

Enterprise Architect and development solution of application framework

J. Sedivy, R. Borkovec

Abstract— Application design can theoretically occur using only pencil, brain and notepad. When we use these basic tools, we create only any simple application model. This model but will use only you familiar syntax and semantics. We will hardly their mysterious pictures to share. Another problem is that behind these sketches usually costs no consistent or logical meta-model formally codified semantics of model elements. Communicating such proposal is accompanied by the words "probably", "maybe" and "how it was meant."

Keywords— enterprise architect, framework, implementation base framework, UML language.

I. INTRODUCTION

WHEN we implementing software contracts we have to respect these fundamental points

- a) professional approach to client issues
- b) reliable execution platform that will be common to all projects implemented
- c) the creation of high quality, accurate documentation, both technical and user
- d) ensure the possibility of product updates and changes with minimal need for recompiling and builds
- e) the open interfaces for multi-language solutions

Libraries vs. frameworks:

Dates library and framework are in the world of programming is often used, but few would probably be able to tell you exactly how they are connected and what is actually from each library and framework differs. Framework would be possible to put the first attempt to define a comprehensive set of thematic libraries, but is caught by this definition really everything? Let's take a closer look at it.

Libraries:

The intuitive idea of the significance of the concept of a library are probably all programmers [1]. Very loosely speaking in terms of a set of functions and classes providing a

range of services [2].

Libraries can use different applications that do not typically used separately but in combination with other libraries. Libraries also can benefit each other. At the lowest level, the library or the actual application code to communicate with the layers beneath them or directly with the hardware [3].

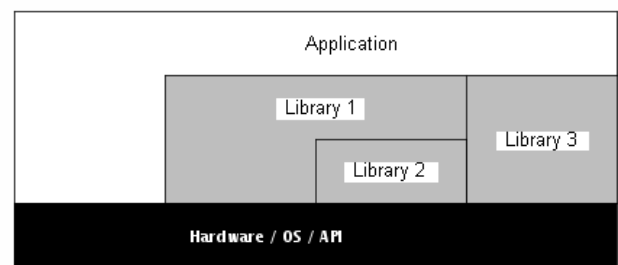


Fig. 1 A typical situation in application with library

Frameworks:

Framework we define a comprehensive set of thematic libraries + policy decisions. The situation looks like this:

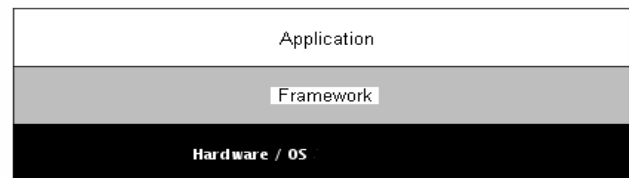


Fig. 2 A typical situation in application with framework

Differences in the situations with libraries and frameworks is possible to comment succinctly: The application uses the library, but it is written in the framework [4]. From this difference arise and different rules makers must adhere to libraries and frameworks. Libraries must be team players, while frameworks can afford to occasionally be "dictators".

II. PROFESSIONAL APPROACH TO THE PROBLEM OF CLIENT

If the editing software procurement incomplete or inaccurate, even the best application can not reach a satisfactory outcome [5]. Prerequisites for that award went well and gave so much developer indispensable basis for their work, could be summarized in the following points:

- a) knowledge of solved problems
- b) the communication skills of the analyst, establish a good communication channel with the client and use those resources

This article was created under the project called Specific research SV 2115 done at UHK Hradec Králové in 2015.

J. Sedivy, senior lecturer, University of Hradec Kralove, Faculty of Science, Department of Informatics, Rokitanskeho 62, 500 03 Hradec Kralove, Czech Republic (phone: +420 493331171; e-mail: josef.sedivy@uhk.cz).

R. Borkovec, IT specialist, University of Hradec Kralove, Rokitanskeho 62, 500 03 Hradec Kralove, Czech Republic (phone: +420 493331171; e-mail: roman.borkovec@uhk.cz).

that will actually use the resulting application [6]

c) respecting the connection to the outside world, whether it is necessary to import and export data, or links to other applications [7].

d) to estimate the direction of further development or degree of autonomy applications

Of those paragraphs that would be at this stage of development analyst paid tool that allows him to communicate with the client at the earliest opportunity to view his ideas in the resulting environment [8]. On the client ca not assume a rigorous understanding of UML schemas, even more can be expected that this ignorance and admit because there is a misunderstanding. Which is where the award is a fatal error. How to solve this conflict, suggesting another subchapter.

