties of the IT enterprise.
Source: own preparation.
The results of the usage phase are deeply dependent on the results of the previous two phases. The preparation phase points out the specification of the required software and the way of its use when it is ready. The implementation phase is responsible for development of software in accordance with requirements. The problem is that these two phases deal with different kinds of issues. First of them covers mainly the business aspects of a given organization while the second one has mostly an engineering character. These differences are visible in the staff engaged in each of these phases (Fig.1) and they are also reflected in the used methods and tools (Fig.2).

![Figure 2. Brief schema of the business and engineering aspects in the IT enterprise. Source: own preparation.](image)

The business aspects are much more variable depending on the sector of activity, size of a given organization, and many other factors which influence the organization functioning. As a consequence almost every organization is unique and it uses different methods and tools. On the other side there is the engineering aspect of the IT enterprises. It is based on the software engineering issues and it uses the methods and tools specific to themselves, much less dependent on the application area. The common problem is establishing an integration platform among these two aspects in such a way that main issues would be understandable for the staff representing mentioned aspects. In figure 2 this problem is represented by “thunderbolt” sign.

The principles of such integration platform must be based on fundamental issues of each of these aspects. These issues should be as much independent of the business variable factors and the particular software methodologies as possible. In presented approach these issues are the roles of goal defining for business aspect and the software measurement for the engineering aspect.
3. GOALS IN THE IT ENTERPRISE AND THEIR MEASURABILITY

The aim of each enterprises is to achieve the organization’s goals. These goals are defined on several levels of detail and form the hierarchical structure [4]. At the top of this hierarchy there are strategic goals which in a rather general way express the aims for the whole organization. In the process of the operationalization, these strategic goals are transformed into the operational goals which are much more detailed and set up the aims for the various organizational units. Continuing this process the next, more detail levels of the goals hierarchy are created, established the goals for a group or even a single person. This hierarchy is presented in fig. 3.

![Goals hierarchical decomposition](image)

**Figure 3.** Goals hierarchical decomposition. Source: own preparation

In such hierarchy the goals at the lower level point out the tasks that must be done to achieve the goals at the upper level. In case of the IT enterprise, creation of such hierarchy is one of the tasks of the preparation phase. The goals of the implementation phase are placed at the one of the middle level. These goals, which set the basic IT project parameters such as specification, schedule or budget, are later transformed in the aims for a project team. These aims are also represented in the hierarchy of goals at the lower levels. Properly created hierarchy should have the following properties:

1. The achievement of goal from an upper layer is dependent on the achievement of goals in its hierarchical decomposition.
2. The achievement of goals from the hierarchical decomposition of the given goals means that this goal is also achieved.

The common requirement for organizational goals is their measurability. In such methods as SMART or GQM the goal measurability is one of the core demands [5, 6, 7]. Measurability of goal means that the goal achievement is expressed in the term of one or few previously defined measures. This is the way of the impartiality assurance in the goal achievement valuation. So it can be assumed that each single goal in the goals hierarchy is linked with the set of measures.
Let $G = G(m_1, \ldots, m_n)$ means that goal $G$ is linked with measures $(m_1, \ldots, m_n)$ and let $G = G(G_1, \ldots, G_k)$ means that goals $(G_1, \ldots, G_k)$ are the subgoals of the goal $G$ in the goals hierarchy. Then the properties (1) and (2) of the goals hierarchy can be recorded as follow: $G(m_1, \ldots, m_n) = F(G_1(m_1, \ldots, m_n), \ldots, G_k(m_1, \ldots, m_{nk}))$, where $F$ means some kind of functional dependence among the goals measures from different levels of hierarchy. The general schema of this dependence is shown in figure 4.

![Figure 4. The hierarchy of goals measures. Source: own preparation.](image)

The basic problem in the creation of the goals hierarchy is the right choice of the measures for the goals at all levels and defining the dependence among them. In case of the IT enterprises this general issue can be restricted to the compliance of the enterprise goals and the goals of its phases especially with the IT project goals.

4. BASIC PROBLEMS OF THE SOFTWARE MEASUREMENT

Software measurement as a concept was created to control the IT projects. Various ratios and measures are the basic tools which give the insight view into the IT project progress. Number of this ratios and their variety reflects the need of controlling the IT projects but also the fundamental problems connected with it. Software as a product is very unique. Computer program is fully immaterial and characterized itself by high complexity, strongly dependent on the development staff and used programming tools (languages). As a consequence there is a problem with setting out the commonly accepted and used definition of such basic properties as size or complexity. For instance, there are several measures of the software size. The most common are the number of lines of code and the number of functional points. But the number of lines of code is strongly dependent on the used programming language: in one language the given software procedure can count few lines of code but in the other it can have hundreds or thousands of them. The solution of this problem are the functional points. A function point is a unit of mea-
surement to express the amount of business functionality software provides to a user. But functional points also have some weak sides: they are not easy to measure, have an abstract character and can be difficult to understand for both the business and the engineering staff. The problem with the software size definition causes the next problems with some derived measures like productivity or unit costs.

These problems are commonly known and still open. Withal these measures are commonly used with the consciousness of their weakness because without them the controlling of IT project proceeding and some software properties is very difficult or almost impossible.

Generally from the managerial point of view we need to answer the question “what data are needed to control the IT project?”. With this general question are linked the next two ones: “what can we measure?” and “how can we measure it?”. In case of software measurement the measures are divided into two main categories: the process measures and the product (software) measures. The examples of these measures are presented at table 1.

<table>
<thead>
<tr>
<th>Measurement category</th>
<th>Measurement issue</th>
<th>Example of measures</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Schedule and budget</td>
<td>• Accordance with time schedule and planned budget</td>
</tr>
<tr>
<td></td>
<td>Work progress</td>
<td>• Number of requirements accomplished or in progress (designed, coded, tested, integrated)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Number of software units accomplished or in progress (designed, coded, tested, integrated)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Percentage of used resources</td>
</tr>
<tr>
<td></td>
<td>Work effectiveness</td>
<td>• Costs per single software unit or requirement</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Resources per single software unit or requirement</td>
</tr>
<tr>
<td></td>
<td>Staff</td>
<td>• Staff experience, utility, stability, availability</td>
</tr>
<tr>
<td></td>
<td>Software engineer-</td>
<td>• Accordance with international standards like CMMI, SPICE, ISO9000, etc.</td>
</tr>
<tr>
<td></td>
<td>ing environment</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Software size</td>
<td>• Number of line of code (KLOC)</td>
</tr>
<tr>
<td></td>
<td>Stability of require-</td>
<td>• Number of requirements added, deleted or modified and their proportion to number of all requirements</td>
</tr>
<tr>
<td></td>
<td>ments</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Errors profile</td>
<td>• Error density – number of detected errors per unit (KLOC, FP)</td>
</tr>
<tr>
<td></td>
<td>Software quality</td>
<td>• Average time between failures</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Various kinds of the internal and external measures of the quality attributes (ISO 9126)</td>
</tr>
</tbody>
</table>

Source: own preparation on the basis of [8, 9, 10].
On account of multiplicity of the software measures, for a given IT project we must choose the set of measures customized to the needs and the capacity of this project. Such sets are often standardized for a given organization and are called the Software Measurement Frameworks (SMF). SMF create the base for the software measurements process and are used not only for the project controlling but also for the projects comparison and the process improvement.

Although the SMF are reflecting the organizational needs there are some general rules which must be fulfilled [11,12]. These rules specify the following:

- the areas of measurement: which properties of process and product should be quantified,
- the principles of the measures selection: what properties should have the measures, especially to be consistent with the measure theory,
- the principles of the measurement process: overall rules of data collection, storage, processing and interpretation,
- the method of tailoring the SMF to the given project with respect to measurement issue and users demands.

The SMF are the elementary tool of controlling the project goals expressed in such basic project parameters as specification, time schedule, budget and resources. But within the IT enterprise, the IT project (the implementation phase) is only responsible for development of the software tools necessary to achieve the enterprise goals. So the SMF are sufficient for controlling projects goals but are insufficient from the enterprise goals point of view. In the next section we describe how to establish link among the SMF and goals of enterprise based on the goals measurement principle.

5. LINKAGE AMONG SOFTWARE MEASUREMENT AND ORGANIZATIONAL GOALS

Presented above hierarchical decomposition of the enterprise goals define the scope of the tasks for the organization and its units. In the correctly build hierarchy each goal is ascribed to a given organization unit (or its group) with pointing out the way of its achievement. It is very important that each kind of planned activities should be easy identified in this hierarchy. It means that there is such level in the goals hierarchy on which different kinds of actions are separated as different goals (subgoals). This feature of the goals hierarchy is indispensable to correctly assign the responsibility for the goals achievement [7].

But it gives us the additional advantage. In such hierarchy it is easy to indicate the goals which are connected with software tools. As a consequence the software
system functionality which is derived from the requirements specification can be
divided into the parts linked with the given goals. The specification is only one of
the project parameters but all project parameters are interdependent (see the so
called “project triangle” [13]), so this partition influences the other parameters like
budget, resource and time. Let’s realize that presented process of partition is in
some respects the converse process to the project definition, in which all user re-
quirements are gathered together and based on them the other project parameters
are specified. Additionally this approach is similar to the short iterations policy
used in the agile methodology.

As a result of the specification partitioning the IT project is divided into the
set of interdependent subprojects. This internal project structure has to be reflected
in the Software Measurement Framework. It means that similar structure must be
introduced into the measurement process. This process must be carried out for each
of the subprojects separately. The results for the whole project are the synthesis of
the results for the subprojects. By this we have got the overview of the whole
project with the inside view in each of the subprojects. The general schema of this
approach is presented in the figure 5.

![Diagram](image)

**Figure 5.** The general schema of the software measurement on the project level.
Source: own preparation.
Proposed approach allows establishing the linkage among subprojects and adequate goals at the enterprise goals hierarchy. Establishing such a linkage means that the measures used in the SMF can be used as the goal measures. Previously without this kind of linkage it was impossible because in general case the project parameters are the synthesis of several goals (linked with them requirements) so the SMF on the level of whole project cannot bring the information useful for the evaluation of the achievement of the single goals from the enterprise goals hierarchy.

6. CONCLUSION

Presented idea of the incorporation of the software measurement framework into the measurement schema used for estimation of enterprise goals achievement allows to establish the direct linkage among the activities undertaken in the project and business goals of IT enterprise. From the managerial point of view it gives some advantages:

- There is a possibility of keeping up to date the progress on the IT project in relation to the given business goals. In classical approach there is no such possibility, you can only trace the progress of a whole project and wait for final results.
- Possible delays or deteriorations of project parameters (over budget, requirements modification, etc.) can be assigned to a given single goal. So, the reaction on the enterprise level can be limited only to a part directly linked with this goal.
- Project team is much more aware of the business goals of their activity and not limited only to project parameters. The goals for project team are more readable and detailed than the overall enterprise (organizational) goals by the insight into the goals hierarchy.
- Process of software project measurement can be incorporated into the process of goals measurements giving more detail information about the enterprise progress.
- Implementation of this idea should not be expensive. It affects mainly the organization of measurement process without introducing new tools or deterioration of the development process.

It is worth noting that the presented conception affects the praxis of the IT enterprises in a very small way but it gives the potentially promising results. In the further research author plans to concentrate on the aspects connected with the implementation rules and wants to answer the question: whether the commonly used SMF are sufficient for the goals measurement purpose.
REFERENCES


DATASE discrimin LONGWALL FACES AS A PART OF INFORMATION SYSTEM SUPPORTING MANAGEMENT OF TECHNICAL MEANS OF PRODUCTION

Stanisław Głodzik, Jan Jasiewicz, Roman Magda, Tadeusz Woźny

Faculty of Mining and Geoengineering, AGH University of Science and Technology

Abstract. The coal mining sector employs very expensive technical means of production and their good management determines the efficiency of production. The technical means of production are understood as those assets plant facilities which are necessary for coal production. Coal production takes place in the longwall faces and main facilities include all face equipment and machines necessary for coal winning, handling and transport. Management of coal company’s means of production requires that the basic data be available on technical resources, their deployment and efficiency of their use. The appropriate information system was developed as a result of project sponsored from the research funds in 2007-2010 as a part of research grant No R 09 003 03. The study presents the general outline of the system and especially data bases concerning the longwall faces in an aspect of technical means of production.

Keywords. Hard coal mining, information system, longwalls, technical means of production.

1. INTRODUCTION

Transformation of the economy into free market conditions in 1989 started processes of the pro-economic activities in many different grounds. In case of coal mining industry it changed the enterprises to a high degree. From the point of view of this article it is particularly worth to emphasize three times shorter stope front with simultaneous three times higher production from average coal wall. Such significant increase has been possible thanks to gradual introduction of more reliable and efficient wall machines instead of used and old equipment.

However, high quality devices are very expensive investment that can currently reach 200 000 000 PLN in case of complete wall equipment. High value of the currently used mining equipment, referred to as technical means of production, requires considered management in order to achieve the most rational use of them. In the enterprises including many plants that group a few coal-mines, amount of operating devices, their ranges of application, different locations in many faces and
posts and different periods of use of different machines in different places makes management of this equipment at the enterprise level a complicated and responsible task. Optimized use of this equipment has to be coordinated with realization of mining works performed with certain technical equipment. Plans of such works are to be established and confirmed with many-months advance. One has to remember that some devices does not have to be owned by a company – there are many forms of lease or rental.

Rational management of technical means of production is a continuous process since the mining industry devices move together with location of coal. Incomparably hard (referring to other branches) working conditions and still not foreseeable ground increase possibility of faults or disturb realization of works according to the assumptions what can lead to the necessity of changing established schedules.

Taking above into account, the R&D project included developing IT system that is to facilitate management of technical means of production in a mining company consisting of a few plants and is to be integrated with IT systems used in mining companies that are to perform different functions.

2. DATA REGARDING WALL EXCAVATIONS FOR THE PURPOSE OF PROPOSED SYSTEM

Data necessary for the mentioned system can be partially obtained from other IT systems that are already in use and can send required data in return. It is necessary to enter data systematically and it is recommended to enter even old data. The problem described in this article separately deals with designed excavations, faces in motion and walls that are already not in use since information regarding them play different roles in the system. Data described in this article are included in a few so called „bases” that are parts of proposed system. Their mutual relations and functional description of the system is presented mainly in item [1]. It is also described in item [3] and it can be understood better thanks to Fig. 1 repeated after item [1].

2.1. Used wall faces

Data for walls that are no longer exploited exhibit documentary value and can be used for analysis and selection of the most rational solutions. Such data, excluding elimination of possible errors, should be treated as final and not be changed.

Every wall face that exploited resources and has been finally abandoned has its characteristics including range of necessary data. Sometimes there are cases in which the work conditions change (expectedly or unexpectedly) during exploitation what results in technology change. Thus, such wall should be artificially split in the system into two separate excavations.
Figure 1. Scheme of the data bases for the proposed system for management of the means of production.

Data interesting for us are in a few so called bases. One of them is called PRZODKI (CHARACTERISTIC OF MINING FACES) and contains all identification data starting from mine name and bed number of described longwall panel and id number of a wall itself and its location (giving excavations surrounding the longwall panel).

The base contains also information describing excavation geometry i.e. length of a face head and thickness of used layer (that not necessarily overlaps with thickness of exploited bed) and slope angle along face head and along longwall panel and length of the longwall panel itself. It also includes necessary geological data regarding the bed – its strength parameters and all related mining threats. Similar parameters should characterize the rocks surrounding the bed within given longwall panel. In case of unexpected tectonic disturbances along wall course information about them should be recorded as well.

Information regarding applied technology of wall excavation realization is very important and should include type of housing and used machines (mainly wall conveyor, miner and other devices like coal breaker, generators for supplying wall devices, devices removing methane, air conditioners etc.). Complete set of infor-
information related to applied technology is located in the base called **TWP (TECHNOLOGY OF DRIVING FACES)**, which includes description of all technologies applicable in wall mining (and other faces). Description of technology is to contain information regarding roof protection, removal of the excavated space, possible realization of wall niches and if the roads along walls are to be liquidated behind wall or not (if yes, which of them). Complete set of information regarding used devices (and their applicability) is included in the base called **SP (MEANS OF PRODUCTION)**.

Significant group of information is related to the wall face and its load with working teams (number of employed personnel in production and maintenance shifts) and is connected with obtained results (progress achieved in case of each shift at wall head, amount of winning and coal, efficiency of particular teams and average values for whole longwall. Documentation should also include records regarding and moments of restart. Information regarding malfunction of devices, repairs, parts replacement etc. are to be simultaneously recorded in **KPM (MACHINE WORKSHEET)**.

Information presented above is used by the system during designing of next excavations, mainly for planning works and strategy including mining activity for longer periods of time. Basing on the collected data characterizing production results achieved earlier, the system calculates what equipment can be used in the future excavations to achieve the best results. If the planning department considers different solutions alternatively, the system can be helpful in their evaluation by simulation of the results achievable for different variants of solutions.

### 2.2. Longwall faces during exploitation

In case of the faces that are currently exploited the system allows current control of consistence of progress with accepted plans. In case of significant deviations in reference to the values established in the schedule the system facilitates its modification, which can be necessary from different reasons (e.g. different geological conditions than assumed). In such case the system facilitates correcting appropriate parts of schedule for whole company according to the needs.

Starting exploitation of every longwall is confirmed by separate sheet in which description of work is recorded along time (including periods of possible pauses no matter the causes). Such sheet is located in the base called **KPP (FACE WORKSHEET)**.

Starting the sheet is connected with moving set of input data regarding particular excavation from the **PRZODKI** base, in which it had been places during earlier preparation of mining schedule. The set contains mainly complete data regarding geological conditions for the bed itself and rocks in the roof and floor (it is mentioned in the previous subchapter). Another group of records describes geometry of the whole panel – its width (it can be identified as face length) and length, dimen-
sions of the wall itself including height of mining (not necessarily equal to bed depth, possible changes along head and along depth) and width of working space (according to the used housing). Recorded information allows 3D representation of the excavations with highly precise localisation.

Next group of data relates to the applied working technology and is initially located in the TWP (TECHNOLOGY OF DRIVING FACES) base. This matter has been already described in the previous subchapter. Technology is connected with information regarding working technical equipment, especially miner, wall and sub-wall conveyor, further path for haulage of winning, set of devices supplying wall set and type of mechanized housing (currently Polish mining industry generally does not use individual housing as basic way of roof support). Comprehensive set of information regarding all machines is available in the base SP (MEANS OF PRODUCTION).

All data mentioned above in this chapter are known in the moment of starting works. Next records are entered continuously and depend on the number of teams members working at particular shifts and their results like face advance, amount and quantity of winning, efficiency and work consumption. Values like number of employees at particular wall, face advance and production are entered every day after obtaining production reports. Some values, like amount of coal, efficiency, work consumption total face advance from the beginning of wall and average values for whole excavation, are computed by the system. Possible information about incidental breaks, faults or other extraordinary events, their duration and causes and moments of restart are to be entered as soon as possible after occurrence. Possible faults and repairs of any machine are also recorded in individual machine sheet (called KPM) that records its whole operating history.

As one can see, the system facilitates continuous control of the conformity of production tasks for the particular longwall with assumed production plans what allows performing appropriate measures in case of significant divergences.

2.3. Designed wall faces

Exploitation of a bed caused continuous movement of heads of the exploitation faces hence in the separated wall field the exploited zone becomes larger and space occupied by coal becomes smaller. After some time whole field becomes exploited and it is necessary to move to next longwall panel.

Thus, underground mining plants have to prepare (earlier enough) exploitation faces to which, after equipping it with appropriate equipment, working teams from finished excavation can come.

The whole process is complicated due to quantity and value of equipment and necessity of matching all operations in time and requires careful planning. Preparation of the appropriate schedules for mining works is performed by a team of de-
signers looking for the most rational solution. Then, the schedule is to be accepted and becomes valid – production process should follow its assumptions.

Described software is helpful for the teams of designers since it allows quick overview of realized, in realization and planned mining works, available equipment and its applicability and availability in time. It facilitates preparing schedules for different types of works in future with optimized use of the machines.