III. RELIABLE IMPLEMENTATION BASE FRAMEWORK

The aim of the framework is the assumption of the typical problems of the region, thereby facilitating the development so that designers and developers can focus on their task [9]. For example, a team that uses Apache Struts to develop websites for the bank can focus on how they will carry out banking transactions and not to ensure flawless navigation between pages.

There are objections to using the framework code will slow or otherwise ineffective and that the time is saved by using foreign code must be given to staging framework. However, when his repeated commitment or large project, there is a substantial time savings. When uninstalling framework will not be able to run some applications [10].

Framework consists of the so-called. Frozen spots and hot spots. Frozen spots define the overall architecture of the software structure, its basic components and the relationships between them. These parts do not change in any way the use of the framework. In contrast, hot spots are components that together with the programmer code creates a very specific functionality and therefore are almost always different [11].

In object-oriented environment framework consists of abstract and conventional classes. Frozen spots can then be represented by abstract classes and custom code (hot spots) was added to the implementation of abstract methods.

Examples:

- JUnit is a framework for testing the units for the programming language Java.
- Spring is an application framework for the Java platform open source.
- Zend Framework is a framework for web applications in PHP open source.
- Vaadin is a framework for web applications in Java open source.
- Nette Framework is a framework from Czech author for web applications in PHP open source.
- CakePHP Framework is a framework for web applications in PHP open source.
- Symfony is a framework for developing web applications in PHP open source.
- CodeIgniter is a framework for developing web applications in PHP open source.
- Apache Wicket is a framework for developing web applications in Java open source.
- Ruby on Rails is a framework for web applications in Ruby open source.
- jQuery is a lightweight JavaScript framework with open source code.
- The .NET Framework is a framework for language C #

In our particular case is not so much about ORM (Object Relational display O / RM or O / R display) is a programming technique in software engineering, which provides automatic conversion of data between relational databases and object-oriented programming language.), or similar solutions. It is a comprehensive system that meets all specified requirements. The basic idea of the whole framework is a generic creation majority of forms and direct modifications without having to build the application itself [12].

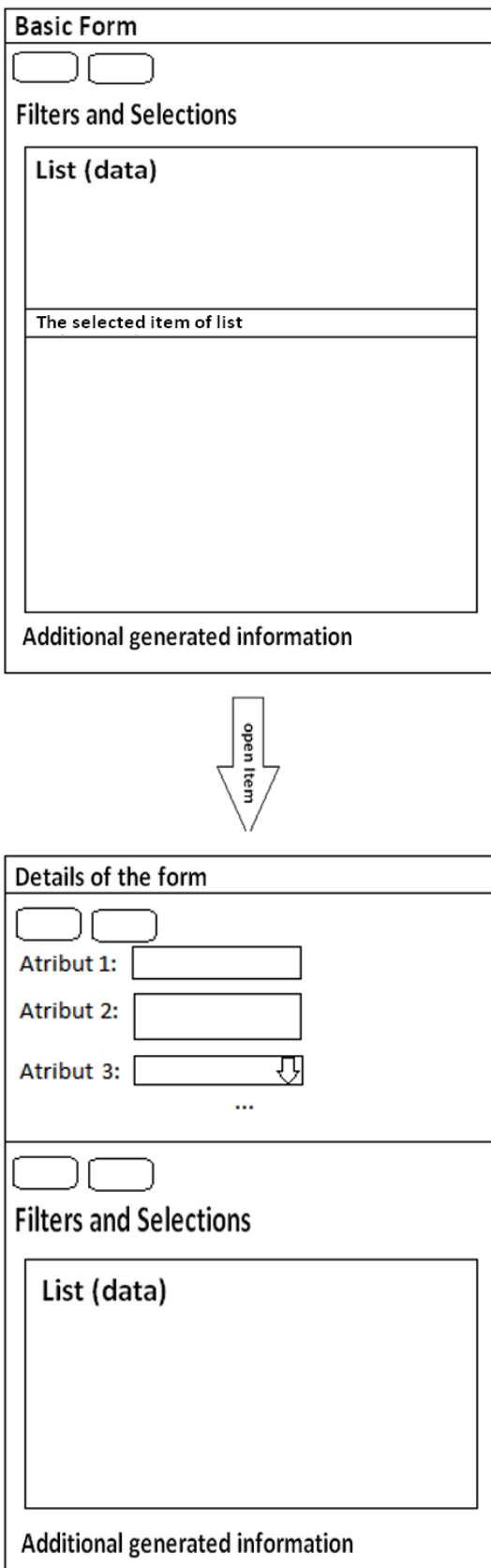


Fig. 3 Majority form

The dependent lists detail the course progresses further recursively, where possible approaches and all things related to the access and business logic of the application [13].

On the form, in the upper part of the visible buttons that allow you to extend the functionality of form, they can be called up outside, programmer-designed elements - whether it is a form of executive or call stored procedures on the database server or specific methods of application code - in Then use .NET reflection in its entirety. In the case of detail is a suitable place for the location type buttons Save, Undo, Copy form, etc.

By being available to the interface, it will receive only a single ergonomic presentation layer, but also an interface that is effective at the time of application design with the contracting authority [14]. Receives direct vision of the future appearance, layout elements (whether columns in the list, the filtering, sorting, but the arrangement detail forms, including selection of appropriate components - calendars, vintage elements, check elements etc.)

Definitions of these elements can be implemented at least two ways - either by direct description in the database where there is both data and business layer and also a description of the presentation. This method allows to realize the most fundamental changes, such as bringing not only development, but also for future use applications directly, without having to build applications. Or, in the case of a consistent three-layer application can be used as source elements class that reflection transformed into the final application solutions. In this case, it is necessary when a change in classes with a build count [15].

By drafting the basic design solutions in collaboration with the client, creating the necessary elements in the database. These can be used for transformation using the Add in EA generate analytical model so that it remains in direct connection with the application. All changes will be reflected on both sides, both from the model to the application, and vice versa.

IV. CREATION OF QUALITY DOCUMENTATION UNITS

Technical documentation is perfect solved on the level of development tools, both Visual Studio, as well as the actual Enterprise Architect. But remains the issue of user documentation, which does have the normal function of the sporadic use - user rarely actually use this documentation, but it is absolutely necessary part of the assembly task.

Proposed Framework enables a solution that ensures due to the following functions:

- Reducing the overhead required for creating user documentation that is partly generated and partly it has the ability to directly generate user
- Differentiation between the different parts of the dossier by access roles. It means that there is no need to provide the resulting complete user manual, which is in charge of a strict application space. Documentation is generated only for those elements that for him the privileges accessible

- Use of framework options for printing and folding the proper coverage documentation

V. LINKS TO THE SURROUNDINGS, EXPORT - IMPORT DATA PRINTS

Component Solutions Framework allows you to gradually expand its capabilities, and this also applies to the issue.

The basic component that will provide these options is a list. After narrowing the data and their arrangement, the user will want this data to be exported outside the application. The first step in the implementation is likely to be linked to Microsoft Office tools, which can be connected to their templates to use very widely. In some applications, this solution is quite sufficient and does not need to dial another extension.

A good step is to ensure communication with the Open Document protocol and XML - in a generic and component solutions is an advantage not only time-space, but also full applicability throughout the application in those places where the import or export offer.

Of course, in specific cases, intervention can not be avoided in the application code, but for standard situations, all of these interventions can be defined directly in the database, or builds again without updating the application itself.

VI. OPEN MULTI-LANGUAGE INTERFACE APPLICATION UPDATE

Today, the requirements for localized increase in proportion with the growth rate of transnational projects and also in providing information and services outside one specific destination.

Question localization is not just a question of translation output strings. There can be addressed relatively simple function, which also ensures the fulfillment of the phrase in the dictionary itself with the application. It is also addressing local codebooks and possible differences in some procedures, e.g. Calculation or legal.

An important issue is to provide a simple interface for translators and the possibility of pairing their work. The translator can not be bother complex operations, but it is necessary to ensure the synchronization of their work and the maximum overall efficiency and flawless.

VII. GENERIC FRAMEWORK AND ITS LINKS WITH DESIGN PATTERNS IN SQL

As has been mentioned in several articles, generic Framework not only for the packaging business layer and its connection with the data, but creates a kit with its own interface for the creation of complete systems. Its purpose is to streamline their implementation, reduce dependence on highly professional staff programmer and transparent processes supporting the entire information system. It also allows developing custom information systems, however, where the company has qualified to create such systems.