At the beginning of designing process of new wall heading one has to make new sheet in base **PRZODKI (CHARACTERISTIC OF MINING FACES)** and enter its proper id number (it is the first record in it). Then, using digital map of bed in which new wall is located, one has to enter all necessary geological data as described above regarding realized walls or walls in realization. Then one has to enter into the sheet records defining geometry of designed excavation and its location in mine 3D space. To avoid repeating – details are as in previously described walls. Using earlier experience one has to choose the most appropriate exploitation technology (system can display recommendations) – information regarding that can be found in the base **TWP (TECHNOLOGY OF DRIVING FACES)**. Then, after entering records including loading the wall with workforce, the system computes (basing on the history) real advance and schedule – designer is to set starting moment. Technology is connected with equipping the wall with appropriate devices – system is to verify if they are available in required time. If certain machine is not available for the selected time, one can consider purchasing or renting it. It can happen that the machine can be available after small change of starting moment – such correction can be accepted or one can consider increasing advance of the wall from which the device could be taken. After adjusting all elements the proposed solution can be discussed and accepted or compared to different variants prepared in the same way. Choosing the best solution deletes other proposals. Accepting the variant by all authorities makes it valid.

As one can see from the above, the system is not only helpful in planning future mining works but also facilitates rational use of technical means of production belonging to the particular company. This matter was the most important reason of creating discussed software.

### 3. CONCLUSION

Proposed IT system can be the most helpful in companies of a few mines (however, it can be helpful in single mine as well). After introducing, the system is to be filled with up-to-date data and records containing information about previous activity (as much as possible) – it relates to all bases of the system.

In order to ensure system functionality it is necessary to provide access to information collected in proper, updated bases. By information one should under-
stand data that can be collected in three groups (that should not be identified as bases) related to:

- devices,
- technology,
- geological data,
- mining works that were realized, are in realization or are planned.

In case of the devices it is necessary to possess complete knowledge regarding equipment treated as technical means of production, including owned and possibly needed. Records have to contain technical parameters, condition, costs of exploitation, level of exploitation and availability.

Similar situation applies to technologies. The base has to contain complete set of all applied and applicable technologies with conditions of their use.

Geological data have been described in the previous chapter - implementation of the discussed system could be connected with elaborating geological documentation of the beds in digital form.

Another group of information includes mining works. Full overview of works that are realized, in realization and planned is necessary. It should include knowledge regarding conditions and time of duration, used technologies and equipment.

Majority of required information has been collected and recorded in different departments of mines so the major problem is to ensure their availability and unification.

In order to avoid performing the same actions twice, the system should be set to maximally use information already collected in mining companies, e.g. in other IT systems adjusted for different purposes. That is why introduction of such system should be preceded by unifying the procedures to ensure that in all plants similar departments enter (send) identically recorded (and understood) information to existing systems.

It has been established that main goal of the discussed system is to provide possibility of complex and transparent operational analysis of technical means of production and their condition at arbitrary moment. Such possibility can be highly useful for the management since it allows controlling and evaluation of usefulness and level of use of the company assets together with needs regarding that matter compared with planned production tasks. The system should also facilitate making decisions regarding the most advantageous forms of acquisition of needed measures.

Applying the discussed system in mines would also significantly facilitate work of people preparing plans of mining works by proposing different possible variants, allowing selection of the most rational solution.

Finally, in later stages of system implementation it should take into account economic side of the problem and measure results of work and applied equipment to analyze them in terms of economy. This field of system has not been finished yet.
Due to specific working conditions in mining industry like incomplete recognition of ground properties, unexpected events and complicated technological process there is a high uncertainty coefficient of achieving planned results and risk regarding realization of plans comparing to other branches. There is a possibility of developing the system in the future to take that aspect into account as well.

Scientific work financed by fund for education in 2007-2010 as Research and Development project no. R09 003 03.

REFERENCES


Abstract. Recently the role and importance of database applications in supporting decision making processes have been still increasing. The modern marketplace requires today’s business to be capable of analyzing itself at every level, on demand. It also requires delivery of relevant and timely information to the right people at the right time. Finally, it requires that businesses have very close and individualized contact with their customers. Data warehouse and Business Intelligence systems can be an answer to this challenge. On one hand, data collected in a warehouse deliver various information, e.g. about market, demand, or selling goods; on the other hand, business intelligence provides sophisticated and user-friendly analysis of those data. This paper shows the whole process of creating data warehouse and implementing MicroStrategy business intelligence system: starting from the process of extracting, transforming, and loading source data to creating professional reports.

Keywords. E-business, data warehouse, business intelligence, ETL.

1. INTRODUCTION

The main goal of this paper is to show the main steps of creating BI solution using data warehouse and business intelligence MicroStrategy platform. The presentation of tools utilized is performed. The benefits of using such systems are emphasized at all levels of development.

Business Intelligence systems facilitate the analysis of volumes of complex data by giving you the ability to view this data from multiple perspectives. A quality business intelligence system gives users access to data at various levels, allows them to ask any question, and rapidly provides them with an accurate and meaningful answer. Modern-day business intelligence systems must also provide a foundation for the proactive delivery of information to users.

The first step of the project is to data warehouse. In order to facilitate the daily data updates in data warehouse, the fully automated ETL process using SQL Server
Integration Services is utilized. Then, the MicroStrategy part of the development takes place. The schema and application objects are being created. That allows to connect the data with business information as well as to allow to work with that information with convenient and intuitive way. The last part of the project is defining reports.

2. MICROSTRATEGY BI ARCHITECTURE

This chapter introduces the basic components of a MicroStrategy business intelligence architecture. It also describes the way they relate to each other and to the rest of the MicroStrategy environment [1]. The following illustration shows a business intelligence architecture.

![MicroStrategy Business Intelligence Architecture](image)

**Figure 1.** MicroStrategy Business Intelligence Architecture. Source: own preparation.

A source system refers to any system or file that captures or holds data of interest. Usually, the most significant sources of data is an Online Transactional Processing System (OLTP). OLTPs are typically databases or mainframes that store transaction processing data. Transactional processing involves the simple recording of transactions, such as sales, inventory, withdrawals, or deposits. [1]

The Extraction, Transformation, and Loading (ETL) process represents all of the steps necessary to transfer data from disparate source systems to an integrated data warehouse. The first step is to extract data from source systems. The second step is to transform the data and prepare it to be loaded into the data warehouse. Transformation procedures may include converting data types and column names, eliminating bad data, correcting typos, filling in incomplete data, and so forth. The third and final step is to load the data into warehouse.
A well-designed and robust data warehouse lies in the heart of the business intelligence system. Data warehouses are generally based on some relational database management system (RDBMS), or a relational database. Relational databases can be queried directly with Structured Query Language (SQL).

While source systems – particularly OLTP systems – are generally designed and optimized for transactional processing, data warehouses are designed and optimized for reporting and analysis. In combination with other tools and applications, the data warehouse can provide the foundation for a robust Online Analytical Processing (OLAP) system. Analytical processing involves manipulating transaction records to calculate sales trends, growth patterns, percent-to-total contributions, trend reporting, profit analysis, and so forth.

The MicroStrategy platform consists of a variety of products that provide a host of your business intelligence environment. The core of the MicroStrategy platform is MicroStrategy Intelligence Server, which consists of engines and server components that are designed to effectively communicate with the data warehouse and retrieve the results that a user requests at a given time. You can even build custom applications to communicate with the engines and server components and take advantage of their functionality. MicroStrategy Intelligence Server contains the following engines and server components:

- SQL Engine and Query Engine – work together to generate and send optimized SQL to database
- Analytical Engine – formats report result sets and adds analytical power to greatly extend capability of the system beyond what your database platform provides
- Server components – responsible for all aspects of report generation, element retrieval, object retrieval, and so forth

![Figure 2](image-url). Metadata database and Data Warehouse in MicroStrategy environment. Source: own preparation.
The metadata is a relational database that contains information to facilitate the transfer of data between the data warehouse and MicroStrategy applications. It stores MicroStrategy object definitions and connection information for the data warehouse and maps MicroStrategy objects to data warehouse structures and data.

3. DATA WAREHOUSE DESIGNMENT PROCES

The Business Intelligence Development Studio (BIDS) is the desktop workstation component used for different purposes, such as: designing, developing, and testing. BIDS provides an user with totally graphical-oriented development environment offering the wide range of menu options, toolboxes, wizards as well as designers and drag-and-drop methods of development. BIDS is a comprehensive development platform that supports collaboration with source code management and version control; provides debugging tools such as breakpoints, variable watch- es, and data viewers [2].

The sample logical model being the first step of development can be divided into four separated dimensions that describe fact data (data being analysed): Product, Geography, Time, and Customer dimensions. These dimensions provide conceptual view for the user and help to understand the data relationships in the future data warehouse.

![Logical Data Model of Sales Project](image)

**Figure 3.** The logical data model of Sales project. Source: own preparation.
Based on the logical model the physical data model is to be created. It will show the physical structure of how the warehouse database stores the information from the logical data model. The physical data model is presented on the following picture.

**Figure 4.** The physical data model of Sales data warehouse. Source: own preparation.

One of the greatest challenges facing business today is that important business information exists in multiple locations and in different formats. With access to so much information, decision makers need one indisputable version of data. Microsoft SQL Server 2005 Integration Services (SSIS) is the tool used to help to implement data integration process applications among the business application system’s files and databases. SSIS is much more than a simple extract, transform, and load (ETL) process. SSIS enables database administrators and application developers to design, implement, and manage complex, high-performance ETL applications. Using SSIS, the user can select data from one or more sources and standardize, join, merge, cleanse, augment, derive, calculate, and perform just about any other function and operation required for his or her data integration applications [3].

The highest-level object within an SSIS application is called SSIS package. A package is a discrete unit of work that is defined for ETL operations or SQL Server Services administration operations or both. It is a collection of SSIS process
control components and their objects that define the operations, process dependencies, and sequence flow of activities and operations required for a data integration application. Package objects include containers, tasks, precedence constraints, variables, data sources, data destinations, SQL Server administration functions, and custom tasks that can be created to address unique requirements for the applications. [2]

4. BUSINESS INTELLIGENCE PROJECT

The project is the configuration object in which all schema and application objects are created. These elements, combined together, provide a flexible reporting environment. A project’s metadata repository is determined by the project source in which the project is created. Its warehouse location is defined by associating it with the appropriate database instance. A project may contain any number of reports as well as objects that make up reports such as attributes, filters, prompts, and metrics. Conceptually, the project is simply the environment in which all related reporting is performed. There are four primary steps involved in the project creation process:

1. Create the metadata shell
2. Establish connections to the metadata repository and the data warehouse
3. Create basic schema objects and project definition using the Project Creation Assistant
4. Configure additional schema-level settings

The metadata does not have to be on the same database platform as the warehouse. When a new project is created, a metadata is automatically populated. The metadata has a predefined, proprietary structure that consists of 10 specific tables for storing the various objects, properties, and settings. The first step in configuring the metadata is to create these 10 blank tables in the metadata database. These blank tables are referred to as metadata shell. MicroStrategy provides SQL scripts for creating the shell, which can be executed using the MicroStrategy Configuration Wizard. The script drops any existing tables of the same name that are in the database, creates the blank tables, and then grants privileges to the tables. MicroStrategy Configuration Wizard that is shown on the illustration below enables to provide connection information for the metadata database, including the DSN, login, and password [4].

The second step in creating new project in MicroStrategy is to establish the connection to the metadata database and the data warehouse. The logical layout of project connectivity is presented on the following illustration.
To connect a metadata database to the Microstrategy project it is necessary to create project source that is the Microstrategy configuration object and contains the connection information about this connection. Project source can connect to the metadata in one of two ways:

- Direct of two-tier mode – points to a specific DSN and database login and password for a metadata database.
- Server of three-tier mode – points to an Intelligence Server definition that, in turn, stores the information of metadata connection.

After creating the metadata and establishing a connection with metadata and warehouse, basic schema objects and project definition can be created - this is the third step in creating new project in MicroStrategy. This task can be completed using the Project Creation Assistant.

![Diagram](image)

**Figure 5.** The logical layout of project connectivity. Source: own preparation.

The Project Creation Assistant is basically a wizard that encompasses several functions. It enables to do the following:

- Name the project and select an associated project source
- Select a database instance and choose the data warehouse tables that are available in the project using the Warehouse Catalog
- Create basic schema objects, such as fact and attribute definitions
Although the Project Creation Assistant goes through many of the primary steps involved in project creation, it does not provide an interface for all of the settings that are necessary to configure as part of project creation. Microstrategy Architect provides additional editors that enable to configure the remaining schema objects. First, MicroStrategy Architect provides a Fact Editor and Attribute Editor that enable to create complex facts and attributes of data being analysed. Second, MicroStrategy Architect provides a Hierarchy Editor that enables to define hierarchies of attributes. Finally, as part of creating a project, there can be created more advanced objects such as transformations, which enable to make data comparisons, and partition mappings, which enable to query physically partitioned tables.

**Schema objects creation**

All of the schema objects together form the project schema. Facts represent measurements of business performance, and they form the basis for metrics, which are used in nearly all analyses and reports. Some examples of facts are: Revenue, Cost, and Profit. Facts relate numeric data values in the data warehouse to the reporting environment. Like other schema objects, such as attributes, facts are logical objects that correspond to warehouse columns and tables. They point to a given location in the data warehouse, and they allow accessing the data stored there.

Attributes are additional pieces of information that provide context for facts. Attribute elements are the possible, unique values of an attribute. Attribute forms are identifiers or descriptors of an attribute. Each attribute form provides details that help to identify and describe an attribute. Every attribute must have at least an ID form (such as Customer_ID or Cust_City_ID), but most attributes have two, the ID form and primary description form. Some attributes even have other descriptive forms in addition to their primary ones. For example, the Customer attribute could have attribute forms, like: ID, Address, e-mail and Customer Name, which usually serves as the primary description form of the attribute.

Directly related attributes can be linked together by defining parent-child relationships. These relationships determine the structure of the system hierarchy – they define how the SQL Engine generates SQL, how tables and columns are joined and used, and which tables are related to other tables. Attribute children and parents can be created using Attribute Editor.

In the Microstrategy environment, hierarchies are groupings of attributes. There are two types of hierarchies:

- **System hierarchy** – reflects the true hierarchies directly related with data modeling, i.e. with relationships determined as parent-child attributes.
- **User Hierarchies** – determine the browsing paths for the attributes and their elements.

The System Hierarchy is created automatically when a project is created. It derives its structure from the specified attributes relationships. It lists all of a
project’s attributes. The System Hierarchy is especially useful for top project designers who wish to examine all of a project’s attributes and their relationships to one another [4].

User Hierarchies provide users with a convenient path for navigating through data in a project. User Hierarchies do not necessarily represent attribute relationships. For example, if most of the users analyze data at the daily level, it might be convenient to create a user hierarchy that allows users to navigate from Year directly to Day, skipping the Quarter, Month, and Week attributes. In addition, it is possible to create user hierarchies that allow users to navigate between attributes that are not related at all.

Application objects creation

There are many different types of objects that can be created in MicroStrategy Desktop. One of the basic application objects is called metric. It performs a calculation on a fact. A metric can employ everything from basic functions, like sum, average, or standard deviation to more advanced functions like moving average, correlation, or n-tiling. For example, when an user wants to see Average Revenue on a report, he or she needs to build a metric that averages the Revenue fact values.

![Figure 6. Schema and application objects. Source: own preparation.](image)

Apart from metrics, other important application objects are filters and prompt. Prompts are used to dynamically modify the contents of a report. With prompt, the user can determine, at run time, the objects he or she wants to retrieve from the report. In addition, the user can make different prompt selections each time the report is running [4].
Reports are the focus of business intelligence analysis. They enable users to gather business insight through data analysis. A report is a requested for specific, formatted data from the data warehouse. It consists of a template plus any desired filtering criteria. A template specifies what information to retrieve from the data warehouse and how this information will be displayed in the report results [37]. A report filter specifies the conditions that the data must meet to be included in report results. A view of report is the actual view of a report subset. The basic report components are displayed on the figure.

5. CONCLUSION

The business intelligence project utilizes several components. The first part of a project is a design and creation data warehouse combining different data sources including external flat files. The ETL process can be designed using SQL Server Integration Services (SSIS) and custom T-SQL scripts in order to enable automated full or incremental loading of data, regarding fact and large lookup tables on the daily basis. The second part of a project is to analyze sales data using the previously defined and prepared data warehouse and MicroStrategy business intelligence reporting system.

The BI solution help to manage a bulk volume of business data and to support the decision making process in a company. Business-intelligence data is what allows a company to grow and exploit future opportunities.
Research granted by Ministry of Science and Higher Education from the funds for science in years 2009-2012 as a scientific project no N N112 049637 “Procesy logistyczne w funkcjonowaniu przedsiębiorstw przetwórstwa rolno-spożywczego”.

REFERENCES


Abstract. Ontologies in the last decade are the object of research in the areas where computer science is applied. An ontology for agriculture should be a description of the concepts, identify their structure, relations between them and also determine the manner in which it must understand the concepts and relationships. The paper presents examples of ontologies and XML-based standards for information management in agriculture. The described standards are also discussed, and forecast of its future development is made.

Keywords. Ontology, XML, knowledge management.

1. INTRODUCTION

Nowadays, almost everywhere data are exchanged, irrespective of whether it is government office, financial institution, manufacturing company or farm. In Poland farmer has to exchange data at least with government institutions, for example there are statistical reports and applications for direct payments. Moreover exchanged data refer to transactions with customers and related financial operations. In agriculture data traditionally was transmitted in paper form. For example farmer sent paper documents to institutions; further data was changed into electronic form in various formats, and transmitted across multiple systems and programs operating in various departments. If these data are transmitted in various formats is becoming burdensome for those who want to use them in another application in a form other than the source. Manual rewriting is acceptable only for small documents. Converting data by employees, even with the help of various tools, is cumbersome, time consuming and, unfortunately, quite often generates errors. In the era of computers and Internet, information sharing is an important issue, information is a source of knowledge. Nowadays even agri-food business acts within the knowledge-based economy. We can say that this is a more complex issue than in other sectors because many aspects and dimensions play important role. Lack of standardization of exchanged data in information systems is real problem which leads to inefficient business processes and slows adoption of new knowledge and technology. In agri-food business information flow is very impor-
tant at whole chain or network level. Right now the exchange of information in sectors connected with agriculture is poorly organized. Generally data can be divided onto: connected with: arable farming and crop production, livestock farming, financial or spatial data. The last one plays an important role in agriculture.

In the rest of our paper we will describe issues connected with exchange data especially the XML language, than we present ontologies and their role in knowledge representation and knowledge management. Further we present the current stage of XML standards for agriculture and ontologies for agriculture. At the end we will compare and evaluate presented standards.

2. DATA EXCHANGE AND XML

Data exchange is part of data management beside storage and retrieval. Historically data in electronic form was stored, retrieved and exchanged as flat files, further file structure became necessary. Development of relational data bases was very important for data management, every year more sophisticated data base management systems made searching and retrieving data easier and more effective. Moreover data warehouses appeared, repositories of a highly organized electronically stored data, designed to facilitate reporting and analysis. Today business intelligence tools give possibility of digging-out and analyzing data. Unfortunately inside such systems, there are required precisely defined data structures (set of tables) which are not always possible to ensure a dynamically changing world. From the other side relational data bases and warehouses did not solve exchange data problem. Special tools for formatting data have to be prepared and many times some work has to be done manually. Data exchange has long history, but common use of computers and especially the Internet has increased its importance. During years standardization was made, for example for trade documents exchange for large organizations without human intervention. There are precisely defined standards of electronic data interchange (EDI), best known is UN/EDIFACT. The EDI standards were designed to be independent of communication and software technologies. EDI documents can be transfer even on diskette, but in practice it was made through dedicated secure networks. Such method of transfer is not cheap and in practice available only for large companies. But nowadays, in the era of the Internet, more companies can exchange electronic data much cheaper through this medium. In the Internet during last years most popular format for data exchange stayed XML (eXtensible Markup Language). XML was developed under the auspices of the World Wide Web Consortium [6]. It is a very flexible text format, readable for both machines and humans similar to the HTML. However, when creating XML documents, users do not use a specific set of tags, like in the HTML, but create their own tags, which can give any name. In this lies the versatility of XML, this format allows for easy storage of any data type. XML is a metalan-
guage, a set of rules that allow a user to create their own language to describe certain classes of documents. We have to note that the XML standard is free to use and is likely to become the data exchange standard for all organizations, irrespective of size. Today XML is a most popular tool used for data exchange. We have to add that common use of XML started development of native XML data bases; also relational data base management systems were extended by XML native storage.

XML-based standard to be understandable to all users must define the correct structure of the document. Such structure is described by schema - kind of grammar. An XML schema is a description of a type of XML document, typically expressed in terms of constraints on the structure and content of documents of that type, above and beyond the basic syntactical constraints imposed by XML itself. Most popular schema is XML Schema defined by WWW Consortium. Summarizing to create an XML dialect, we have to define tags and schema that describes the structure of the document.

3. SEMANTIC INTEROPERABILITY AND ONTOLOGIES

Effective data exchange does not solve all problems. Available data are often incomplete or redundant, in addition come from various sources and terms used for their description are often interpreted differently. In heterogenic, multilingual environment many terms can be understand in different ways and the scopes of many semantic concepts overlap. This is the problem of semantic interoperability. Even in restricted domains, like banking, veterinary, biology, or plant production, it is very difficult to find an agreement on common vocabularies and shared conceptualizations. In such situation different systems use the same word to mean different things, use different words to mean the same thing, use different precision to define the same domain, or describe a domain from different points of view. A reasonable solution to solve semantic disorder is to prepare shared representations of meanings. Today very promise tool to mentioned representations is ontology. Ontologies in the last decade are the object of research in the areas where information technology is applied. It is clear that interest in this subject is large.

According to Wikipedia in computer science and information science, an ontology is a formal representation of knowledge as a set of concepts within a domain, and the relationships between those concepts. It is used to reason about the entities within that domain, and may be used to describe the domain [13]. This philosophical term first time in computer science was used by Gruber in 1993. Original definition was: "An ontology is a formal, explicit specification of a shared conceptualisation" [7], it may be formulated even more simply: “An ontology is a specification of a conceptualization” [8]. According to Gruber ontologies are often equated with taxonomic hierarchies of classes, class definitions, and the subsumption relation, but ontologies need not be limited to these forms. Ontologies are also
not limited to conservative definitions – that is, definitions in the traditional logic sense that only introduce terminology and do not add any knowledge about the world. To specify a conceptualization, one needs to state axioms that do constrain the possible interpretations for the defined terms. In the context of computer and information sciences, an ontology defines a set of representational primitives with which to model a domain of knowledge or discourse. The representational primitives are typically classes (or sets), attributes (or properties), and relationships (or relations among class members). The definitions of the representational primitives include information about their meaning and constraints on their logically consistent application. (see [9]).

Because ontologies are independent from lower level data models, they are used for enabling interoperability among disparate systems, integrating heterogeneous databases, and specifying interfaces to independent, knowledge-based services. Ontologies are used in artificial intelligence, the Semantic Web, systems engineering, software engineering, biomedical informatics, library science, enterprise bookmarking, and information architecture as a form of knowledge representation about the world or some part of it. The creation of domain ontologies is also fundamental to the definition and use of an enterprise architecture framework. To manage information and knowledge we need appropriate knowledge representation. Ontologies stated a very useful tool to represent knowledge in a manner that facilitates drawing conclusions from knowledge. To build effective technologies for knowledge management, knowledge has to be encoded in computer-readable forms. It means that ontologies have to be defined in the form possible to automatically use by computers. Only in such situation we can build tools which can effectively search through files, databases, web sites and other data sources, on the semantic level to extract information and capture its meaning. Many researchers and software developers try to use ontologies for large-scale semantic applications in distributed organizations; and to create open infrastructure for wide cooperation with other applications.

Several ontology representation languages have been developed in the last few years. There are: Common Algebraic Specification Language (CASL) a general-purpose specification language based on first-order logic; Common logic (CL) is a framework for a family of logic languages based on first-order logic, intended to facilitate the exchange and transmission of knowledge in computer-based systems; CycL an ontology language used by Cyc artificial intelligence project; LOOM is a knowledge representation language for knowledge representation and reasoning in Artificial Intelligence; IDEF5 (Integrated Definition for Ontology Description Capture Method) is a software engineering method, with graphical notation, to develop and maintain domain ontologies; Knowledge Interchange Format (KIF) is a computer-oriented language for the interchange of knowledge among disparate computer programs. There are many other ontology languages but
most interested is connected with Semantic Web initiative RDF (Resource Description Framework, see [16]). The Resource Description Framework is a family of World Wide Web Consortium specifications designed as a metadata data model. It has come to be used as a general method for conceptual description or modeling of information that is implemented in web resources, using a variety of syntax formats. There are many possible notations for RDF, N3 or Turtle are very useful but RDF/XML is a syntax which express an RDF as an XML document. W3C defined RDF Schema (RDFS) providing basic elements for the description of ontologies intended to structure RDF resources. We have to mention two close initiatives: DAML and OIL. The DARPA (Defense Advanced Research Projects Agency) Agent Markup Language (DAML) focused on the creation of machine-readable representations for the Web. OIL (Ontology Inference Layer or Ontology Interchange Language) was created as infrastructure for the Semantic Web, and was compatible with RDFS and based on concepts developed in Description Logic (DL) and frame-based systems. DAML+OIL (see [5]) is a successor language to DAML and OIL that combines features of both. In practice it is replaced by Web Ontology Language (OWL). OWL is a family of knowledge representation languages for authoring ontologies. The languages are characterised by formal semantics and RDF/XML-based serializations. OWL, even created for the semantic web, seems to be well positioned to become the standard to represent ontologies during next years.

4. DATA STORING AND EXCHANGING IN AGRICULTURE

In modern agriculture, including Poland, the role of information technology increased considerably. Agricultural and rural areas are now affected by the rapid development of information technologies as it could be. European regulations forced this development. Today not only basic data on production and economic results for statistical purposes are transferred and stored. Computer programs are commonly used for determining economic performance of farms. Moreover many applications support various agricultural activities, such as livestock and crop production. They concern, for example, whether the soil fertilization is balanced or proper composition of animal feed. These programs generally are used locally and do not exchange data with other systems. However, there is need for much larger systems, which cover all Polish farmers. This situation occurs especially after Poland’s accession to the European Union and entry into the Common Agricultural Policy. Spectacular example of IT infrastructure for agriculture, covering the entire country is the IACS system for direct payments. The data collected in the system represent a potentially huge source of knowledge about Polish agriculture. Unfortunately, currently data filled in paper documents by farmers (or advisors), are manually entered into the system by clerks. On the other hand the use of the col-
lected data more widely in other systems is impossible in practice because system does not export aggregated data. The lack of existing Internet based solutions for agriculture and rural areas is one of the factors of insufficient usage of information technology. As huge amount of data is used in modern agriculture, the absence of such solutions becomes a serious problem. Sharing knowledge across borders can benefit farms and whole regions in many important ways. To properly exchange information within the agricultural sector and rural regions, suitable standards for data and information exchange should be selected. For such selection analysis of existing standards, methods and their use is necessary. It helps in definition of new extensible data formats for agribusiness.

There are currently no generally accepted XML standard for agriculture, both crop production and livestock. It seems that the most important initiative undertaken in Europe is developed in Germany agroXML. This initiative was originally implemented only in German language, but since version 1.3 it was decided to switch to English. Developers would like to extend a standard so that it could be used in information systems for agriculture, not only in Germany. Some impact on the agroXML had a Geography Markup Language (GML) developed by the Open Geospatial Consortium (OGC). AgroXML schemas are universal, intended to be conceived as a common denominator for various activities relating to agriculture. They are useful for integrating applications used by a farmer with central systems (government, local governments and various other organizations), agricultural advisory systems, suppliers and customers information systems, service providers, trading systems (including electronic), and finally with the systems for monitoring food quality. The creators have taken the regulation 178 of the European Union in 2002. AgroXML is also connected with the ISO11783 standard protocol ISOBUS. This is a standard for data transmission between a tractor, tools (such as planters, etc.) and farm management software installed on personal computer.

Another interesting project is eDAPLOS developed by UN/CEFACT Europe - Trade and Business Group. The project is based on previously developed EDI messages for transactions between customers and farmer - DAPLOS (Data Plot Sheet – message UN/CEFACT-D05B). They developed ebXML messages, currently they continue the development of messages for “Cattle registration and movement data exchange and animal passport.” It seems that eDAPLOS and related projects will be important in e-commerce, because ebXML is an open standard available to everyone. Next interesting initiative is the SBCS (Scottish Beef Calf Scheme XML) prepared by the Scottish Government. Provided schemas allow independent software vendors to integrate with the government system. Unfortunately, the schemas are not developed and extended to other sectors of agricultural production. It must be stressed that these are not the only XML-based standards used in modern agriculture. We have many standards that are not public and their internal documentation is not available. For example the Finnish company Bit-
Comp prepared software named BitFarm, which uses XML to communicate with IACS in Finland. In the United States a consortium developed standard AgXML to support the transmission of commercial documents in cereals and oilseeds industry. This standard is available, but it is not free. Its application is limited to a group of U.S. producers. Somewhat similar American initiative is mpXML (Meat and Poultry XML) suitable for livestock production. It is used to transfer data between producers and processors of meat products. The basic idea is to trace the origin of meat in the whole chain of suppliers. The basic documents relating to the standard are available. In Poland, we also try to use XML-based formats in areas related to agriculture. The most interesting initiative is prepared by Agricultural Market Agency set of XML schemas - eWnioski. Unfortunately the resulting file must be sent to the agency by e-mail or on electronic media and shall be signed by XAdES. Such an organization does not give too many opportunities for data sharing; however, is a good beginning for future interoperability. Another example is the activities of the Agency for Restructuring and Modernisation of Agriculture. Some forms of applications to import animal was prepared in XML format, at this moment there are seven forms, such as notification of slaughter of cattle in a slaughterhouse, the application of sheep slaughtered in a slaughterhouse or a goat, or the declaration of pigs slaughtered in a slaughterhouse. As in the previous case, this is only the starting point for the interoperability with other applications. Basically between the two initiatives, there is no connection, they are clearly subjects developed independently without further perspective. To sum up in Poland there are not yet any coherent strategy for the introduction of standards in XML format in the fields of agriculture.

The situation at the EU level changed in recent years strongly, the need for interoperability in agriculture applications becomes clear. Proof of this is both a subject and the results of projects carried out in the seventh EU Research Framework Programme. The first project worth mentioning is “FutureFarm”. One of the tasks - knowledge management in the future FMIS (Farm Management Information System), requires a determination of standards for data exchange. The project will be completed in December 2010, at the moment does not have final results, however, not yet developed evidence shows that the standard should be based on agroXML.

The second project is agriXchange [1]. The overall objective of this project is to coordinate and support the development of networks for sustainable development of common data exchange in agriculture. This will be achieved through: a platform for data exchange in agriculture in the EU to develop a reference framework for the interoperability of data exchange, and identify key challenges necessary to standardize the exchange of data related to agriculture. The project just started in January 2010 and will last three years, but already preliminary evidence shows that the most promising and advanced standard will be taken into account agroXML.
5. ONTOLOGIES IN AGRICULTURE

An ontology for agriculture should be a description of the concepts, identify their structure, relations between them and also determine the manner in which it must understand the concepts and relationships. This approach is the basis for the design and implementation of knowledge management applications in agriculture. The first historically ontologies connected with agricultural issues arose in the field of biology [4]. The need for biological ontologies has risen in recent years in large part due to the rapid development of large biological databases. Especially in genomics was created Gene Ontology (GO), it is a collaborative effort to address the need for consistent descriptions of gene products in different databases. GO is a part of bigger initiative Open Biomedical Ontologies (abbreviated OBO; formerly Open Biological Ontologies) [14]. It is an effort to create controlled vocabularies for shared use across different biological and medical domains. We have to note that there was developed special format for ontology representation called OBO format. Important initiative in the ontologies related to agriculture is the Animal Trait Ontology developed by the European Animal Disease Genomics Network of Excellence for Animal Heath and Safety Food with partners [11]. Following is Plant Ontology which objective is to integrate species-specific vocabulary terms into unified flowering-plant ontologies for rice, maize, Arabidopsis and other Angiosperms [15]. Other known ontologies are Gramene Ontologies [10], Ontology for Maze developed in Maize Mapping Project [12] and Ontologies & Controlled Vocabularies for Gene Annotation of Arabidopsis developed by The Arabidopsis Information Resource [18]. In the agricultural sector there exist some well-established and authoritative controlled vocabularies, such as FAO’s AGROVOC Thesaurus and the National Agricultural Library Thesaurus in the United States. However, for semantic tools to be entirely effective on the Internet, there is a need to move towards ontologies. At the moment there are not many ontologies dedicated strictly to agriculture, but the FAO Agricultural Information Management Standards (AIMS) website is an most interesting initiative. Its goal is to improving coherence among agricultural information systems that will make such systems interoperable. The objectives of AIMS are to create a clearing house for information management standards, and to share and promote the use of common methodologies and tools.[2]. The part of AIMS is Agricultural Ontology Service (AOS), which objective is to serve as a reference initiative that structures and standardises agricultural terminology in multiple languages for use of any number of systems in the agricultural domain and provide several services. The purpose of the AOS is to achieve more interoperability between agricultural systems [3]. Next part of AIMS website is the Agricultural Metadata Element Set (AgMES), this initiative aims to encompass issues of semantic standards in the domain of agriculture with respect to description, resource discovery, interoperability and metadata exchange for different types of information resources. Finally FAO has been concerned with develop-
ing a new model for the AGROVOC thesaurus that accounts for semantic and lexical relations in more refined and precise ways. The objective is to build a multilingual repository of concepts in the agricultural domain, the Concept Server (CS). The CS will serve as a base repository from where to build domain specific ontologies and export traditional thesauri, as well as other forms of knowledge organization systems. According to mentioned initiatives there are many other semantic technologies at FAO [17].

6. CONCLUSION

To ensure interoperability among information systems in agriculture, well-defined standards are necessary. XML-based standards seem to be the best solution. At the moment there is no leading standard in agriculture, but it seems that the most promising initiative is AgroXML. There is necessity to start in Poland research how to integrate Polish systems with AgroXML. Well defined standard does not support automatic interoperability at semantic level. The direction of development towards an ontology will allow such cooperation available. One can conclude that RDF and OWL formats are most promising. Right now good start point to prepare agricultural ontology is AGROVOC Concept Server and FAO experience. AgroXML can be developing towards ontology and ultimately can provide cooperation among agricultural information systems at semantic level.

Scientific work financed from funds for science in the years 2010-2011 as research project No. N N310 038538

REFERENCES


ONTOSLOGY OF E-COMMERCE APPLICATION FUNCTIONALITY

Tomasz Ordysiński
Institute of IT in Management, University of Szczecin

Abstract. The number of on-line shops is constantly increasing annually parallel with offered product diversity caused by market needs. Obviously not each of those initiatives is going to succeed, however each time new Internet entrepreneur faces the problem of choosing proper e-commerce application. The issue, which in the beginning seems to be simple, occurs to be complex problem. The number of offers of Internet shops providers is extremely high (not mentioning opensource solutions). Additionally, this choice usually is a long time decision. The goal of this article is a trial of building ontology of e-commerce application focused on it’s functionality. The further development of this ontology would become the bases of an expert system supporting an entrepreneur decision in described matter.

Keywords. E-commerce Functionality, Ontology, Expert System.

1. INTRODUCTION

Research done on Polish Internet initiatives shows that the number of new on-line companies is constantly increasing. The main reasons of that situation are considered UE support programs for e-business development and new groups of graduates (young entrepreneurs) for whom Internet usage is natural from elementary school. 99% of them are doomed to failure but this 1% is successfully programmed, published in Internet and creates profit for it’s owner. Statisticis shows that most dynamically developed range of e-business in Poland is electronic commerce. Global network enables to trade in model 24x7 and the offer of Internet shops contains both electronic (e.g. music, reports, e-books) and traditional goods (e.g. books, CDs, household or computer equipment). Almost 70% of European Internet commerce concentrates in 4 countries: Great Britain, Germany, France and Belgium. Polish e-commerce has reached the same level as Italian and is much more dynamic than in Spain and Portugal. In 2009 Polish electronic market reached almost 2% of total national retail trade (comparing to 1.6% in 2007) [1]. There are more than 7.3 thousand on-line shops operating in Poland and their turnover in 2009 was calculated on 9.6 billion of PLN [2].

Concerning so many e-commerce initiatives there appears a question how new entrepreneurs proceeded from idea to specific on-line shop. Most of them are not
IT specialists but still they had to make a decision, which e-commerce application should they use. Searching the Internet shows two main and usually in parallel used solutions: checking user’s opinions on on-line forums about specific application and studying, rather rare and not very complex, e-commerce specialist’s articles.

The problem is the number of those open source and commercial application offered by many vendors. At first glance there seems to be no or very small differences between e-commerce solutions but after some period of usage the user discovers the olf truth: “devil is in the details”. Sometimes lack of automation of one process can cause usage frustration and when the software was expensive there can be a problem to change it. That’s why detailed studies of complex functionality of e-commerce application is necessary, first to reduce a set of possible solution and later choose a proper one. Semantic character of the problem and indispensable classification flexibility points ontology as suitable description concept.

The article presents of short report about proceedings in ontology application in the area of e-commerce solution’s functionality.

2. FUNCTIONALITY OF E-COMMERCE APPLICATION

For some businesses, e-commerce is a natural fit. Retailers of small consumer items – clothes, gifts, books, electronics, packaged foods – are a perfect example. The products are easy to choose, commonly sold online, and inexpensive to ship. However each entrepreneur must ask him/herself several questions concerning logistic, products, customers, promotion or shop administration matters. The range of those areas can change and for specific business different area can be the most important. In logistic the e.g. warehouse issue is important: how many product would be hold in own storage and how many would be ordered for special customer order; what customer delivery options would be or how the returned product procedure would be organized. Product area causes questions about product presentation (how detailed description, photos, movies) or product navigation in case of huge assortment. Customer issue matter can focus on customer profile (what features would be important to collect) or customer type: retailers or only individual customers (what would be price policy – discount options: manual or automatic rules). Marketing or promotion branch must cover issues like unique e-shop graphical layout, e-commerce passages integration or simply the sale statistics for campaign planning.[3] From the administration (back-end panel) the automation seems to be the most important – questions like: how long customer order service takes to or how much time I need to input/update product’s catalogue. Those are only exemplary questions – to choose proper e-commerce application the entrepreneur must ask much more and get specific and certain answer.
Functionality features, comparing different e-commerce solution, can be identified optional in both types: available/non-available or more complex collection of settings e.g. area of marketing management.

Most professional e-commerce systems calculate appropriate taxes and shipping costs, but owner may have to do some setup work: entering shipping costs for each item in your store, for example. One common complication is items that have multiple options: colors, sizes, and the like. The simplest version has just one option per item; slightly more complex systems can handle items with two or three options; much more robust systems can handle pricing that changes based on the options or certain combinations of options that aren't available. Any worthwhile e-commerce system should connect to specified merchant account to accept credit cards for payment. Other popular features include automated emails to customers, coupons or discount codes, and "related items" links.[4]

Comparing a set of e-commerce applications we can classify functionalities into basic and advanced. The basic ones are:

- Product catalogue and basket – the customer can collect products from different categories and afterwards decide to put the order. The order process in simplest form is just an e-mail to shop owner with a list of ordered products and customer data. However nowadays standard in that matter is a guided process with a order status change and automatic customer notification.
- Specific product promotion in a form of crossed previous price and the new, lower just next to it. It can be also new product’s module, bestseller box or recommendation “Customers, who bought this product bought also …”.
- Catalogue navigation (breadcrump) and searching (simple/advanced)
- Payment and delivery options.

Advanced functionalities are offered by more complex script. There are features like:

- Electronic payment option from money transfer to credit cards service.
- Product comparison module – useful for the customer when shop offers almost similar products.
- Price and discounts management (order amount or items number rules; discount coupons etc.).
- Newsletter and mailing.
- Some accountancy tasks e.g. invoice printing or just direct integration with specified accountancy system.
- Loyalty program.
- Partnership program.
- etc. … [5]
The above classification is just an example – if we consider real decision situation we must get really deeper into each of presented points. For inexperienced entrepreneur there is no way to check hundreds of e-commerce applications across all those functionalities. The most common solution for them is to base their choice on expert opinion given usually in Internet as an article e.g. 15 Open Source eCommerce Platforms presenting most popular platforms, breaking each one down into its respective pros and cons. Most share a basic set of functions, but offer a couple of unique features on top [6]. Another example is portal presenting four e-commerce platform in a form of table comparsion [7]. In case of further questions and doubts new e-commerce entrepreneurs check Internet forums which are full of posts titled “Which e-commerce shop shall I choose?” (e.g. http://www.biznes forum.pl/ or http://www.bankier.pl/forum/).

As we can see the problem exists cause there is an enormous number of e-commerce solution with almost the same functionality features. However when someone has to spend some or much money for specific one, he would like to be somehow advised. For inexperienced users the best would be naming condition in almost natural language to pick a platform suitable for his needs. As solution ontology of e-commerce application functionality is proposed.