Generic Framework perfectly solves the problems of development itself. In previous articles we have dealt with the issue of linking development and design. It is very

counterproductive if it makes an excellent proposal and then the development will differ from him, mostly because of the small teams are not able to maintain a synergy between the development of a rear design changes - those in real development occurs almost always.

A detailed description of similar problems includes e.g. work prof. Repka, as well as other artists. Historically, it has been presented Len Silverston and the reasons for these problems (linking design and development) are in principle remains the same - the human factor.

If we manage to link the generic Framework with modeling tools, so that there was a mutual updates is to reach a solution.

These issues are very closely related design patterns. They are actually the foundation of modeling, in all layers of conventional systems.

VIII. PRINCIPLE AND JUSTIFICATION OF DESIGN PATTERNS IN SQL

Design patterns are a kind of general rules that were created on the basis of good experience in the development. There are several levels, a classic case of the developer's design pattern for programmers Singleton pattern - ensures the creation of just one instance of the object, which means that the constructor if its existence is no more does not create but merely passes this instance. Following is a sample of code:

```
class Singleton {
    // immediate initialize an instance when the first use of
    // the building.
    private static Singleton instance = new Singleton();
    static Singleton()
    {
    }
    //Private constructor
    private Singleton() { }
    //Static property ensure the return of instances
    public static Singleton Instance { get { return instance; } }
}

class Program {
    //Use
    public static void Main() {
        var s = Singleton.Instance;
    }
}
```

Imagine that we want to generate such a class wherever it is needed - no sense in every project to deal with it again. If the modeling tools such class is defined, it is relatively easy to add internal interpreter requirement to conduct such an object or generating code this into account. The developer will not have to worry about the correctness of the code is not necessary modifications, it is simply a design pattern.

The proposals and the business layer and in the actual coding of this methodology very soon took over. The reasons are obvious, first of all it's security code, his lucidity (when used in the model is seen as a requirement for the development of a model to solve), and higher overall processing efficiency.

At the level of the data link layer (ie primarily SQL Server), this methodology has not taken off too. Certain design patterns exist in SSIS, or integration services, but they still are not active base information system.

While the issue of coding the database server side should be just as effective as any other layer. It is true that in recent times is to move everything into the business layer and often deal with this on the client side. Or means of client applications. Unfortunately, it also brings large disadvantages. In the first place actually they might lose the use of specific solutions to these servers are not used to their physical differences, as well as internal solutions and semantic possibilities transaction languages.

Therefore, it is appropriate, especially in applications where high performance is required, the burden rest on the performance of the database server. For effective and routine use of design patterns are then SQL very supportive.

IX. SAMPLES OF DESIGN PATTERNS IN SQL

Selecting random record:

Random selection allows recording from any dataset choose just one random record. It may be used where it is necessary to model or make random selections from a predefined data sources of any complexity.

The pattern uses the ORDER BY qualities that allow you to interpret any, syntactically correct expression. Although this command uses mostly only on linear shifting, it can be used in this case. Added function NEWID () returns a random GUID for each selected record and subsequently returns only the first record in the dataset whose location is completely unpredictable.

Entering pattern diagram contains RANDOM, or RANDOM n, if required a certain number of random records.

Task:

Select five random records from a list of employees, where DateOfBirth <01.01.2000

Solution:

```
SELECT TOP 5 * FROM employee WHERE BirthDate
<01.01.2000
ORDER BY NEWID ()
```

Conditional shift:

The model solves the situation when it is necessary in one data set to realize more conditions to its sorting. It uses mostly not very used command option ORDER BY.

As already indicated in the pattern Random entry, click ORDER BY has significantly wider possibilities than mere linear transmission. According to the internal interpreter translates basically everything was delivered, and it is a mechanism which can solve the requirement for conditional shift. Entering the pattern in the chart contains a definition of CASE.