3. ONTOLOGY AS COMPLEX DESCRIPTION METHOD

According to W3C Recommendation (10 February 2004) an ontology defines the terms used to describe and represent an area of knowledge. Ontologies are used by people, databases, and applications that need to share domain information. Ontologies include computer-usable definitions of basic concepts in the domain and the relationships among them (note that here and throughout this document, definition is not used in the technical sense understood by logicians). They encode knowledge in a domain and also knowledge that spans domains. In this way, they make that knowledge reusable. The word ontology has been used to describe artifacts with different degrees of structure. These range from simple taxonomies, through metadata schemes to logical theories. The Semantic Web needs ontologies with a significant degree of structure. These need to specify descriptions for the following kinds of concepts:

- Classes (general things) in the many domains of interest.
- The relationships that can exist among things.
- The properties (or attributes) those things may have [7].

Universal character and flexibility of reflecting real world issues made ontologies very popular. Several languages (from RDF, XML to OWL) and universal ontologies and editors were adopted or developed and are constantly improved. The list of application which can help in building an ontology divided into subcategories is presented on Michael Bergman website [9].
Due to international researches concerning domains like gene, geography etc. some methodologies of ontology building were identified. The most popular are:
- On-To-Knowledge,
- MENTHONTOMETRY,
- Uschold and King’s,
- CYC,
- Gruninger and Fox’s,
- DILIGENT,
- KAKTUS,
- SENSUS,
- Noy and McGuinness.

However some researches shows that in almost 60% of cases when an ontology is built no methodology is used. All the process is based on expert knowledge of analyzed domain, his intuition and several prototypes of prepared ontology [10].

The usual steps of creating new ontology are: identification of ontology scope, capture phase, encoding phase, ontology integration, ontology evaluation and finally ontology documentation. They are built for specific applications like problem solving methods, domain-independent applications or software agents.

![Semantic web architecture in layers. Source: http://www.w3.org](http://www.w3.org)

Ontologies can be specified at different levels of formality – the same classification can be expressed in natural language (highly informal), structured form of
natural language (semi informal), formally-defined language (semi formal) and formally with theorems and proofs (formal). WEB 3.0 assumes semantic WEB to be the future of Internet (Fig. 1).

4. ONTOLOGICAL DESCRIPTION OF FUNCTIONALITY OF E-COMMERCE APPLICATION

The research work started with identification of suitable ontology editors. There were checked and tried several option (open source and commercial) and the final set, which was examined more carefully was: OntoStudio, Swoop, Protégé from 3.1 to 4.1 beta version. According to trial ontologies building the most friendly solution was OntoStudio however the 3 month evaluation period was not enough to complete planned research. In future (in case of given research grant) a purchase of this software is predicted. Cause of the financial reasons Protégé platform in version 4.1 was chosen. For the consistency check there were installed and tried several reasoners like HermiT, Fact ++, Pellet which current versions are compatible with chosen Protégé platform.

Next step was identification of e-commerce functionality features which were classified in hierarchical list of options. That stage was based on literature studies, several e-commerce platforms trial (self-installed and on-line demo) and expert knowledge.

Finally the list was implemented in Protégé platform as set of classes and sub-classes in domain of e-commerce application functionality with defined object properties. The reasoned confirmed ontology consistency. An example code in RDF/XML is presented below.

```xml
<!-- http://www.owl-ontologies.com/ecommerce_functionality.owl#LoyaltyProgramManagement -->
<Class rdf:about="&ecommerce_functionality;LoyalityProgramManagement">
  <rdfs:subClassOf rdf:resource="&ecommerce_functionality;MarketingManagementOptions"/>
</Class>

<!-- http://www.owl-ontologies.com/ecommerce_functionality.owl#MailingManagement -->
<Class rdf:about="&ecommerce_functionality;MailingManagement">
  <rdfs:subClassOf rdf:resource="&ecommerce_functionality;MarketingManagementOptions"/>
</Class>

...  

<!-- http://www.owl-ontologies.com/ecommerce_functionality.owl#hasOption -->
<ObjectProperty rdf:about="&ecommerce_functionality;hasOption">
```

62
Next step was creation of a set of e-commerce application. Due to performed tests on future functionality of designed ontology the examples of e-commerce applications were implemented as classes with specified individuals. For the ontology testing there was chosen two open source platforms: OsCommerce and QuickCart. A part of QuickCart definition is presented below.
For the future research development in direction of expert system some defined classes were created and checked with DL Query module giving expected results. There were also some trials in SPARQL usage but it required conversion of designed ontology for Protégé 3.4 requirements and caused some problems. An example code of defined class is presented below.

```xml
<!--http://www.owl-ontologies.com/ecommerce_functionality.owl#WideRangeOfElectronicPaymentsOptions -->
<Class rdf:about="&ecommerce_functionality;WideRangeOfElectronicPaymentsOptions">
  <equivalentClass>
    <Class>
      <intersectionOf rdf:parseType="Collection">
        <rdf:Description rdf:about="&ecommerce_functionality;EcommerceApplication"/>
        <Restriction>
          <onProperty rdf:resource="&ecommerce_functionality;hasElectronicPaymentOption"/>
          <minCardinality rdf:datatype="&xsd;nonNegativeInteger">3</minCardinality>
        </Restriction>
      </intersectionOf>
    </Class>
    <rdfs:subClassOf rdf:resource="&ecommerce_functionality;EcommerceApplication"/>
  </equivalentClass>
</Class>
```

The final shape of designed ontology is presented on Fig. 2. The number of classes and sub-classes was limited for the purpose of this article giving just a prototype of final solution. During the analysis and ontology implementation phases it appeared that one more main class creates a list of sub-classes, which considered further where beginning of another sub-class. The ontology was growing in geometrical progress. Future research plan is to develop this set to options available in the most sophisticated e-commerce application. However the author realizes that the number will never be finished. Each day new functionalities are created to make e-commerce solution unique and competitive (at least for a moment) on the market.

During implementation process some problems with chosen software platform appeared. The main inconvenience was met during datatype restriction defining – there was a problem with numeric type recognition and it must be solved before project is continued. What is more some option available in Protégé version
3.4 were removed and author would found them very helpful during individuals definition process (lack of forms in this specific case).

The conclusion from the ontology implementation phase based on Protégé platform is consistent with general opinion about open source platform. They give huge functionality (huge number of plugins, tools etc.) and support planned research but in case of problems the time spent to solve it makes the project schedule extremely longer. Project continuation assumes purchase of OntoStudio license and migration to that commercial platform.

![Designed Ontology of e-commerce application functionality](source: self study)

**Figure 2.** Designed Ontology of e-commerce application functionality. Source: self study.

5. CONCLUSION

The goal of this article was to prove usability of ontology as a knowledge codification tool for e-commerce application functionality. Author’s literature and Internet resources research identified a problematic decision situation. The most complex tool, which helps the user to choose proper e-commerce platform was found on [http://www.ecommerceknowhow.com/](http://www.ecommerceknowhow.com/). The eCommerce Know-How Solution Finder shortcuts your research time in finding solutions that fit your needs. Create a short list of solution options based by filtering on key criteria that are important or relevant to your business. Once a shorter list is created, drill down to detailed information including key capabilities, proof points and pricing. [11]
However, there is no query language available and no way to develop the data set by anyone else accept the website owner.

The goal and future of information exchange is to give the user possibility to add his knowledge, experience or opinion to existing domain knowledge base and share it without any borders. Application independence of ontology enables such a operation keeping still the knowledge- and database consistent.

The further plan of designed ontology assumes classes and restrictions development to cover the most possible number of present e-commerce application functionality. Then the research will focus on query languages (SPARQL) and adopting the ontology as knowledge base of an expert system.

REFERENCES


http://www.egospodarka.pl/


http://www.webdesignecommerceguide.com/

http://www.brandonline.pl/2010/08/04/funkcjonalnosci-sklepow-internetowych/


http://www.i-sklepy.pl/sk.php?o=jakie_oprogramowanie_wybrac

http://www.w3.org/TR/nowbont-req/#onto-def


http://www.ecommerceknowhow.com/ecommerce-solutions/solution-finder/
PROCESS OF SOFTWARE DEVELOPMENT USING CMMI MODEL

Aneta Poniszewska-Marańda \textsuperscript{a), Artur Tworzydło} \textsuperscript{b)}

\textsuperscript{a)} Institute of Information Technology, technical University of Lodz
\textsuperscript{b)} Bravura Solutions Poland

Abstract. The paper shows a practical approach of the CMMI (Capability Maturity Model Integration) model. It presents the software development process build to solve management issues in a concrete situation, which takes place in a software solution office. Therefore there are many small tasks but with a wide range of difficulty. There are very different people working on the client's side, that is why the consultants may in parallel work with software architects or just administrators who do not know much about the software development. These consultants have to be mobile and be able to switch the projects very often as the client's requests are very randomized in time. The described situation needs a good requirements handling process and project progress tracking.

Keywords. Software Development Process, Capability Maturity Model Integration, Client Requirements, Management of Software Development.

1. INTRODUCTION

Nowadays, when the hardware and software are so advanced, the developers do not have to take care of the product performance or memory usage. The most important issue is to build the applications faster and to make them more and more useful, functional and user-driven. When to this situation we add the fact that many frameworks and software libraries make the building process of big applications easier even for people without much experience in the information technologies, it can altogether end up in one chaotic mess. The managers, mostly in new small companies, do not take care of the product quality and reliability. Nobody makes the tests because almost all the time deadlines are too short even to finish the implementation. A target of many companies is to sell the product, no matter whether it works correctly or not, as far as the client will pay. If there will be problems, the client will have to buy the updates and he will have no choice.

This article shows a practical approach of the CMMI (Capability Maturity Model Integration) model. It shows the software development process build to solve management issues in a concrete situation, which takes place in a software...
solution office. The company mostly deals with customization and bug fixing projects for several huge products. Therefore there are many small tasks but with a wide range of difficulty. Even though the office and the consultants are prepared for servicing international clients, there are many issues with correct acquiring requirements from the clients. There are very different people working on the client's side, that is why the consultants may in parallel work with the software architects or just administrators who do not know much about the software development. These consultants have to be very mobile and be able to switch the projects very often as the client's requests are very randomized in time. There are times when a client sends so many tasks that several consultants have to solve them, another time the client may suspend sending requests because of many reasons like department budget, strategic projects or employer free days. The described situation needs a very good requirements handling process and project progress tracking.

The objective of this paper is to present and implement the CMMI model for the software development process that can be used in software companies. The first part of the paper presents the outline of the CMMI model and the second describes our implementation of software development process based on this model.

2. CAPABILITY MATURITY MODEL INTEGRATION

Thanks to the flexible nature of the CMMI it is possible to adapt the model according to two different representations: continuous and staged [1, 2].

The continuous representation offers a great flexibility in using the model in a process development. Its big advantage is choosing only one process area that may be especially important for a company without needing to learn the entire model. There is also possible to choose a pack of several process areas strongly related to each other that are most important for the company core management. Every process area may be deployed at different speed according to the actual needs and situation. The continuous representation is the best solution when a company knows which processes are the most important.

The contrary representation is the staged one that propounds the systemic and fully hierarchical attitude for the process development in a company. It is only possible to deploy one level of the representation at one time because every level depends on the previous one. The process performance is staged by the maturity levels. All process areas are associated with the maturity levels so a company has to develop the given process areas in order to reach the higher stage. This representation is a good choice when we do not have much knowledge about the company structure and actual processes. It gives the certitude that all process areas are developed in correct order [1, 2].

Each representation has its own gradation rules and levels. The continuous representation uses the capability levels and the staged representation uses the ma-
turity levels. Both types of levels have the same target that is describing the improvement level and managing the deployment of updates. To achieve the chosen level it is obligatory to deploy all connected rules.

The main difference between both gradation systems is that the continuous representation centers on the process areas capability and the staged representation takes care of the maturity on an organizational level. The capability levels describe the process development according to a process area. There are six of them and they are numbered from zero to five: Incomplete, Performed, Managed, Defined, Quantitatively Managed, Optimizing. The maturity levels appraise the process deployment of an entire company taking care of several process areas simultaneously. There are five of them and they are numbered from one to five: Initial, Managed, Defined, Quantitatively Managed, Optimizing.

Although the CMMI is well described and can be used in many configurations, its implementation and deployment is not an easy and fast task. The managers not only have to know this model, the representations and process areas but they also have to understand the company needs and the working mode. Even though the model is very big and contains various organizational activities not all of them have to be implemented at the same time. This task has to be divided into some smaller phrases.

3. SOFTWARE DEVELOPMENT PROCESS BASED ON CMMI

One of the first decisions to be made is whether to choose one of the existing processes or to create a new one. There are a great number of software development processes, methodologies and guidance how to create the perfect software without any risk. In the situations when the project environment is quite original and not typical for most software development processes, a good idea is the creation of a totally new process that will be resistant only to our management issues, not to other ones that we may never run across. Because the CMMI model is described very well, has long lifetime and has been tested by many professionals we found it a very good product for us [3].

Another, an even harder decision was to choose a model representation that would suit our situation best. For the new offices that do not have any management it is a very time-consuming and risky task to build a new process based on the staged representation. The company would not be able to work properly without acquiring at least the second maturity level, which may take a lot of time and money. Using the continuous representation needs some software development process in the company that could be upgraded. Because none of those requirements were fulfilled, we have decided to choose the most important process areas from the second maturity level and leave others on the same level for later work. Therefore the key areas we took into consideration were: requirements manage-
ment, project monitoring and control, and configuration management with some elements of the measurements and analysis.

![Figure 1. Phases of development process.](image)

The final process is an entirety and does not divide into the process areas. It is divided into high-level phases (Fig. 1). The first one is the *project inception* that takes care of preparing the work environment. Next into play comes the wide understood *analysis* that focuses on the understanding of the client's requirements and formulating its solutions [4]. The next phase is the *development and implementation* where a solution is built and tested. The last phase, the *deployment* takes care of providing and deploying the product to a client, to assure that the product will be used as it was supposed to. The three phases like analysis, development-implementation and deployment may be iterated. It allows a fast and coherent reaction towards the changes in the client's requirements. There is one more phase called *management* that takes place during the entire project lifetime and traces all other phases to assure that everything is on time and within the given budget.

### 3.1. Inception

Every new project begins with the environment preparation that supports the management. It is built of several elements described below. A project manager who can ask the trained administrators to do it takes the responsibility for them.

The first activity creates a project in the integrated modeling tool that in this case is the Enterprise Architect. There is a centralized server that hosts every project and traces modeled changes very similar to the source code repositories. Because this kind of configuration is complicated and requires a lot of technical knowledge, specially prepared scripts and tools come to the rescue. One of such tools, the *Project Builder*, helps to create and configure the new project connected to the version control system. The first thing user has to do is to give the project a name and an access password. It is also needed to provide the information about a database that will host the new project.

Next, a user can use a detail window where he can choose the template to use for the project. All templates are standard and kept on the global repository. Using
such temples guarantees that all projects will be compliant with the company internal standards. It is also possible to choose the archives with standard documentation templates of the company. In the end all, even different projects, are created according to actual company policy. The figure 2 presents exemplary project template.

Figure 2. View of new project created basing on a template.

The next step is to create the project structure for the source code tracking system. The source code tracking system has to exist practically in every programming project. It is a helpful tool during the programming as well as during creating any kind of documentation. To avoid misunderstanding and long time data searching in different projects we introduced one directory structure standard in a repository that every employee has to obey. In the main directory of the repository there exist many directories named according to its project name. It is common that these directories contain the subdirectories like: branches, tags, trunk, business-resources, communications, deployment, test and source-code.

The branches directory keeps the files associated with the project but in a different version than the actual one. The tags directory contains official and final product versions that should not be changed. The trunk directory has the actual files for a currently developed main project version. It has rich structure of subdirectories that contain different types of documents. Business-resources directory preserves the business objects such as templates, client specifications and product life cycle schemes. The communications directory contains the elements describing the communication history between different functional groups, e.g. reports of progress work. Deployment directory contains the objects used in the product de-
ployment process, i.e. installation scripts, database models, network configuration, data files. The doc directory preserves the product documentation. The tests directory contains the application tests. The main directory of each project is the source-code directory that contains the user interface code, the code of main application and the documentation generated from this code. This repository structure contains also one file with the instructions how to use a repository and its structure.

Even though we have set up the configuration of internal office work we also need to take care of a client and have appropriate contact with him. The clients do not have access to the company internal documentations and released products, therefore it is a good idea to use an easy tool, therefore the client could access the published new releases in one place. When the clients get their product in varied ways, different packages for each version of every product can become a problem. The tool Project Link is used. It only requires a web browser and also keeps track of the changes and allows easy communication with every invited person. After creating a new project (giving the project name, specifying the project template and time duration) we can specify which users will use it with different access rules. It is possible to define a role for each project participant. We can have the following roles:

- project manager who creates the project and modifies the project attributes,
- developers who build the project,
- client for whom the project is created,
- administrators of servers on which the project will operate,
- observers e.g. high level managers who do not change anything in the system but only oversee the work of others and are informed about each change.

It is the place where the developers, analysts and clients may come in contact, share their ideas, opinions and even files. The last step is a definition of detailed information about the project: project number, project category, project group and its priority. Finally the entire environment needed for a new project was created. It may take some work but in longer projects it is very helpful for the management and for the people to share data, communicate with each other and keep the information in the structured way.

3.2. Analysis

The first thing of the analysis is to understand the client situation and document it as a domain model "AS-IS". The target of this task is to better know and picture the client problems and reasons for a new project. Next, also according to the cooperation with a client another domain model called "TO-BE" is created. It shows a target situation that the client wants to have. This model is built on discourses with a client and shows the abstraction of the environment and its interactions (the situation does not exist yet). This is the target to which we aspire. The domain models contain mostly the abstract data that build the client's environment.
According to the differences between both models the model of real project requirements is built and it shows the client's requirements (Fig. 3).

![Diagram](image)

**Figure 3.** Client's requirements.

According to the domain models and the communication with a client, the client needs are determined [5]. These are mostly requisites and conditions under which the client will pay for the project. They are used to build the acceptance tests. The feature describes the visible functionality of the new product that responds to the client's needs. It also defines how to use the product in order to solve the client's problem. The features are realized by the use cases and by the non functional requirements. The example of the client's requirements presents the fig. 4.

![Diagram](image)

**Figure 4.** Example of modeled client's requirements and example of feature diagram.

The use cases defines what a developer has to implement to obtain the product functionality. The non functional requirements describe the system characteristics that do not correspond with the performed activities. They may come from the
technological boundaries, security or performance constraints. The non-functional requirements may be divided into four categories by taking into consideration its following features: usability, reliability, performance and support.

The client's requirements can be defined in different moments of the project realization because he can change them at any time. Therefore the requirements can have different stages (Fig.5):

- proposed - preliminarily suggested,
- approved - validated after exact definition and negotiating,
- incorporated - implemented and included into the product,
- obsolete – outdated,
- rejected - rejected, for example after the cost definition.

Figure 5. States of client's requirements.

After the definition of the client's requirements the system analyst starts to define the product features. He creates the feature diagram that is very similar to the diagram presenting the client's requirements (Fig.6). According to the feature diagram two other diagrams are created: use case diagram and diagram of non-functional requirements.

3.3. Development and Implementation

The purpose of this phase is to build the final product according to the analyses done before. The first task is to create the component model (i.e. component diagram) based on the domain model and the client's requirements. The developer specifies the main components and their interactions as well as the defined interfaces. Afterwards the class model should be created. Depending on the project needs there may be a different number of class diagrams with different levels of details. From this kind of diagrams the source code can be generated in various languages.

Next, the programmers have to implement the created solution using the chosen technologies and development tools like integrated development environments. It is important that all developers use the same development environment. If
needed, the user interface model can be created to prevent many problems like forgetting some obvious buttons.

In parallel to the previous activities, there are performed works on defining the tests. It is a very important aspect of building the software, very often treated without necessary caution. This phase is realized by using also the Enterprise Architect. The acceptance tests are associated with the corresponding use cases. Each test has its name that is also his number. The tests are described by the status that indicates whether a test is realized, passed correctly or not or it is canceled. If it was processed, the person who performed the test and the last date when it was run should be typed in. There is also a description of steps needed to be executed to perform the test and also the test prerequisites that should be granted to correctly perform such a test (Fig.6).

![Figure 6. Description of acceptance test.](image)

The most important part of a test description is the acceptance criteria that explain how to check whether the system reacted correctly or not. After performing these steps the final product is built and tested and the next phase can be started.

3.4. Deployment

During the work with the big systems the deployment phase becomes a highly complicated procedure. Every activity should be correctly planned and prepared. To help in the understanding of the cooperation of multiple systems the deployment model is built. This kind of model makes it easier to find the places where special consideration about technological configurations is required. After setting up the deployment plan of a new system at the client's environment the system can be taken from the repository. Each final configuration of a product is persisted in the tags. Each tag keeps a snapshot of all documents created in a concrete moment before the deployment. It gives the certitude that the product status, finished in the certain version, will not be changed during the future works on its next versions.

After the creation of the final product package it should be transferred to the client. An exemplary package sent to a client contains the file archive with the ready product and the documentation with user instructions (Fig.7).
Additionally it is possible to trace all object modifications. The system contains the information about the persons who put in the materials and make the data modifications. Also the commentaries to each product version or iteration numbers are visible.

3.5. Management

The project management is a separated phase that takes place during the entire project lifetime. It allows the acceleration of the reactions on arising organizational problems and takes care of the schedule of all tasks. The quality and coherence control in different project phases are also important.

A relational matrix is a very important tool enabling checking if all the client's needs are fulfilled in a new product. In an accessible and readable way it shows the report about the connections between different objects in a model. Therefore it is possible to check for example whether all features of a new product were implemented in modeled use cases, which answers the question whether the functionality modeled by the analysts solves all the client's problems (Fig.8).

![Figure 7. Structure of a package sent to the client.](image)

![Figure 8. Exemplary connection table.](image)
Another aspect of the project management is the assignment of employees. Here also the Enterprise Architect helps. It is possible to associate each object with the people with tasks to realize. Also the reporting of the work progress and planning of ending are accessible. To complete all these features a set of reports showing the project activities and status from different points of view can be generated.

In order to implement the integrated management and composite documentation a set of templates prepared according to the company standards can be used. These templates are very easily incorporated with the Enterprise Architect template system that allows the producing of various kinds of reports in one common standard for the entire organization. During the works the following templates were created:

- requirement template - documents all project requirements with their description and status,
- use case template - gives the requirement template concerning the use cases,
- test template - contains all project tests with their detailed description, status, creation date, realization data and scenario,
- class/component template - presents the objects included in the class models or component models with their descriptions and connections.

When a company changes its marketing targets, the changing of several templates and regenerating of the reports are much easier than replacing a great number of documents in different standards.

4. CONCLUSION

The process of software development is a very long and demanding task. Many companies and organizations try to build a perfect process that would solve all management issues that can be found during everyday work. It is of course impossible. Our target was to build the process that would fit our needs. It does not solve all issues but on the other hand it is very helpful for our profile, i.e. software development. The described activities are quite well integrated with each other. The found problems or lacks of it will be continuously improved in further works.

REFERENCES

Abstract. Development of tele-information technologies has enabled the appearance of various network organisational structures, which are often characterized by dynamic changes of the number of connected enterprises as well as used information resources. Because traditional integrated systems based on central databases cannot be used in such conditions, alternative solutions enabling common sharing of information resources in quickly changing environment have to be elaborated. In the paper the possibility of using ontology for data and information integration in network structures is discussed. Characteristics and components of such systems are described. In addition, the example of using them by heterogeneous organisations is explained.

Keywords. Integration, ontology, integrating models.

1. CONTEMPORARY DISTRIBUTED ORGANISATIONAL STRUCTURES

Tele-information technologies development has contributed to the appearance of new organisational forms, e.g. network structures based on mutual connections between companies do not related by capital. The network joins autonomous or slightly connected units. The crux of this organisational form is that independent enterprises, using tele-information technologies, form voluntary, often occasional configurations [7]. The networks may be organised within one country, becoming local, regional organisations or they may cross borders establishing global, supranational structures. The network connections may be temporary, when intensification of contacts between organisations changes in time and depends on current necessities, or more formal, if organisations connect formally creating corporations operating on specific area or having global range.

The aim of designing of networks is striving for improvement of operational conditions effectiveness, costs decreasing, growing of competences and so on. As far as benefits for enterprises connected with organising network structures are concerned there should be mentioned [1, 3, 6]:

- easier information flow between organisations,
- easier access to information,
• the possibility to use information resources belonging to partners,
• the increase of quantity of information resources and easier access to resources and skills belonging to partners.

Such organisational structures have also some weak points, including [11]:
• problems connected with mutual connections management,
• increase of transactional costs when the number of network participants increases,
• the possibility to emergence of egoistic and opportunistic behaviour,
• possible exploitation of weaker partners by those controlling the most important information,
• the problems connected with information management, its safety and security, which is more complicated in network structures than in traditional organisations.

The networks are especially useful when quick and precise information (essential from commercial and social points of view) flow is necessary [1]. Such organisations may be of various forms, establishing dynamic structures, often occasional. The structures are:
• cluster – this organisational form concentrates a group of companies belonging to the same branch as well as other organisations operating in the same region; establishing such a structure companies cooperate for mutual benefits, jointly organising distribution of products, marketing or promotion,
• technological park – this form joins economic institutions whose main task is to create conditions for running various businesses; the task is achieved by supporting innovative enterprises in development of products and manufacturing methods, the optimisation of technology transfer conditions and commercialisation of research results from research institutions to business,
• process organisation – it arises as a result of separation of activities which constitute processes fulfilled in an enterprise, followed by identification and naming of these processes in order to achieve reasonable management,
• virtual organisation – the organisational structure in which resources of cooperating enterprises are being connected in order to achieve higher benefits than in the situation when every company operates separately,
• fractal – contemporary organisational form arising mainly within one company; such a structure may be established by a whole company (macrofractal) or its organisational units (microfractal); fractals are based on self-organising and autonomous employee teams which have full access to quickly flowing information [2].

Discussed above network organisational forms (clusters, technological parks, process organisations, virtual organisations and fractals) join distributed and heterogeneous organisations, so tele-information technologies enabling fast exchange
of data, information and knowledge are crucial for their existence. So, it is necessary to elaborate and employ adequate processes management systems and information systems enabling separate gathering data and information but common their using and exchanging. The described organisational structures are dynamically changed, connections between individual partners are often informal, units, sometimes completely different, are merged, they are organised occasionally in response to appearing market needs, challenges and changing conditions. To achieve such dynamically changing functionality it is necessary to coordinate commonly fulfilled processes, establish connections between applications and systems employed by individual units and appropriately manage of separately gathered information resources. So, information systems designed for such organisations should have at least following characteristics:

• be scalable, which means the ability to enclose new units to the system,
• be flexible, which means the possibility to adjust their structures to changing environment,
• activities and processes should be fully or at least possibly highly automated,
• they should be as user friendly as possible, the structure of user interface has to enable, among other things, uncomplicated access to necessary information resources gathered by individual partners.

Because traditional database systems do not have such functionalities, alternative solutions that enable information resources management in dynamically changing environment in which organisations creating network structures operate have to be elaborated.

2. THE ARCHITECTURE OF INTEGRATING SYSTEMS BASED ON CO-SHARED ONTOLOGY

Significant characteristics of integrated systems based on central database are the lack of flexibility and scalability, for this reason they cannot be used in quickly changing network organisations. So, alternative solutions have to be elaborated. There have been various systems proposed in this area:

• for integration of various kinds of information resources [see e.g. 5, 10, 13],
• for integration of semistructural data [see e.g. 9, 12],
• Personal Information Management (PIM) systems [see e.g. 4, 8].

Due to the complexity of the problem of heterogeneous distributed information resources integration, independently of employed solution, an integrating system has to be composed of modules fulfilling partial tasks constituting the
whole process of integration. The exemplary integrating system consists of four layers (fig. 1):

- **local information resources**, comprising integrated component systems,
- **local communication wrappers**, being the interface to those information resources stored in local systems that are made available in integrating system,
- **global**, storing metainformation about co-shared information resources; this layer, among others, is used in the process of queries formulation,
- **user interface**, used for communication with users during queries formulation and is responsible for chosen information resources presentation.

The proposed solution has three crucial features being deciding factors as far as usefulness of information resources integration in dynamically changing environment is concerned:

- one of modules of the global layer is dynamically modified co-shared ontology comprising metainformation about information resources actually available in integrating system,
- local systems constituting the local information resources layer remain unchanged,
- queries processing is two-staged – information resources sent to users in response to their queries have to be finally processed in users’ local systems.

The whole process, from queries formulation to sending of chosen information resources to users’ systems, is automatic. In individual layers following tasks constituting the process are fulfilled (fig. 1):

- queries formulation (the user interface layer with the use of metainformation about available information resources stored in the ontology),
- sending of formulated queries (the user interface layer → the global layer),
- sub-system choice and sending queries to chosen sub-systems (the global layer → the local communication wrappers layer),
- queries reformulation, names translation and sending of converted queries (the local communication wrappers layer → the local information resources layer),
- the choice of information resources and sending them in untransformed formats (the local information resources layer → the local communication wrappers layer),
- names negotiation and sending chosen data and information (the local communication wrappers layer → the global layer),
- chosen information resources integration and sending them to users (the global layer → the user interface layer),
- final processing of obtained information resources (the user interface layer).
3. THE STRUCTURE OF CO-SHARED ONTOLOGY

Due to the popularity of XML and its wide application to various information resources exchange, this language was chosen for the ontology codification. In the proposed solution the structure of the ontology consists of three sections:

- **<concepts>**, comprising metadata about all concepts and connections between them; this section will be used to design graphic user interface enabling the choice of facts; the concepts chosen by users will be placed in the FROM clause of formulated queries; every concept has a name **<concept_name>**, category parameter which specifies its kind (which may be *hermetic, open* or *service*) and *info* parameter including understandable information for users about the concept’s content, helpful during the process of queries formulation; sub-concepts, being
the elements of sup-concepts, have also *predicate* parameter being used for understandable description of the connection; the structure of this section of the ontology is depicted in fig. 2.

```xml
<concepts>
  <concept_name category="hermetic/open/service"
  info="information about the concept’s content">
    [ {<concept_name category="hermetic/open/service"
          predicate="connection name"
          info="information about the content">
      </concept_name>
    }
  </concept_name>
</concepts>
```

**Figure 2.** The structure of `<concepts> section of the ontology. Source: own preparation. Description: in braces – set of elements, in square brackets – optional elements, slash – alternative, the choice of one of many.

- `<attributes>`, where, for every concept defined in the `<concepts> section, the attributes describing it are stored (fig. 3); the attributes are divided into groups:
  - `<where_clause>`, where attributes that may be used for the formulation of select condition are described; they may be of simple (parameter *type*) or validation (*validation*) types.; in the last case, during logical condition formulation users have to choose one or more values (more, if the parameter *where* has multiple value) from a dynamically prepared list,
  - `<others>`, where the rest of attributes characterising concepts are described,

```xml
<attributes>
  <concept_name>
    [ <where_clause> {<attribute_name type="type_name"/
                      |validation="validation_name" [where="multiple"]/> }
      </where_clause> ]
    [ <others> {<attribute_name type="type_name"/> }
      </others> ]
  </concept_name>
</attributes>
```

**Figure 3.** The structure of `<attributes> section of the ontology. Source: own preparation. Description: see fig. 2, additionally: | – marks of elements constituting alternatives.
• `<validation>`, comprising validation values sets for attributes described in the `<attributes>` section that are of `validation` type (fig. 4); this section is created only if at least one attribute in the `<attributes>` section is of the `validation` type.

```xml
<validation>
  { <validation_name>
    "value_1" ... "value_n"
  </validation_name> }
</validation>
```

**Figure 4.** The structure of `<validation>` section of the ontology. Source: own preparation. Description: see fig. 2.

4. THE EXAMPLE

In the exemplifying solution the conceptual project of the ontology was designed (fig. 5). The following decisions were taken:
• the ontology consists of five connected concepts,
• the attributes that may be used for definition of select and project operations are illustrated as ellipses; the double line ellipse specifies attributes that may be used for formulation of select condition by the choice of suggested values,
• remaining attributes, enabling only the definition of project operation, are put in a rectangle representing a concept, under the concept’s name.

**Figure 5.** A conceptual project of the ontology. Source: own preparation.
Next, according to the conceptual project (fig. 5), the structure of the ontology of the system in XML language was defined (figs 6, 7, 8).

The ontology, as in an established framework (figs 2, 3, 4) , consists of three sections. In the first one, <concepts>, all concepts and connections between them have been defined (fig. 6). There are two main concepts: <paper> and <person> and three concepts connected with the <person>: <academic>, <experience> and <education>. The second section (fig. 7) illustrates attributes description. For the <person> concept it is divided into two parts: <where_clause> and <others>, for the rest only <where_clause> exists. Appearing twice the <field> attribute is of validation="field_list" type and has a parameter where="multiple". A type field_list was defined in <validation> section and actually it consists of four values (economy, history, management, IT). The where="multiple" parameter means that, when formulating queries, during the definition of a project clause a user may choose any number of field_list elements and simple logical clauses created this way will be connected by OR operator. Similar situation exists in the case of the <degree> attribute of the <academic> concept. For this attribute in <validation> section the degree_list was defined.

Figure 6. The <concepts> section of the ontology (fig. 5). Source: own preparation.
<attributes>
  <paper>
    <where_clause>
      <title type="string"/>
      <author_name type="string"/>
      <author_surname type="string"/>
      <year_of_publication type="year_type"/>
      <field validation="field_list" where="multiple" />
    </where_clause>
  </paper>
  <person>
    <where_clause>
      <surname type="string"/>
    </where_clause>
    <others>
      <name type="string"/>
      <tax_id type="string"/>
      <date_of_birth type="date"/>
      <date_of_employment type="date"/>
    </others>
  </person>
  <academic>
    <where_clause>
      <degree validation="degree_list" where="multiple"/>
    </where_clause>
  </academic>
  <experience>
    <where_clause>
      <field validation="field_list" where="multiple"/>
    </where_clause>
  </experience>
  <education>
    <where_clause>
      <field validation="field_list" where="multiple"/>
    </where_clause>
  </education>
</attributes>

<validation>
  <field_list>
    "economy" "history" "management" "IT"
  </field_list>
  <degree_list>
    "master" "doctor" "assistant professor" "full professor"
  </degree_list>
</validation>

**Figure 7.** The `<attributes>` section of the ontology (fig. 5). Source: own preparation.

**Figure 8.** The `<validation>` section of the ontology (fig. 5). Source: own preparation.
Own data types have been defined by using XML-Schema (fig. 9). For the <year_of_publication> element an integer type named year_type was defined with the scope between 1900 and 2100. Because a list of fields (field_list) and academic degrees (degree_list) are sets of values, the list_type type was defined. It was indicated that field_list and degree_list are of list_type type. The available values list for these parameters were defined in the <validation> section of XML file containing the description of the ontology. To sum up, defining logical clause for the <paper> concept users may establish <title>, <surname> and <name> as texts, <year_of_publication> as integer between 1900 and 2100 and any set of values for the <field> attribute – the set is defined in the <validation> section of the ontology.

```
<xs:simpleType name="year_type">
  <xs:restriction base="xs:integer">
    <xs:minInclusive value="1900"/>
    <xs:maxInclusive value="2100"/>
    <xs:pattern value="\d{4}"/>
  </xs:restriction>
</xs:simpleType>
<xs:element name="field_list" type="list_type">
  <xs:element name="degree_list" type="list_type">
    <xs:simpleType name="list_type">
      <xs:list itemType="xs:string"/>
    </xs:simpleType>
  </xs:element>
</xs:element>
```

**Figure 9.** XML/S for attributes placed in the ontology (fig. 7). Source: own preparation.

5. CONCLUSION

Traditional integrating systems based on central databases are not appropriate for integration of dynamic network structures. So, it is necessary to search for alternative solutions. Among them there are information resources management systems based on dynamically modified ontologies, comprising metainformation about co-shared data and information. However, such solutions are more complicated as far as technological aspects are concerned. So, when embarking on designing such a system it is necessary to solve a considerable number of problems, such as: a system architecture elaboration, a query language definition, the design of modules fulfilling complex task of information resources searching and integration. In this paper a system integrating various information resources was proposed. Its architecture was discussed, the structure of co-shared ontology was characterised and the example of using the ontology was talked over. The elaboration of instructions concerning the problem of design and exploitation of remaining elements of the proposed system will enable designing of flexible and scalable solutions which will contribute to various information resources co-shearing in distributed and dynamically changing environment.
REFERENCES


Abstract. The article presents the modelling framework of the ADOit system, as well as its methods for IT architecture and process management. An analysis is made of the prospects for applying an integrated process management system to the modelling of IT architecture and processes within an organisation. A process-oriented approach towards improving IT architecture and process management in a company is presented. An analysis is performed of contemporary project and implementation frameworks for IT architecture management. Also presented and discussed are the characteristics of ITIL and COBIT, frameworks for the management of IT architecture and services. The applicability is investigated of the ADOit system for controlling and supervising an organisation's IT network infrastructure, and for modelling network infrastructure with the use of the OSPF multi-area routing protocol. It is shown that using the ADOit system for IT architecture modelling and management optimizes and evaluates various actions without the need to make unnecessary and misguided changes within an organisation. Advantages and disadvantages are identified of current ADOit solutions with regard to IT architecture management in terms of their design and applicability.

Keywords. ADOit, Enterprise Architecture Management, IT architecture management, BPM, IT governance ITIL, COBIT.

1. INTRODUCTION

Information technology (IT) plays a fundamental role in today's organisations, as it permeates almost every aspect of corporate operation. Consequently, expenses for IT-related investments often exceed half of the company's investment expenses. This results from the fact that the only way the company can adapt to the constantly changing realities of today and achieve the flexibility needed for dealing with customers and competitors is to employ modern management concepts and methods [1]. One of the prerequisites for efficient corporate management is the use of IT systems, many of which support and implement modern management methods, such as the service-oriented approach [2, 3]. The idea of a service-oriented management emerged in the previous century thanks to growing
popularity of modern management concepts, such as Total Quality Management (TQM), reengineering, value chain analysis, or benchmarking. Most of contemporary concepts concerning methods for modern management share a common thread, i.e. service management. This approach aims to optimize business processes and to adapt them as much as possible to the organizational structures, as well as to available IT architecture and resources. The issues of the service-oriented approach were discussed by T. Davenport, J. Becker and D. Khan in [4, 5] and by K. P. McCormack and W. Johnson in [6]. A typical feature of modern IT systems supporting organizational processes is that they provide a clear graphical visualisation of all the relations and processes within an organisation [7].

By systematizing the approach towards effective IT architecture management, the company can maximize its value. The operation of today's economic units directly depends on support from IT systems. By implementing appropriate IT solutions not only can organisations lower costs, but also create new sources of income. On the other hand, IT systems tend to incur additional expenses that are not always justified. Adequate IT architecture management is a major issue for today's organisations. IT services generate certain costs, depending on functional and non-functional requirements. The services should be therefore carried out in a cost-efficient manner. Adequate structures and mechanisms supporting IT architecture and IT-related issues effectively improve a company's business operation. The service-oriented approach towards IT architecture management proves particularly useful. Initially, the term “architecture” had a purely technological meaning in the subject literature. In 1990s, however, the concept of the information system (IS) architecture emerged [8, 9]. Issues concerning the IS architecture were discussed by Sage [10], Cortad [11], and Cook [12]. Academic centres and business enterprises are now intensively investigating and developing management solutions based on principles of IT governance and the ITIL (Information Technology Infrastructure Library) [14, 15]. Service-oriented frameworks for IT architecture management can be employed in private enterprises, as well as in public administration units. Their paradigm defines decision areas and mechanisms, and provides solutions that facilitate IT architecture management. Therefore the article aims to analyse methodologies for managing processes and IT architecture with the use of an integrated system supporting the service-oriented approach. The tool used was the ADOit system, developed by BOC Information Technologies Consulting. The subject of the IT architecture modelling analysis were processes executed within an organisation's network infrastructure. The network was configured to use OSPFv2, a multi-area routing protocol. Additionally, the applicability was evaluated of the ADOit system for modelling processes and integrating them with work environment.
2. MODELS FOR IT ARCHITECTURE AND SERVICE MANAGEMENT

Management and supervision of IT services pose a serious challenge for safe and economical operation of IT infrastructure, and are essential for efficient company management. One of the solutions addressing these issues is Enterprise Architecture Management (EAM) [16], which enables the company to develop a complete architecture according to its strategic guidelines and the requirements of individual business areas. According to Giga Information Group and Forrester Research [17], the main reasons behind the efficiency of enterprise architecture management are that:

- IT project costs can be reduced by approx. 20% by avoiding constant surveying and collecting information about the IT environment;
- Unnecessary elements of the architecture and redundant expenses can be identified and eliminated, which reduces IT operating costs by approx. 20%;
- Costs can be minimized due to the synergy effect resulting from merging application development requirements which do not contribute to the company's profitability (in approx. 10-20% cases).