Task:

Select all the records of addresses whose AddressID is <100, while there was a sort according to the following rules:

- If the OrderBy = 'Postal' while OrderByDirection = D, these records Sort by PostalCode and descending
- If the OrderBy = 'Postal' while OrderByDirection! = D, these records Sort by PostalCode and ascending
- If the OrderBy = 'State' while OrderByDirection = D, these records Sort by StateProvinceID and descending
- If the OrderBy = 'State' while OrderByDirection! = D, these records Sort by StateProvinceID and ascending
- If the OrderBy = 'City' while OrderByDirection = D, these records Sort by City and descending
- If the OrderBy = 'City' while OrderByDirection! = D, these records Sort by City ascending

Solution:

```
SELECT AddressID, City, StateProvinceID, PostalCode
FROM Person.Address
WHERE AddressID <100
BY ORDER
CASE WHEN OrderBy = 'Postal'
AND OrderByDirection = 'D'
THEN END PostalCode DESC,
CASE WHEN OrderBy = 'Postal'
AND OrderByDirection! = 'D'
THEN END PostalCode,
CASE WHEN OrderBy = 'State'
AND OrderByDirection = 'D'
THEN END StateProvinceID DESC,
CASE WHEN OrderBy = 'State'
AND OrderByDirection! = 'D'
THEN END StateProvinceID,
CASE WHEN OrderBy = 'City'
AND OrderByDirection = 'D'
THEN END DESC City,
CASE WHEN OrderBy = 'City'
AND OrderByDirection! = 'D'
City THEN END
```

Column with ranking

Pattern is used to display the serial number of the record, which is not related to any specific indication in the data. After obtaining the data source, thus excluding the acquired data is added as the first column of the order number.

MSSQL server from the 2005 version offers for the realization of this design function ROW_NUMBER

```
select rank() OVER (ORDER BY a.au_lname, a.au_fname)
as rank, a.au_lname, a.au_fname
from authors a
order by rank
```

Table 1 Generating a sequence number

Rank	Au_Lname	Au_Fname
1	Bennet	Abraham
2	Blotchet-Halls	Reginald
3	Carson	Cheryl
4	DeFrance	Michel
5	del Castillo	Innes
6	Dull	Ann
7	Greene	Morningstar
8	Green	Marjorie

Such patterns can be designed countless, without any further demonstrations can be designed e. g.:

- Counting
- Sequence generator
- Decomposition Retec / collection
- Aggregation lists
- Enumerated couples
- Enumerated set
- Compared intervals
- Distribution discrete intervals
- User-defined aggregation
- Pivot
- Symmetric difference
- Histogram
- Queries Skyline
- Relational division
- Duplication of records

X. EFFECTIVE INTERCONNECTION OF GENERIC FRAMEWORKS AND DESIGN PATTERNS IN SQL

In conclusion we come to his goal - to effectively interconnect design patterns in SQL with a generic framework.

As already stated, the opening of scissors when being implemented model is due to a dangerous thing. Especially when extensive and complicated process information system. If the model is incomplete, inaccurate, it may cause more problems than if the developer was forced to work without it.

Using design patterns in models and their interpretation to direct development to prevent unwanted resistances, so that may arise. Work analyst and the developer is thus not inconsistent within and complete cooperation could result in giving the best of what is possible and what is expected from the sponsor.

XI. CONCLUSION

Options and concrete solutions are very wide and after each team has to learn basic functionalities opportunity to constantly replenish their framework and improved. The basis should always be the effectiveness of development, optimizing data access and maximum user friendliness. In an environment. NET is worth solve database level using DB Providers - then

there is no need to develop two frameworks just because the company develops applications e.g. for Firebird and MSSQL or Oracle. Basic core framework is thus independent of the database, and it enables you to provide a wider client access, which in turn are inputs into other financial firms.

A conclusion section is not required. Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions.

ACKNOWLEDGMENT

This article was created under the project called Specific research No. 2115 done at UHK, faculty of Science, Hradec Králové in 2015.