According to research statistics, investments into architecture management can generate significant profits, but they require appropriate methods for IT architecture management within the company. Diversified business requirements often require complex and diverse IT environments, which can include a large number of different applications, interfaces and features, and which comprise interdependency networks. Managing individual elements of a complex IT environment poses a major challenge and requires a well-thought-out management strategy. The adequate mechanism of supervision over the IT systems and their development is a significant consideration when evaluating the reliability of a company; it is also one of the principles of the so-called “corporate governance” [18].

2.1. Cobit framework

There is a number of reliable methods and standards for IT architecture management which offer a framework for evaluating technical and organisational solutions, improving the control over complex architecture and providing quality IT services. One of them is the set of best practice reference models defined in the COBIT (Control Objectives for Information and related Technology) standard [19]. The COBIT constitutes an IT management control framework which allows the company to develop a precisely defined policy of good IT management practices. It also provides information criteria which help ensure that information conforms to certain control standards defined within the framework. The information criteria
include effectiveness, confidentiality, availability and reliability. The COBIT methodology can be applied in order to assess whether:

- IT is aligned with the business goals,
- IT resources are used responsibly,
- IT maximises benefits,
- IT-related risks are managed appropriately.

The main advantages of implementing the COBIT processes as the management control model are that IT is better suited to the company's needs and that it can be presented to the management in a clear manner. COBIT clearly determines information ownership and management responsibilities based on the process approach. Moreover, it has become a standard widely recognized by employers and third parties. Finally, it ensures the consistency with internal control requirements defined in the COSO (The Committee of Sponsoring Organizations) policy concerning IT control environment. The COBIT framework enables IT architecture management in three areas: business objectives, IT processes, and IT resources (Fig. 1). Each business objective can be described by seven information criteria, which are also used to measure whether the objective has been met (e.g. the criterion of confidentiality ensures that sensitive information can only be disclosed to authorised individuals). The area of IT processes consists of three segments: domains, processes, and activities within those processes. COBIT defines four specific domains: planning and enterprise, acquisition and implementation, delivery and support, monitoring and evaluation; they correspond with the lifecycle of IT solutions in an organisation.

Figure 1. The basic principle of the COBIT framework. Source: [19].

2.2. Itil framework

The concept of IT services management extends the range of support provided by IT departments (through ICT equipment and software): it applies IT services and the possibilities offered by information technology to achieving measurable business goals. One framework that can aid that process is the ITIL, which defines
a set of best practices for information technology services management. In its early stage of development, the framework aimed to provide support for ICT infrastructure management projects developed by British government agencies. The idea proved nothing less than revolutionary, and what originated as a means to ensure the IT area in British institutions was well-organised became a worldwide standard over the next years. Nowadays, the ITIL is no longer limited to IT services, but also supports business processes and includes a set of procedures based on best practices recommended by IT experts, IT solutions consultants and ICT users. It is independent of the type of business operation, technologies used within an organisation and technology providers. The first ITIL book was published in 1989 by the CCTA (Central Computer and Telecommunications Agency). In May 2007, the version 3 of ITIL was issued as a result of research commissioned by the OGC (Office of Government Commerce). The ITIL v3 defines the service lifecycle model and its five stages: Service Strategy, Service Design, Service Transition, Service Operation, and Continual Service Improvement.

**Service strategy** – is the key stage of the service lifecycle and provides guidelines for aligning business and IT objectives by formulating policies and measures required to achieve appointed goals. Each stage of the service lifecycle should focus on customer outcomes and relate to all the following process elements which interact with the planned service. Service Strategy embraces services and value chain management, bringing information technology into accord with business requirements, and implementation of service provision strategies.

**Service Design** – based on the business requirements and policies formulated in the previous stage, it defines the means required for the preparation and implementation of the planned service. Service Design provides guidance on service design objectives and elements, benefit and risk analysis, cost models, and measurement and control methods.

**Service transition** – focuses on the delivery of services and encompasses management and coordination of all the elements required for designing, testing and implementing the service with a focus on inter-service relationships.

**Service Operation** – describes delivery and control activities to achieve operational excellence on a day-to-day basis. It is the key phase of the lifecycle from the customer’s perspective, since it is at this stage that the service generates profit for the end-user. It coordinates all the means needed for service operation. It also covers the concepts of application management, change management, operations management, and control processes and function.

**Continual Service Improvement** – emphasizes the importance of service quality improvements, defines business and technological factors for improving services management, and tackles service retirement scenarios. The key areas of
activity at this stage include assessing the compliance with processes and procedures, and controlling the effectiveness and efficiency of services and processes. The main advantages of implementing the ITIL guidelines include:

- Optimization of IT structure with respect to current and anticipated business objectives,
- Constant reduction of total ownership cost for IT systems thanks to a more efficient IT management,
- Improved quality of IT services, resulting in higher reliability of IT and service provision systems,
- Enhanced communication between IT and other departments due to the development of a common IT policy and methods for achieving efficient data exchange,
- Improved customer relations by focusing on customer outcomes and employing a service-oriented approach to IT management.

![Figure 2. ITIL service lifecycle model. Source: [20].](image)

To summarize, it can be stated that the ITIL v3 is first and the foremost a description of the full IT service lifecycle. The framework addresses the basic question, “How to deliver quality IT services and provide customers with business value?” A major advantage of implementing the ITIL is a standardization of modelling languages within IT architecture. The framework helps create a reference model for IT processes, definitions, roles, and measures. A full description of the ITIL v3, however, requires a thorough analysis of the five service lifecycle stages listed above. Publications on the subject are available at the official ITIL website (www.itil-officialsite.com).

3. ARCHITECTURE MODELLING AND MANAGEMENT WITH ADOIT

ADOit is an IT architecture and service management tool that belongs to an integrated software package developed by BOC Information Technologies.
Consulting. It can be integrated with ADONIS, a business process management system, which felicitates business and IT process modelling and simulation. IT process models created by the ADONIS system can be exported to ADOit and used for modelling specific processes. The system features a graphical presentation format for displaying IT architecture and models; it also provides capabilities to perform analyses and produce full documentation for the planned IT infrastructure and services. The wide range of available model types and objects enable a faithful representation of even the most complex IT environments and models. Since the system follows a meta-model based approach, it can be customized to the customer's specific requirements. The predefined meta-models can be subdivided into several layers, such as business architecture, information systems, or technical architecture. The analysis and reporting framework of ADOit assists with decision making, report creation and query preparation. The system also allows for the use of modelling methods based on the COBIT and ITIL v3 frameworks, which were already discussed.

The first step of effective IT management should be identifying the current state of the base architecture (sometimes referred to as the “as-is” architecture). This facilitates tracking changes and evaluating the results of planned software and architecture changes by comparing the “as-is” state against a “to-be” state. Creating the base architecture model is the starting point for subsequent decisions on various scenarios concerning the implementation of ADOit to the company's IT environment development strategy. The accurate architecture model simplifies the process of identifying and evaluating risks related to architecture modifications. ADOit provides a built-in analysis framework for visualising the effects of business requirements on IT architecture. The requirements can be evaluated and prioritised based on cost criteria and strategic importance.
4. OSPF ROUTING PROCESS MODELING

OSPF (Open Shortest Path First) is a classless link-state routing protocol. The IPv4 communication currently uses OSPF Version 2, which was introduced in RFC 1247 and updated by J. Moy in RFC 2328 [20, 21]. Multiple areas can be configured within an OSPF-based network. Dividing large networks into smaller areas simplifies administration by minimizing the link-state database size and providing the possibility to isolate unstable areas from the rest of the network. Multi-area OSPF routers were implemented in the investigated model. The main advantage of the OSPF protocol is fast convergence and excellent scalability to much larger network implementations. Fig. 4 presents an OSPF-based ICT infrastructure model generated in the ADOit system. The first phase of ICT infrastructure management was to create connection models between the OSPF backbone area (area_0) and its neighbouring areas (area_1 and area_2), which together comprise a wide area network controlled by an organisation. In order to represent the difference between bandwidths, links were marked with solid and broken lines, which represent various types of WAN connections. A typical wide area network encompasses a large number of Ethernet and fibre-optic connections that are highly diversified in terms of connection speed and standards. From the perspective of IT administration they comprise an extremely important infrastructure element. By analysing intra-network connections potential bandwidth bottlenecks can be identified. VLAN networks controlling the departments of logistics, accounting, and trade were arranged between the edge areas (area_1 and area_2). In the next stage, the localisation of network equipment, such as routers and switches, must be determined. It is a good idea to use the name of the place where a router or switch is located to simplify the administration of large WANs and LANs. For this purpose, ADOit uses an ICT localization model. In order to ensure valid and up-to-date information on the network technologies and standards, a comprehensive review of IT architecture and software must be conducted. Thanks to its software architecture models, ADOit provides an easy way to implement reference models of software information.

![Figure 4. ICT infrastructure generated in ADOit. Source: personal study.](image-url)
The structure of ICT process course is related to organizational structure, as each process is assigned its executor – an employee (e.g. a network administrator) who is responsible for the process realization (e.g. flawless network operation in area_0). The process owner is responsible for individual process activities. He is assigned to a specific work environment, which determines his role. An example of work environment generated in ADOit is presented in Fig. 6a. The model takes into account all the employees who are process executors, and their respective organizational units were marked. It was asserted that organizational structure comprises three units: the department of trade, logistics, and accounting, plus an IT network administration unit. At the highest level, the organisation is supervised by the company manager. The employees can be assigned a number of additional information elements, such as education, availability, required skills (or the lack thereof), or availability during the working hours (Fig. 6b). The resulting models and reports provide a reasonable basis for ensuring efficient communication between all the individuals at all the architecture levels. ADOit features complex analysis and modelling methods, which facilitate visualisation of the influence of business objectives on IT architecture and a quick identification of interdependencies.
The objectives can be evaluated and prioritised according to certain criteria, such as costs or strategic significance. On-going modelling of all IT projects and aligning them with business objectives leads to a full integration between the objectives and IT development. The links between projects with architecture elements prove the existence of interdependency between different projects, which can be easily analysed and evaluated with the use of available reports and portfolio management.

5. CONCLUSIONS

ADOit is a complex tool for IT architecture and process modelling and management. It brings considerable benefits resulting from well-organised documentation on the “as-is” architecture of a company. Once an accurate model of IT architecture is created, further analyses become greatly simplified. It enables the evaluation of changes planned in the sphere of software, architecture, business processes, and other aspects of an organisation; it also facilitates the assessment of anticipated effects and simplifies risk identification and evaluation. The tool provides solutions based on IT governance and on the ITIL and COBIT, the two most popular service management frameworks in the world, which are intensively developed by academic centres and businesses. The ADOit system proved fully useful for modelling network infrastructure with the use of the OSPF v2 routing protocol; the modelling analysis showed that IT architecture implementation and management is fully possible in the investigated conditions. The architecture model that was created can be easily aligned to work environment, as well as to other models supporting IT management. A significant drawback, however, is the lack of symbols representing network infrastructure elements, such as routers, firewalls or the ASA equipment. In fact, there is only one symbol of the kind (for bridge/router-like devices). Graphical elements representing WAN connections are also very few. The limited number of graphical symbols makes it difficult to evaluate the ICT architecture model visually. If additional symbols and graphical elements were introduced, ICT network modelling would be much more effective and easier to view.

REFERENCES


[18] A. Zalewski, R. Cegielka, K. Sacha, Modele i praktyka audytu informatycznego, e - Informatyka.pl


THE ECONOMIC SUBSTANTIATION OF XML DATABASES COMPRESSION

Jakub Swacha
Institute of Information Technology in Management, University of Szczecin

Abstract. In this paper we provide the economic substantiation of XML data compression. We analyze the most effective XML compression methods (in terms of compression/decompression time and compression ratio) and show how to evaluate the benefits that can be gained from applying them. We demonstrate the achievable gains on examples.

Keywords. XML, data compression, XML compression algorithms, economic substantiation of data compression.

1. INTRODUCTION

In recent years, Extensible Markup Language (XML) became a format of choice for data interchange over the Internet, used in such applications as, e.g., Web Service Definition Language, Really Simple Syndication, or Synchronized Multimedia Integration Language [14]. Yet another kind of XML application is to use it as a format of offline documents, be it scientific databases, office documents, or other types of content. The benefits of doing so are many: a format that is self-describing, platform-independent, human-readable and can be easily processed.

However, with large data volume, the verbose nature of XML-encoded data becomes a nuisance. An obvious idea to counter this problem is data compression. XML data can easily be compressed using any general-purpose compression algorithm (see e.g., [7]), still, by applying an XML-specialized compression method, one can achieve not only higher compression ratios (see, e.g., [7]), but also obtain the ability to decompress selected structural elements without need to decompress the entire XML document (see, e.g., [8]).

Although the need for XML data compression can be substantiated on technical grounds, there is still question whether the gains of applying compression are worth the effort in economic terms. In this paper we are going to answer this question, by providing an economic model for XML data storage and usage, that takes into consideration various effects of applying data compression.

The paper is organized as follows. In the next section, we discuss the available methods capable of reducing the size of XML data. In section three, the possi-
ble economic consequences of applying compression to XML data are described, and the formulas to measure their value are proposed. Section four presents exemplary calculations showing achievable gains for different compression methods, storage devices, and value of user time. The final section concludes.

2. HANDLING XML VERBOSITY

There are many solutions for dealing with the verbosity of XML. Following the classification proposed by S. Sakr [5], we shall distinguish:

- general text compressors, i.e., implementing compression methods conceived for text (or, wider, general data), not especially for XML, still achieving satisfactory results when applied to XML documents;
- XML-conscious compressors, i.e., implementing compression methods conceived especially for XML, thus capable of taking advantage of specific XML traits and achieving, at least theoretically, superior compression results.

Among the general text compressors that are often used to compress XML data one can find widely used utilities such as zip, gzip, bzip2 and 7-zip. They employ general-purpose compression methods, such as Deflate (zip, gzip and 7-zip), LZMA (7-zip), Prediction by Partial Match (7-zip) or Burrows-Wheeler Transform (bzip2 and 7-zip) (a comprehensive explanation of these methods can be found in [10]). As XML documents are generally well-compressible, the compression ratios obtained using these compressors seem to be high – until compared to what can be attained using a specialized approach (see, e.g. [7]).

The XML-specialized compression methods are typically implemented as extensions of general purpose methods. They typically add some preprocessing steps that rearrange and transcode data, then apply a general purpose method, either to the entire preprocessed document, or parts of it.

One of the first XML-specialized compressors that gained popularity was XMill by Liefke and Suciu [4]. XMill makes use of three XML-conscious processing techniques. The first one splits the XML document into: element and attribute symbol names, text content, and document tree structure. Each of the resulting parts has different type of contents, so compressing them separately improves compression ratio. The second technique groups together contents of same XML elements, which helps compression methods with limited buffer, such as Deflate. The third, optional technique allows to apply a dedicated method for each type of data (such as numbers or dates), yet as it must be the user to guide XMill, which method should be applied to which container, it is hardly practical.

Originally XMill used only Deflate as the back-end compression method, later versions allow to replace it with more advanced methods: PPM or BWT. Still, the best relative compression improvement was measured for the Deflate.
The first XML-specialized compressor designed for the PPM-based back-end was XMLPPM by Cheney [2]. It applied techniques such as substituting element and attribute names with dictionary indices, removing closing tags marking their position (they can be reconstructed on decompression provided the document is well-formed), but the most important element of XMLPPM is ‘multiplexed hierarchical modeling’ consisting in encoding different kinds of data (element and attribute names, element structure, attribute values, element contents) with distinct PPM statistical models. Additionally, in order to exploit some correlation between different kinds of data, the previous symbol, regardless of the model it belongs to, is used as a context for the next symbol.

A modification of XMLPPM is SCMPPM by Adiego, de la Fuente and Navarro [1], in which every class of XML element is treated with a separate PPM model. This helps, but only in case of large XML documents, as the adaptive PPM model needs to process a significant number of symbols to become effective. A big flaw of SCMPPM is very high memory usage.

XBzip also uses PPM as back-end compression method, but only after applying a transform based on Burrows-Wheeler’s to represent XML document structure linearly using path-sorting and grouping [3]. XBzip has two work modes. The first one does not support queries over compressed data, but can attain compression ratios even higher than XMLPPM. The second, query-supporting mode, supported by XBzip Index utility, splits data into containers, and creates an FM-index (a compressed representation of a string that supports efficient substring searches) for each of them. For this reason, query processing times are very short, but storing the FM-index seriously decreases the compression ratios.

Recently, very high XML compression ratios were attained by XWRT [7]. As XWRT works as preprocessor, it can be combined theoretically with any general-purpose compression method, though the most promising results were obtained for LZMA — even though using PPM (or PAQ) helps achieve higher compression ratios, it significantly increases decompression time [9].

The preprocessing stage applies several transformations to the XML document. The main of them is the substitution of certain alphanumeric phrases with short identifiers. The substituted phrases include: ordinary words (that pass both the length and frequency thresholds), XML element start tags, URL’s, e-mail addresses, XML entities, and runs of spaces. Other transformations include succinct binary encoding of numbers, IP addresses, dates, and times, as well as replacing with special flags selected digrams and XML element end tags.

A modification of XWRT, QXT, offers lesser compression ratios for the sake of allowing partial decompression of the XML document and fast searching [8]. The main differences with XWRT are that QXT creates individual container for the contents of each XML element, and also packs data into blocks that can be decompressed separately.
3. ECONOMIC CONSEQUENCES OF APPLYING DATA COMPRESSION TO XML DATABASES

The most obvious outcome of applying data compression is the reduction of the volume of stored data. One can measure its economic consequences by multiplying the difference in storage media part taken up by the data before and after compression by the ownership cost of the storage media during the period of data storage (cf. [13]).

Another parameter that is affected by the application of data compression changes is the time, required to search, read or modify the compressed data. Similarly to the previous case, one can measure its economic consequences by multiplying the difference in time spent on accessing the data before and after compression by the value of the time unit. Although calculating the value of computer system time may become very complex (cf. [12]), we can make it much more practicable by restraining only to its most important component, that is the value of user time, and calculating it using the simple Time Savings Times Salary approach [6].

Applying data compression may have various other consequences (see, e.g., [11]). A well-substantiated approach for measuring data storage and compression costs and benefits [13] considers value of stored information, both a profit from its use and a loss from its unavailability. As in this paper we deal only with compression methods which are completely lossless, there is no change in the value of information, either stored or retrieved from the compressed form. Also, although in some circumstances compressing data may have some subtle effect on the risk of its loss (due to its smaller size), we shall also neglect it.

Taking into consideration the aforementioned observations, the following formulas are proposed:

\[ g_s = (l_o - l_c) \cdot c_s, \]  

where:
- \( g_s \) – gain due to reduced storage media usage for a single time unit (in monetary units),  
- \( l_o \) – original document length (in data storage capacity units),  
- \( l_c \) – compressed document length (in data storage capacity units),  
- \( c_s \) – ownership cost of storage media for a single data storage capacity unit and time unit (in monetary units per time unit);

\[ g_a = (t_o - t_c) \cdot c_i, \]  

where:
- \( g_a \) – gain due to reduced data access time for a single operation on data (in monetary units),  
- \( t_o \) – original data access time (in time units),  
- \( t_c \) – compressed data access time (in time units),  
- \( c_i \) – value of user time.
$t_c$ – compressed data access time (in time units),
$c_t$ – value of a single unit of time for the user (in monetary units);

g = g_s \cdot t_s + g_a \cdot a_d,$

where:

$g$ – overall gain due to applying compression during entire document storage period (in monetary units),
$g_s$ – gain due to reduced storage media usage for a single time unit (in monetary units),
$g_a$ – gain due to reduced data access time for a single operation on data (in monetary units),
$a_d$ – number of access operations during entire document storage period,
$t_s$ – time of data storage (in time units).

4. EXPERIMENTAL VERIFICATION

The economic consequences of applying data compression may vary depending on circumstances. In order to estimate the economic consequences of applying data compression to XML databases in possibly large number of cases, the following procedure has been established.

First, a number of experimental results has been gathered, which include compression ratios and query execution times for a number of XML compression methods. Instead of measuring them in an own test environment, results published earlier by this author [8] were used for this purpose.

Second, the following economic values were calculated: the gain due to reduced media usage (for different types of storage media), the gain due to reduced query execution time (for different numbers of queries), the overall economic effect of applying data compression (being the sum of the two previous values for a chosen set of parameters).

Table 1 contains compression results measured on the Lineitem test file [7], containing business order line items from the 10 MB version of the TPC-H database benchmark, for two general purpose compression methods (Deflate as implemented in gzip and LZMA as implemented in 7-zip) and seven XML-conscious compressors described in section 2. The “C.Ratio” row shows compression ratios in output bits per input characters, the “D.time” row shows decompression times of the entire XML document in seconds, and the “Q.time” row shows average query execution times in seconds; the three queries for which the average was calculated were (XPath notation): “/table/T/L_TAX”, “/table/T/L_COMMENT”, “/table/T/[L_COMMENT= "slowly"]”. For non-query-supporting compressors (all but XBZipIndex and QXT), the query execution times were obtained by summing up decompression time and query time on decompressed data using sgrep utility.
Table 1. XML compression test results.

<table>
<thead>
<tr>
<th>File</th>
<th>Deflate</th>
<th>LZMA</th>
<th>XMill</th>
<th>XML PPM</th>
<th>SCM PPM</th>
<th>XBzip</th>
<th>XBzip Index</th>
<th>QXT Deflate</th>
<th>QXT LZMA</th>
</tr>
</thead>
<tbody>
<tr>
<td>C.ratio</td>
<td>0.721</td>
<td>0.461</td>
<td>0.380</td>
<td>0.273</td>
<td><strong>0.244</strong></td>
<td>0.248</td>
<td>0.332</td>
<td>0.285</td>
<td>0.245</td>
</tr>
<tr>
<td>D.time</td>
<td><strong>0.203</strong></td>
<td>0.829</td>
<td>0.219</td>
<td>3.188</td>
<td>13.157</td>
<td>4.512</td>
<td>6.259</td>
<td>1.120</td>
<td>1.340</td>
</tr>
<tr>
<td>Q.time</td>
<td>0.870</td>
<td>1.496</td>
<td>0.886</td>
<td>3.855</td>
<td>13.824</td>
<td>5.179</td>
<td><strong>0.055</strong></td>
<td>0.097</td>
<td>0.099</td>
</tr>
</tbody>
</table>

Average query time on uncompressed data was 0.667. Source: [8].

The gain due to reduced media usage has been calculated using formula (1) for the following types of storage: Solid State Drive (MLC Flash), Hard Disk Drive (7200 rpm), Hard Disk Drive (15000 rpm). For each of these device types, the lowest cost per megabyte found using an Internet price comparison tool on 2010-09-22 has been assumed as purchase cost, and the cost of ownership during device lifetime has been estimated at four times the purchase cost. Figure 1 shows the gain in Polish zlotys achieved per gigabyte of stored XML data.

![Figure 1. Gain due to reduced storage media usage after compression.](image)

Although there are significant differences in compression ratios achieved by different methods, the economic outcome of applying them, in terms of storage space savings, varies by less than 10%. As could be expected, there are huge diff-
ferences depending what type of storage has been used. Applying compression to
data stored on cheap 7200 rpm SATA disks seem to be least substantiated, taking
into consideration only the storage space savings.

The gain due to reduced query execution time (per single query) has been cal-
culated using formula (2) only for Hard Disk Drive (7200 rpm), as only such mea-
surements were available from the quoted source. Figure 2 shows relation between
average user time value and the achieved gain (or loss).

As one can see, only the query-supporting XML compressors are economical-
ly substantiated if just the value of data access time is considered. Both versions of
QXT (shown as a single line due to minuscule differences between the Deflate and
LZMA modes) and XBzip Index provide very similar savings. Applying the non-
query-supporting PPM-based methods may result in high costs due to significant
increase of data access delays.

The overall economic effect of applying data compression has also been cal-
culated only for Hard Disk Drive (7200 rpm), assuming the size of database at
100 GB. Figure 3 shows relation between total user time value (for all the data
access operations during entire device lifetime) and the overall gain (or loss)
achieved by applying compression to XML database in Polish zlotys.
In case of cheap SATA drives, the savings due to reduced database size dominate over the effect of compression on access time only for very rarely accessed databases (or, assuming a very low value of time). For the numbers shown on the figure, they play insignificant role, and, as a result, the two query-supporting compressors (XBzip Index and QXT) once again seem to be the best choice.

5. CONCLUSIONS

Applying data compression to XML databases results in measurable economic effect, coming from reduced storage capacity requirements and modified data access time (and resulting work delays). The effect of the former is always positive (provided the compression is effective), yet its value is very small, unless very expensive storage media are used, or the size of the compressed database is huge. The effect of the latter can be positive only if queries over compressed data may be executed without decompressing the entire XML file. Applying compression methods that offer this functionality, such as QXT or XBzip Index, may produce significant savings.
REFERENCES


INDEXING XML DATA AND THE PERFORMANCE OF XQUERY IN RELATIONAL DATABASE BASED ON SQL SERVER 2008

Marek J. Śmiateński
Faculty of Mathematics and Computer Science, University of Lodz (UL)

Abstract. In the paper we describe issues concerning an effective usage of the XML indexes in processing XML data stored in relational database. One of the most important tasks of designing an efficient database is a selection of appropriate indexes. Because well-known commercial DBMSs strongly support features related to the XML and relational data integration, indexing XML data is relevant in terms of efficiency of XQuery queries. We consider strategy of choice of XML indexes based on Microsoft SQL Server 2008. Analysis focuses on the secondary XML indexes and full-text index on an XML column, as their proper usage may bring significant benefits. The performance test results of XQuery queries executed on some relational database containing weather XML data are reported at the end of paper.

Keywords. Relational database, XML, data indexing, queries, XQuery, SQL Server.

1. INTRODUCTION

In recent years, a support for XML data was included to the most commonly used relational and object-relational database management systems, just to mention IBM DB2, MS SQL Server and Oracle, therefore they are called as XML-enabled database systems (XDBMS). In this way a costly transformation XML data into relational and vice versa is possible to avoid (as a matter of fact, an effective data mapping is not always suitable e.g. when XML data is document-centric or has recursive structure), what gives the database users indisputable benefits together with increasingly important built-in capabilities and relatively easy integration XML and relational data. Also we must not forget the significance of XML document order, which is not ensured in the relational database, as well as a requirement of well-formedness and/or validity of data, of which the automatic guarantee brings an additional inefficiency of shredding approach.

One of the major tasks of the database administrator is to define indexes on the design phase. When the XML data are stored natively in a relational database, XML indexing is recommended regardless of whether the data is content-centric or document-centric, or whether it is fine-grained or coarse-grained. Naturally, an
indexing mechanism is necessary to speed up the execution of queries, which select only XML data or integrate XML and relational data. Therefore, key issues, which must be considered during the selection the best index set for a database containing XML and relational data, are:

- Should be an XML column included in the relational index, or no?
- How to create XML indexes?

The first issue is relatively simple to resolve: when XML data are often searched and rarely modified, then including XML column in the relational index definitely brings positive outcomes. We derive an especially large advantage for queries combining XML and relational data.

In this paper we discuss issues related to the indexing XML data based on the relational DBMS Microsoft SQL Server, in particularly we try to answer the second question. The rest of the paper is organized as follows. Section 2 provides a short and informal background material on the indexing XML data. Section 3 gives a detailed overview of capacities of indexing XML data in Microsoft SQL Server 2008. Theoretical strategies of indexing XML data for SQL Server are analyzed in Section 4. The experimental results are presented in Section 5, the main part of this paper. Finally, the paper concludes with a summary in Section 6.

2. INDEXING XML DATA

Index is a data structure, which makes it possible to increase the speed of data operations. A proper choice of index in relational database allows significantly to reduce the access time to the processed information. On account of processing XML data, a very important functionality of relational database servers is the ability to indexing such native data. As is well known, one of typical approach to XML data storage in a relational DBMS is decomposition XML instance into relational tables. However, XML data is not always sufficiently wieldy to be able to use full power of the relational tools to process it. Then, the storing XML instance in XML type column as large binary objects (BLOB) is considered as an attractive alternative. An XML indexing mechanism is designed to speed up such queries, in which just native XML data is processed, particularly when:

- a large amount of XML data is stored in a database (i.e. either there are several rows with XML data or XML instances in each row are relatively large in size) and relatively small data is often queried simultaneously;
- most queries are performed on several rows with XML data;
- single SQL query contains repeated XQuery expressions, which require a multiple parsing of XML instance.
A natural way of indexing XML data is index based on B+-tree (similarly as in relational databases). Reassembling the result of XML type from B+-tree allows to preserve data fidelity in terms of document order and document structure based on hierarchical relationships between XML nodes.

In the world of relational database is well-known so-called Index Selection Problem (ISP). ISP solution for a relational DBMS should give an optimal index configuration from the point of view of query and update operations. However, the ISP belongs to the class of NP-complete problems and is computationally difficult, therefore implemented tools determine a solution of problem using some ranking heuristics. In commercial database systems we can find various methods for suggesting the optimal index configuration, e.g. linear programming algorithms (in fact, 0-1 integer programming problem) in IBM DB2 Advisor [6]) or genetic algorithms. The first time, application of ISP in XDBMS was shown in [2], where the relational definition was properly adapted and the key-oriented XML index (KeyX) was introduced. KeyX uses values of elements or attributes, which are accessed by specified XPath path expressions. However, in some XML enabled database systems (in particular in SQL Server) such approach is not possible to apply, because XML index can be created only on XML columns.

3. XML INDEXES IN SQL SERVER 2008

Although the first capabilities of handling XML data appear in SQL Server 7.0, mechanism for indexing XML data was involved in version 2005. It was natural consequence of the introduction of a native XML data type, which allows to create in relational table one or more columns containing XML documents or fragments stored as BLOB objects up to 2GB in size. SQL Server 2008 (just like preceding version) offers two kinds of XML indexes: one primary and three secondary.

An existence of a primary index on XML column eliminates an overhead, which is caused by the need to shred XML data into relational when every XQuery query is performed (because SQL Server makes preshredding BLOBs at run time). The primary index is used when the engine evaluates an every query except of retrieving the full XML document (then document is retrieved faster from the internal representation). On the primary XML index you can create one or more secondary XML indexes, which may increase performance of some kinds of queries. Secondary indexes provide additional information for query engine to facilitate choice of execution plan. Generally, index selection depends on a type of queries (path-based, value-based queries and property bag scenarios) [1], whose performance might be improved.
To create XML index on XML type column in SQL Server we use a modified form of the CREATE INDEX statement of Transact-SQL DDL. A general abridged syntax is as follows:

```
CREATE [PRIMARY] XML INDEX index_name
ON table (xml_column_name)
[USING XML INDEX xml_index_name
[FOR {VALUE|PATH|PROPERTY}]]
```

where the **USING** clause is applied, when we create the secondary XML index, then the **FOR** clause determines type of such index. Additionally, we can also use nine options using the additional clause **WITH** [5].

### 3.1. Primary XML index

As we mentioned above, the primary XML index in SQL Server is B⁺-tree containing the shredded representation of XML data, which is conformed to XML Information Set items. Nodes are labeled through a hierarchical labeling system ORDPATH [3], which allows capturing the structural fidelity of XML data without access to XML schema, only within a single column of the primary XML index. For example, subtree of a given XML node can be retrieved from index using a range scan over the OrdpPath values of this node and its descendants.

If the primary XML index exists, then query optimizer decides how to carry on the XQuery expression included in SQL query, choosing between:
- **top-down execution** - rows of the base table (table, which contains targeted XML column) are processed before the primary XML index,
- **bottom-up execution** - the primary XML index is scanned before back-joins with the base table.

Structure of the primary XML index was described in details e.g. in [4]. Here, we mention only that the number of rows in the primary index is approximately equal to number of nodes in XML blob, and every row contains the following information about the node: tag name (an element or attribute name), node value, node type, document order information, path from each node to the root of the XML tree and primary key of the base table (total 11 columns). Hence, the primary XML index is larger in size than the text form of XML instance and unfortunately contains some redundancy (e.g. the primary key of the table is repeated in all rows for an XML instance).

The primary XML index requires that the table containing the indexed column must have a clustered primary key defined on it. It causes that the primary key cannot be modified as long as the XML index exists. In contrast to the relational index, the XML index must be created on a single XML type column (however it does not matter whether a given column is typed or untyped) and there is no possibility of creating the XML index on a computed column. On a given XML column we can create only one XML primary index and simultaneously many XML secondary indexes.
3.2. Secondary XML indexes

XQuery query performance can be additionally optimized in many cases by the secondary XML indexes: PATH, VALUE and PROPERTY. No secondary index cannot be created on an XML column, if the primary XML index does not exist on it.

The PATH secondary index is used for XQuery queries based on path expressions performed on XML type columns especially when predicates containing the comparison operators are specified (e.g. in the query method). This type of index was pointed out in [1] as particularly useful, when the exist method is used in the WHERE clause of the base SQL query.

The PROPERTY index is designed to optimize the queries, in which from single XML instances the scalar values are selected based on their properties such as paths and known primary key.

The VALUE index can speed up query when a node value is important, and a path expression is not precise i.e. name or exact location of the node are not known. In other words, it refers especially to queries, in which the wildcard character in a path expression is used and queries with a specified node value in a predicate and the descendant-or-self axis (so-called descending search axis).

The secondary XML indexes improve performance of queries in the bottom-up approach regardless of their type. Therefore, an interesting issue is the analysis of the XML indexing strategy based on the class of queries, which efficiency can be increased by the appropriate choice of the secondary XML indexes. Further discussion on this topic is in Section 4.

3.3. Full-text index

In SQL Server it is also possible to create the full-text index on XML column (in the same way as on relational column). Then the content of the XML data is indexed and the XML markup is ignored (so XML tags and attributes are not indexed). Because the full-text search returns the whole XML instance (instead of only a fragment containing the requested data), then in practice it is usually used combination the full-text search index with the XML index. Such approach allows to filter the XML data using the SQL queries (e.g. with predicates CONTAINS or FREETEXT), what is particularly useful for document-centric data. Such queries can use the XML methods value and query, however the SQL predicates have a higher priority than XQuery expressions regardless of the form of query. From the point of view of XML data processing, it is worth to mention that full-text search in SQL Server 2008 is available completely in 51 languages.
4. STRATEGIES OF INDEXING XML DATA

A very important disadvantage of the XML index is its size, which unfortunately significantly exceeds the size of the XML data stored in the indexed column. How painful can be this effect, it largely depends on the structure of the XML instance. Generally, a relatively small number of tags even with a large amounts of XML data makes that size of the index only slightly exceeds the size of data, otherwise the size of index can be even several times larger. Using knowledge of the requested data, it is possible to limit the size of the XML index. It is achievable in SQL Server by selection of an appropriate configuration of the secondary indexes. So we have a similar situation as in the relational databases, where it is not advisable to create indexes on all available combinations of columns of the table. Equally important is a fact, that benefits achieved by use of the XML indexes, which were created for specific queries, can be reduced by a weakness performance the DML operations (SQL DML as well as XML DML). It implies that type of database has to be one of the criteria for use the secondary XML indexes (so, it is particularly recommended in the analytical databases).

Clearly, the easiest solution is to create the primary and all secondary XML indexes and to leave a choice to query optimizer. But we cannot forget, that the secondary indexes are non-clustered, what in turn causes a very large disk space consumption, e.g. when there is several rows with XML data or XML instances are large in size. Moreover, the maintenance of the secondary indexes is very expensive, especially when the both XML and relational data are often modified (vide the operational databases). Similarly, the XML data modification related to the insert or delete of nodes, which are in the sibling order, may significantly increase the cost of the primary XML index usage.

However, note that on the design phase we have to make the appropriate initial index choice, because even having a tool, which automates the creation and modification of indexes, a noticeable results would be felt only after some time (based on a many operations performed in database). It is easy to notice, that the creating the primary XML index must be an administrative decision, while the selection of the secondary XML indexes requires an analysis of statistics, since the choice should be based on cost.

In fact, a serious problem of selection of the secondary indexes occurs only if there is more than one XML type column in the relational table. Otherwise, the situation is quite obvious, because then we have only one indexable XML column, for which there are seven admissible configurations of the secondary indexes.

Strategies of the XML indexes creation on the design phase of the relational database, in which we perform an integration of the XML and relational data, can be briefly described as the following rules resulting from the assumptions of implementation (see e.g. [1, 4, 5]):
(i) we create the primary XML index from the assumption, because its existence almost always improves a query performance, and its absence prevents the creation the secondary indexes. An exception is if we know *a priori* that the most of queries will scan all rows of the primary index;

(ii) if one type of XQuery query is used more often than other, and there is an appropriate secondary XML index, then we should consider using this index;

(iii) we create the secondary XML index *PATH*, when queries often contain path expressions in the *query* or *exist* methods (path-based queries);

(iv) we use the secondary XML index *PROPERTY*, when we mostly perform queries with the *value* method and a value of the primary key is known (property bag scenario);

(v) we use the secondary XML index *VALUE*, when the XML data stored in the database has jagged hierarchy or when there are ambiguous paths in the queries;

(vi) we apply XML indexing instead of the full-text one, when the important data or a lot of data can be found in attributes (value-based queries).

Generally, when we work with database, we should analyze statistics on performed XQuery queries and respond ad hoc to anomalies, which may be noticeable, when an XML query set will change significantly. Therefore from the theoretical point of view it is worth to consider the introduction of metrics, which would allow to rate what the configuration of the secondary XML indexes is most optimal for a given query set. A detailed discussion of techniques for cost evaluation of index configuration and optimization of index selection process is beyond the scope of this paper.

5. EXPERIMENTAL RESULTS

In this section we present an experimental study conducted using XQuery queries processing XML data, which was indexed in different ways. The test XML data comes from National Digital Forecast Database provided by U.S. National Oceanic and Atmospheric Administration [7]. Our relational database contains three tables: basic information about U.S. states (*states*), weather stations (*stations*) and detailed information about observed weather conditions (*measurements*). The measurement results are stored as original XML documents in the untyped XML type column (*data*), which is only indexable XML column in database. Queries were performed on three databases, which differ only in the number of rows in the table *measurements* i.e. 400 (1.49GB), 800 (2.98GB) and almost 1200 thousands (4.37GB).

All tests were carried on the computer with Intel Core 2 Duo clocked at 3GHz with 4GB RAM under Windows Server 2008. Owing to space limitations, we pre-
sent the results of selected tests using only the CPU execution time without indexes (in seconds) and some metric which represents benefits of using the XML indexes as a time profit relative to the execution time without any XML indexes (i.e. using XML blobs)

\[
\text{rtp} = \frac{\text{exec time using XML blob} - \text{exec time using XML index}}{\text{exec time using XML blob}}
\]  

(1)

In fact, rtp quantifies a relative gain of execution time, while the rtp sign determines loss or profit. All results are summarized in Table 1 (bold indicates the best index configurations).