REFERENCES

- [1] P. Voborník, Migration of the Perfect Cipher to the Current Computing Environment. *WSEAS transactions on information science and applications*. 2014, 2014(11), p. 196-203.
- [2] R. Nemeč, S. Hubálovsky, The use systems for measurement using a computer in the school laboratory to explanation for temperatures during the water heating. In: *Education and modern educational technologies (EMET 2014)*. Piscataway: IEEE, 2014, p. 103-107.
- [3] P. Voborník, Mini-Language for Effective Definition of the Color Gradients. In: *Materials, transportation and environmental engineering II (CMTEE 2014)*. Zurich: Trans tech publications, 2014, p. 1882-1885.
- [4] R. Nemeč, M. Hubalovska, S. Hubalovsky, User Interface of System SMPSL. In: *Communications and information technology (CIT 2014)*. Salem: North atlantic university union, 2014, p. 324-329.
- [5] P. Voborník, Modification of the perfect cipher for practical use. In: *Manufacturing, engineering, quality and production systems (MEQAPS 2014)*. Athens: World scientific and engineering academy and society, 2014, p. 64-68.
- [6] S. Hubalovsky, J. Sedivy, *Algorithm Development and Computer Simulation of Color Decoding of Mastermind Board Game*. Applied mechanics and materials. 2013, Vols. 333-335, Ch. 6, p. 1349-1352. Zurich: Trans tech publications
- [7] S. Hubalovsky, J. Sedivy, Education of student's project team cooperation using virtual communication supported by LMS system. In: *14th International Conference on Interactive Collaborative Learning (ICL2011) - 11th International Conference Virtual University (VU'11)*, Bratislava, Slovenská technická univerzita, 2011, p. 456-459.
- [8] S. Hubalovsky Remote desktop access us a method of learning of programming in distance study. In: *14th International Conference on Interactive Collaborative Learning (ICL2011) - 11th International Conference Virtual University (VU'11)*, Bratislava, Slovenská technická univerzita, 2011, p. 450-455, ISBN 978-1-4577-1746-8.
- [9] S. Hubalovsky, Research of Methods of a Multidisciplinary Approach in the Teaching of Algorithm Development and Programming. In: *9th International scientific conference on distance learning in applied informatics (DIVAI 2012)*, Nitra, Univerzita Konštantína Filozofa, 2012. p. 147-156, ISBN: 978-80-558-0092-9.
- [10] R. Nemeč, S. Hubalovsky, *Software Design of System SMPSL*. WSEAS transactions on computers. 2014, Vol. 13, p. 329-337. ISSN 2224-2872.
- [11] M. Hubalovska, S. Hubalovsky. *Implementation of the Systems Approach in Mathematical Modeling, Dynamic Simulation and Visualization Using MS Excel Spreadsheet*. International Journal of Mathematics and Computers in Simulation. 2013, Vol. 7, No. 2, p. 267-276.
- [12] S. Hubalovsky, M. Musilek. *Modeling, Simulation and Visualization of Real Processes in LOGO Programming Language as a Method of Development of Algorithm Thinking and Programming Skills*. International Journal of Mathematics and Computers in Simulation. 2013, Vol. 7, No. 2, p. 144-152. ISSN 1998-0159.
- [13] Hubalovsky, S., Milkova, E.: Modeling of a real situation as a method of the algorithmic thinking development. In: *Advanced Educational Technologies, Proceedings of 6th WSEAS/IASME International*

Conference on Educational Technologies (EDUTE'10), WSEAS Press, Kantoui, Sousse, Tunisia, May 3-6, 2010, pp. 68–72

- [14] E. Milková, A. Hůlková, Algorithmic and Logical Thinking Development: base of programming skills. *WSEAS transactions on computers*. 2013, 12(2), s. 41-51.
- [15] E. Milková, Development of Algorithmic Thinking and Imagination: base of programming skills. In: *CSCC 2012: proceedings of the 16th WSEAS international conference on communication*. Athens: World scientific and engineering academy and society, 2012, p. 347-352.

Ing. Mgr. J. Sedivy, Ph.D., was born in 1963 in Czech Republic. Doctor degree in Theory of technical education in 2006 on University of Hradec Kralove, Faculty of Education, Czech Republic. University of Hradec Kralove, Faculty of Education, Department of Technical Subjects, Rokitanskeho 62, 500 03 Hradec Kralove, Czech Republic (phone: +420 493331171; e-mail: josef. sedivy@uhk.cz). His scientific activities are computer graphics and communications in education and informatics.

R. Borkovec, was born in 1963 in Czech Republic. System Engineer on University of Hradec Kralove, Czech Republic. University of Hradec Kralove, Faculty of Education, Rokitanskeho 62, 500 03 Hradec Kralove, Czech Republic (phone: +420 493331171; e-mail: josef. sedivy@uhk.cz). His scientific activities are programming, education and informatics.