<table>
<thead>
<tr>
<th>Query</th>
<th>rows</th>
<th>CPU time</th>
<th>primary index</th>
<th>secondary XML indexes</th>
<th>secondary XML index combinations</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Path</td>
<td>pRop</td>
</tr>
<tr>
<td>Q1a</td>
<td>400</td>
<td>0.86</td>
<td>-0.124</td>
<td>-0.126</td>
<td>-0.343</td>
</tr>
<tr>
<td></td>
<td>800</td>
<td>1.72</td>
<td>-0.018</td>
<td>-0.136</td>
<td>0.001</td>
</tr>
<tr>
<td></td>
<td>1200</td>
<td>2.81</td>
<td>0.027</td>
<td>0.005</td>
<td>0.055</td>
</tr>
<tr>
<td>Q1b</td>
<td>400</td>
<td>37.46</td>
<td>-0.514</td>
<td>0.038</td>
<td>0.255</td>
</tr>
<tr>
<td></td>
<td>800</td>
<td>74.08</td>
<td>-0.703</td>
<td>0.048</td>
<td>0.283</td>
</tr>
<tr>
<td></td>
<td>1200</td>
<td>107.80</td>
<td>-0.799</td>
<td>0.046</td>
<td>0.265</td>
</tr>
<tr>
<td>Q1c</td>
<td>400</td>
<td>64.09</td>
<td>0.201</td>
<td>0.076</td>
<td>0.201</td>
</tr>
<tr>
<td></td>
<td>800</td>
<td>127.23</td>
<td>0.173</td>
<td>0.127</td>
<td>0.191</td>
</tr>
<tr>
<td></td>
<td>1200</td>
<td>188.75</td>
<td>0.201</td>
<td>0.158</td>
<td>0.205</td>
</tr>
<tr>
<td>Q2</td>
<td>400</td>
<td>279.09</td>
<td>0.415</td>
<td>0.422</td>
<td>0.408</td>
</tr>
<tr>
<td></td>
<td>800</td>
<td>553.46</td>
<td>0.412</td>
<td>0.424</td>
<td>0.415</td>
</tr>
<tr>
<td></td>
<td>1200</td>
<td>811.24</td>
<td>0.418</td>
<td>0.421</td>
<td>0.413</td>
</tr>
<tr>
<td>Q3a</td>
<td>400</td>
<td>58.84</td>
<td>0.421</td>
<td>0.391</td>
<td>0.232</td>
</tr>
<tr>
<td></td>
<td>800</td>
<td>120.66</td>
<td>0.450</td>
<td>0.402</td>
<td>0.269</td>
</tr>
<tr>
<td></td>
<td>1200</td>
<td>172.22</td>
<td>0.439</td>
<td>0.382</td>
<td>0.263</td>
</tr>
<tr>
<td>Q3b</td>
<td>400</td>
<td>59.90</td>
<td>0.378</td>
<td>0.390</td>
<td>0.216</td>
</tr>
<tr>
<td></td>
<td>800</td>
<td>122.26</td>
<td>0.398</td>
<td>0.387</td>
<td>0.221</td>
</tr>
<tr>
<td></td>
<td>1200</td>
<td>176.16</td>
<td>0.387</td>
<td>0.386</td>
<td>0.216</td>
</tr>
<tr>
<td>Q4a</td>
<td>400</td>
<td>39.47</td>
<td>0.349</td>
<td>0.351</td>
<td>0.337</td>
</tr>
<tr>
<td></td>
<td>800</td>
<td>79.87</td>
<td>0.362</td>
<td>0.360</td>
<td>0.356</td>
</tr>
<tr>
<td></td>
<td>1200</td>
<td>120.45</td>
<td>0.351</td>
<td>0.358</td>
<td>0.353</td>
</tr>
<tr>
<td>Q4b</td>
<td>400</td>
<td>59.16</td>
<td>0.174</td>
<td>0.161</td>
<td>0.179</td>
</tr>
<tr>
<td></td>
<td>800</td>
<td>117.41</td>
<td>0.163</td>
<td>0.159</td>
<td>0.163</td>
</tr>
<tr>
<td></td>
<td>1200</td>
<td>176.50</td>
<td>0.177</td>
<td>0.156</td>
<td>0.159</td>
</tr>
<tr>
<td>Q5</td>
<td>400</td>
<td>67.40</td>
<td>-0.213</td>
<td>0.108</td>
<td>0.068</td>
</tr>
<tr>
<td></td>
<td>800</td>
<td>136.70</td>
<td>-0.221</td>
<td>0.129</td>
<td>0.070</td>
</tr>
<tr>
<td></td>
<td>1200</td>
<td>199.85</td>
<td>-0.275</td>
<td>0.122</td>
<td>0.070</td>
</tr>
<tr>
<td>Q6</td>
<td>400</td>
<td>90.78</td>
<td>0.108</td>
<td>0.164</td>
<td>0.145</td>
</tr>
<tr>
<td></td>
<td>800</td>
<td>181.82</td>
<td>0.129</td>
<td>0.168</td>
<td>0.169</td>
</tr>
<tr>
<td></td>
<td>1200</td>
<td>267.18</td>
<td>0.122</td>
<td>0.186</td>
<td>0.181</td>
</tr>
<tr>
<td>Q7</td>
<td>400</td>
<td>66.15</td>
<td>0.444</td>
<td>0.397</td>
<td>0.309</td>
</tr>
<tr>
<td></td>
<td>800</td>
<td>132.60</td>
<td>0.441</td>
<td>0.412</td>
<td>0.271</td>
</tr>
<tr>
<td></td>
<td>1200</td>
<td>191.34</td>
<td>0.434</td>
<td>0.410</td>
<td>0.315</td>
</tr>
</tbody>
</table>

Source: Own preparation based on performed tests.
It seems that despite the ambiguity of results (e.g. anomalies for Q1a), the theoretical assumptions of the index implementation described in Subsection 3.2 and Section 4 have not been partially confirmed in practice. For example, it happens that the primary index slows the query (Q1a, Q1b, Q5), and the VALUE index is not best for queries with axis descendant-or-self in path expression (Q3b, Q7). In turn, the index configuration based on all three secondary indexes gives the smallest gain in almost all tested queries (except of full-text search in Q4), and sometimes brings up even a big loss (Q6). However, it does not mean that the index creation according to the aforementioned rules will not bring positive results. In fact, the index configuration depends strongly on the structure of the processed XML documents as well as usually performed queries. On the other hand, there are queries, whose performance can be significantly improved using almost every index configuration, e.g. full-text search, especially when the exact path is known. Besides, large profits for single secondary indexes do not give necessarily a similar profit for their combination (see Q2 and index configuration PATH+PROPERTY).

Moreover, based on the omitted results we can conclude, that relative increases in consumption of disk space caused the index size are comparable for three test databases (about 2.63 times) and had no effect on any statistics. Next, regardless of the XML index configuration, the CPU execution time of query and the number of logical reads depend approximately linearly on the number of the stored rows. The index configuration affects on buffer hit ratio $bhr$ (indicating the efficient performance of I/O operations) only in rare cases (some configurations for Q1b, Q1c, Q3), although ratio is higher for queries performed on the non-indexed XML data (sometimes minimal, but for all queries). The $bhr$ was less than 92% besides only for Q1b and two combinations of the secondary indexes. With a full insight into results, it is noticeable that the indexing allows often to achieve a much smaller benefit, when the number of rows actually processed by the XQuery expression is the same or close to the number of all rows in a source table (as for Q5, Q6), but it is not necessarily the only cause. So, it is surprising that this obvious weakness of the gains of indexing is usually omitted or underestimated.

For completeness of the presentation, we briefly describe below some queries, which are tested in studies and discussed in this paper.

**Query 1** (selection): Data selection with a comparison operator implemented in various ways (SQL, XQuery and SQL with XQuery):

```xml
<query>'<pomiar>
  <temp>({data(/observation/temp_f)})</temp>
  <wiatr>({data(/observation/wind_mph)})</wiatr>
</pomiar>'
```
The same XQuery code above was embed in three queries to filter data using
(a) SQL clause WHERE with condition on the relational column
   WHERE STATION= 'KSAC'
(b) XQuery conditional expression and XML column
   if (/observation/station_id="KSAC") then ... else ...
(c) SQL clause WHERE and the exist method on XPath expression with predicate
   WHERE DATA.exist('/observation/station_id[.=‘KSAC’])=1

Approaches (a) and (c) return only measurements from a requested station, while (b) - all source rows with Null corresponding measurement from other station.

Query 2 (computation): Computations performed on numerical data stored in the XML column as the element value of the XML instance:

```xml
query('if (data(/observation/wind_mph)!="" and data(/observation/wind_mph)!="NA")
   then <data>
      <wind v="kmh">{data(/observation/wind_mph)[1]*1.609}</wind>
      <wind v="ms">{data(/observation/wind_mph)[1]*0.447}</wind>
   </data>
   else ()')
```

Query 3 (aggregation): Counting the XML documents containing a specified element using the exist method in two ways: (a) using full path to a requested element, (b) descending search axis:

```sql
SELECT COUNT(DATA.query('data(/observation.visibility_mi)'))
FROM XML.measurements
WHERE DATA.exist('/observation/visibility_mi')=1
```

Query 4 (full-text search): Full-text search XML documents in the exist method (in two ways as Q3):

```sql
SELECT DATA.query('data(/location),data(/observation_time_rfc822)')
FROM XML.measurements
WHERE DATA.exist('//text()[contains(.,"gusting")]=1
```

Query 5 (data integration): Integration of the XML and relational data stored in different columns of the same table:

```sql
SELECT DATA.query('<Station StationID="{sql:column("Station")}"'>
   (/ observation/location/text())
</Station>')</nFROM XML.measurements
Query 6 (join): Joining data from different tables with simultaneous integration of the XML and relational data:

```
SELECT DATA.query('<Measure>
    <Station>{sql:column("a.station_name")}</Station>
    <Date>{data(/observation/observation_time_rfc822)}</Date>
    <Temp>{data(/observation/temperature_string)}</Temp>
</Measure>')
FROM XML.Stations a, XML.Measurements b
WHERE a.STATION_ID=b.STATION
```

Query 7 (sorting): Sorting of the unique text data using the value method and XQuery functions on strings:

```
SELECT DISTINCT DATA.value('substring((//ob_url)[1],36,12)','varchar(12)') as Result
FROM XML.measurements
WHERE DATA.exist('//ob-url')=1
ORDER BY Result
```

6. CONCLUSION

XQuery is a query language for XML structures, which allows user-friendly way to create almost any query to database containing both usual relational data and native XML data. In order to improve query performance in SQL Server it was implemented the XML indexing mechanism based on B+-tree. The study shows that in practice a noticeable increase in speed of performed queries can be obtained by proper choice of the XML indexes, especially for processing large and complex XML documents, however an extreme gains have not been confirmed as in [4]. In general, the XML indexing in SQL Server does not offer a very significant improvement in efficiency and it is not difficult to find queries, for which the indexing gives too poor benefits and too much consumption of disk space. Moreover, the selection of the best XML index configuration may be difficult if relatively complex XPath expressions are used in queries.

REFERENCES


ONTOLOGIES IN INTEGRATION OF WEBSITES EVALUATION METHODS

Paweł Ziemba, Ryszard Budziński
Faculty of Computer Science, Westpomeranian University of Technology

Abstract. The article deals with issues regarding the integration of methods and criteria used by evaluation of quality of different kind of websites. Many different websites quality evaluation methods use various criteria for the evaluation. In addition, some of them use criteria which have different names, but deal with the same quality dimension. In that case, drawing up the methodology of linking different evaluation models will allow to get the transparent structure of quality indicators. The authors are indicating bases for applying ontology for integration of diverse assessment methods into a homogeneous structure. In the future, the structure received in this way can be used for creating a new method of quality evaluation of diverse websites.

Keywords. Ontologies, websites quality, eQual, Technology acceptance model.

1. INTRODUCTION

The popularity of websites is dependent on their quality and usefulness. It is important for such websites to be attractive, interesting, to attract attention of users - both private and entrepreneurs being potential investors. One of the best ways to increase websites attractiveness for the users and to adapt them to the needs of the visitors is to examine their quality. Examining the quality contributes to the permanent development of a website and to maximum use of resources invested in a website. It is possible because the research on the quality gives instructions to the website designers what changes should be made, so that a website meets users expectations, which are often difficult to express. There are many publications, in which authors are making attempts to formalize evaluation methods for different kind of websites. To obtain evaluations of individual websites, i.e. following methods are used: questionnaire surveys [1], expert evaluation [2], or tracking users activity [3, 4].

Evaluation methods are usually adapted for websites of a given kind, putting emphasis on particularly important elements in the given kind of websites. Basic differences between methodologies appear in evaluation criteria and their structure, but they often also appear in marking scale, importance of criteria, or also in meth-
odology of acquiring the evaluation marks. There are many websites quality evaluation methods and each of them is using diverse evaluation criteria, in addition some of the methods are using criteria having different names, but concerning the same quality dimensions. Similarly, while creating new evaluation methods, the authors often use existing models of website quality evaluation, choosing evaluation criteria which seem to be suitable for their method. The selection of the evaluation criteria is performed by authors usually in a not-formalized form, being based on literature analysis and their own reflections. It seems that it is not the best way to create a new website quality evaluation model, because such an operation is chaotic in its nature. In this paper a hypothesis is being tested, that applying ontology will allow to create a homogeneous and transparent structure of the quality rates which in the future can serve for creating a new websites quality evaluation method. Creating such a structure is supposed to facilitate the process of selecting the criteria used in evaluation of websites of different kind.

2. WEBSITES QUALITY EVALUATION METHODS

Out of many existing website quality evaluation methods, for the conducted examinations two models, i.e.: eQual and the Ahn model were chosen. This choice is justified, because certain resemblances exist between the methods mentioned above, i.e. in some evaluation criteria and in marking scale - both methods are using the Likert scale for the evaluation.

The eQual method was created based on the Quality Function Deployment which provides means of identification and entering the opinion of users on the quality of product at consecutive stages of forming it. Originally this model used 24 survey questions reflecting values of 8 used criteria in the model. Criteria were grouped together in 4 categories [5]. Furthermore, to the existing criteria, some new criteria focusing on interaction between user and website were added. Originally, the added criteria were borrowed from the SERVQUAL scale [6]. In the next version this method, the grouping of criteria and survey questions were changed. In the end, a list of 22 theorems and 5 criteria was obtained and they were groups into 3 categories: (1) usability, (2) design, (3) empathy, (4) trust, (5) information. Polled users evaluate the degree of fitting of every theorem to reality, applying the spot scale with values ranging from 1 to 7. The methodology of obtaining the opinion on the website, apart from getting the value of assessments of individual subcriteria, also lets get the importance of each of the theorems, importances also being ranked in the range 1-7. Apart from giving points to each of the theorems, respondents are also giving an overall view of a website. Based on this evaluation, a credibility of fragmentary opinions of every user is being verified [7].

The T. Ahn method was studied using the Technology Acceptance Model which explains and predicts user acceptance of an information system. The major contribution of TAM to measures development is the identification of two key
beliefs: perceived usefulness and perceived ease of use [8]. The first version of the Ahn method was supposed to study influence of trusting the Internet banking on their acceptance by users. In its all quality dimensions, the criteria coming from the TAM model were distinguished (perceived usefulness and perceived ease of use, attitude toward using and behavioral intention to use) and a criterion of the trust was additionally introduced [9]. In the course of work on the Ahn method, a primitive model was extended there against next elements, among others elements important on account of the online shops websites quality: services, information, system, product and his delivery [10]. The conclusive model contains 8 quality dimensions: (1) system quality, (2) information quality, (3) service quality, (4) playfulness, (5) perceived ease of use, (6) perceived usefulness, (7) attitude and (8) behavioral intention to use. It includes 54 survey questions belonging to different criteria of the quality. Similarly as in the eQual method, users evaluate here the degree of fitting of each theorem to reality, applying the Likert scale with values ranging from 1 to 7 [11]. In this method, the importances of individual criteria and questions are not defined, so all of them are equally essential.

Certain resemblances appear between the two discussed methods, so to merge them seems to be rational. However, in order for the merging process not to be conducted chaotically, for each method a notional was created, that aims at explaining the structure of individual evaluation methods. These models were drawn up in the form of an ontology.

3. BUILDING ONTOLOGY FOR WEBSITES QUALITY EVALUATION METHODS

The most often quoted definition of ontology tells that it is "specification of conceptualization", or "formal specification of shared conceptualization" [12]. Other definition talks about the fact that in the computer science ontology is treated as a data structure, a data representation tool to share and reuse knowledge between Artificial Intelligence systems which share a common vocabulary [13]. This definition justifies using ontology in integration of structures criteria and websites quality evaluation methods.

Ontologies for the methods discussed in the chapter two were built using the Protege 2000 editor. This editor was created at the Stanford University and implement in the Java language. What is important, it is an OpenSource project and it is available free of charge, so no costs are involved acquiring it [14]. Thanks to using the Java language, Protege is a platform-independent environment for creating knowledge bases. A knowledge-base is represented as a frame-based ontology consisting of classes and slots. A graphical user interface (GUI) is provided to develop and maintain the hierarchy of classes and to create and assign slots to them (corresponding to class attributes or binary relationships with other classes) [15].

Parts of ontologies which were created for eQual and Ahn methods were included in figures 1 and 2.
Figure 1. Fragment of ontology built for the eQual method. Source: own study.

Figure 2. Fragment of ontology built for the Ahn method. Source: own study

The graphic outlines introduced in figures 1 and 2 do not contain full ontologies in order to keep the legibility of these schemes. In each of these ontologies, the Quality class is the highest in the hierarchy. The Quality class is connected through appropriate slots with subclasses determining individual elements of the quality. On the bottom stair of the hierarchy of classes isolated theorems are included, appearing in used questionnaire forms in studied methods. These theorems are connected with the evaluation class, instances of which contain values assigned by users to individual theorems (in Likert scale). In the evaluation class a difference between methods is visible. For the eQual method this class contains two slots which refer to: evaluation value of the given theorem, importance assigned by the user to this theorem. The evaluation class in the Ahn method contains only a slot referring to evaluation value of the theorem, whereas the importances of criteria are not appearing in this method. Slots, in other words relations, with which classes were tied together are: part_of (whole-part) and has (object-feature). These relations were borrowed from the semantic networks theory [16].

Full hierarchies of classes for ontologies created for presenting studied quality evaluation methods were included on figures 3 and 4. In these hierarchies one can...
see, that created ontologies are exactly reflecting structures of the criteria and theorems appearing in studied methods. One can see differences between forms of these structures in studied methods. The eQual method contains three groups of criteria in which five criteria and 22 theorems are included. The Ahn method consists of 8 criteria containing 54 theorems.

After building ontologies for the shown methods one should merge them.
4. INTEGRATION OF THE AHN AND EQUAL ONTOLOGIES

Integration of ontologies is defined differently in different publications. Integration can be understood e.g. as creating a new ontology from existing one by specializing or adapting it, or by extending its fragments [17]. Other definition is saying that integration consists in creating a new ontology from two or more existing ontologies with overlapping parts, which can be either virtual or physical [18]. This definition refers not only for integration but also for merging of ontologies [19]. In that case in this article, notions “integration” and “merging” they are being used interchangeably. The algorithm of merging ontologies applied at this work consists of three steps:

- finding the points of intersection (similarities and/or differences) of the ontologies,
- associating semantically similar concepts through equivalence and subsumption relationships,
- checking of the results in order to obtain a consistent, coherent and nonredundant ontology [18, 20].

Seeking intersections between ontologies can be made by applying parallelly: analysis of names (string-based technique), analysis of the structure (graph analysis) and semantic analysis [21]. Such methodology was applied in this work.

Based on hierarchies of classes included in figures 3 and 4 it is easy to notice that four theorems included in the Information_Quality criterion of the Ahn method are in accordance with theorems appearing in the same criterion of the eQual method. Furthermore, the reliable (Ahn) and believable (eQual) criteria are the same. Also certain inconsistencies are appearing in these hierarchies between methods with regard to the theorems structure:

- theorem 1 from System_Quality criterion in the Ahn method, is included in Design criterion in the eQual method,
- theorem 4 from System_Quality criterion in the Ahn method, is included in Usability criterion in the eQual method,
- theorem 8 from System_Quality criterion in the Ahn method, is included in Trust criterion in the eQual method,
- theorems 1, 2 and 4 from Perceived_ease_of_use criterion in the Ahn method, is included in Usability criterion in the eQual method,
- theorem 1 from Service_Quality criterion in the Ahn method, is included in Empathy criterion in the eQual method,
- theorem 5 from Service_Quality criterion in the Ahn method, is included in Trust criterion in the eQual method,
• there is an overall_evaluation class determining an overall view of the website in eQual ontology, this class is being used for verifications of fragmentary evaluations given by users.

Differences in structures and the earlier described divergences among evaluation classes cause, that integrating the discussed ontologies does not limit to connecting all criteria in the Quality class, but requires also solving the inconsistency.

As a result of applying two first steps of the integration algorithm, ontology was received with the redundancy of the relation. The received ontology is included in figure 5. In some ontologies such redundant relations are acceptable, but the created ontology is supposed to allow to get the homogeneous and transparent criteria structure, so the redundancy of slots should not appear. In order to remove the redundancy we decided to assume that eQual will be the preferred ontology. Thus, in shown cases, slots appearing in this ontology were kept, however relations coming from Ahn ontology were removed. Additionally, the criteria structure and the theorems coming from the Ahn method was partially reorganized. The Service_Quality criterion was entirely removed, and its components were moved to the Service_Interaction group to the Trust and Empathy criterion. In the end, a homogeneous criteria structure of criteria and theorems included in figure 6 was received.

Figure 5. Redundancy of the relation appearing in ontology get as a result of integration. Source: own study.
5. CONCLUSIONS

The cardinality of the website quality evaluation methods and their relative resemblance determine the need of creating a methodology of their unification. It is tied up with the process of creating new models of quality evaluation, which are usually based on criteria used in existing methods. Such an approach is rational, but needs very wide literature analysis in order to select the right evaluation criteria.

In this work, an attempt to connect two quality evaluation models was made and ontologies were used for it. To achieve this goal, tools offered by ontologies were used, such as: semantic analysis, analysis of the ontology structure and the analysis of names, coming from the methodology of ontologies integration. The ontological approach allowed to formalize a procedure of merging evaluation methods, criteria and hierarchies. In the end, a homogeneous structure of quality rates coming from two distinct methods was received. This structure can facilitate the process of selecting quality criteria for websites of different kind, giving the composition containing criteria used in different models. Of course, the discussed structure will get such a functionality only in the case, when it will unified a large number of evaluation methods, many more than only Ahn and eQual models discussed in this work.
References


