

Estimation of Feasibility for Business Information Systems

Krešimir Fertalj, Boris Milašinović and Damir Kalpić

Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia

This paper describes the architecture of a system called Business Information System Feasibility. The purpose of the system is to establish a base for an impartial and computer-aided decision-making tool for choice of information system software and/or its provider, for evaluation of computerization projects and for validation of existing information systems. In the paper we present the system model, the implemented features and the algorithm used for calculation of product scores feasibility query.

Keywords: ERP procurement, software product evaluation, feasibility analysis.

1. Introduction

In 2001, a comparative analysis of domestic and foreign Enterprise Resource Planning (ERP) software present at that time in Croatia was performed and presented within the project “Comparative analysis of ERP software in the Republic of Croatia” (Grant by the Ministry of Science No. 00-146) [2]. After the project completion, some suppliers expressed interest in evaluation of their ERP systems and there were users who wished to obtain information about some ERP systems that had not been covered by this primary investigation. This interest motivated us to propose the project continuation under the name Business Information System Feasibility (BISF, or IPIS in Croatian) [3]. This continuation was not limited to ERP systems, but it has been extended to other issues concerning the information systems software, like:

- Criteria for software selection,
- Criteria for selection of software vendor,
- Risk estimation factors,

- Elements for project feasibility estimation,
- Elements for project evaluation.

The purpose of the system is to establish the base for an impartial computer-aided decision-making tool for the choice of an information system software and/or its provider, for the evaluation of computerization projects and for validation of existing information systems.

To users or prospective users the answers to the following questions are offered:

- What is an appropriate solution, and for whom is it appropriate?
- To buy from existing software or to develop on purpose?
- What has to be done and what conditions should be met for a computerization project to succeed?

We tried to make the criteria as objective as possible, believing that there is no absolutely best solution. There are many solutions and, for a specific user, one of the solutions could be the most appropriate [1][6].

BISF software enables vendors to present their products. BISF also enables users to compare the products and to select a product that meets best the requirements in terms of characteristics and price. Furthermore, the system enables users to assess their projects.

The analysis of business-oriented state-of-the-art software offered in Croatia (vendors' view) and the analysis of desired features of business-

oriented software (customers' view), are performed respectively through the evaluation of product features, or by enquiring about the desired features.

Based upon the defined criteria, desired features and offered characteristics, including the vendor's support, the degree of matching between desired and offered features, can be determined. Also, general weighting factors for the evaluation of features can be derived.

The resulting knowledge-based system is aimed for selection of ERP software and/or vendor, for evaluation of computerization projects and for validation of existing information systems.

2. System Architecture

2.1. Database Model

Fig. 1 shows principal entity sets and their relationships. Table 1 contains brief descriptions of each entity set.

2.2. Users and Products

For simplicity reasons, we can assume that each product belongs only to a single evaluation category. As shown in later paragraphs, there are some products that fall in two or more categories (e.g. ERP2000 and ERP2004), but for the modeling purposes we made copies of these products. This approach was selected because sets of properties belonging to different categories are often disjunctive. Some categories may have common properties, but positioned on different hierarchy levels of the tree data structure. Therefore, it is more convenient to treat them as separate copies of properties.

Since most products, except some of those in the ERP category, really belong only to a single evaluation category, the assumption appears reasonable.

Our model allows any user to evaluate any product. This principle has some advantages and disadvantages. An obvious advantage is to avoid assigning unrealistically desirable scores by software manufacturers to their products.

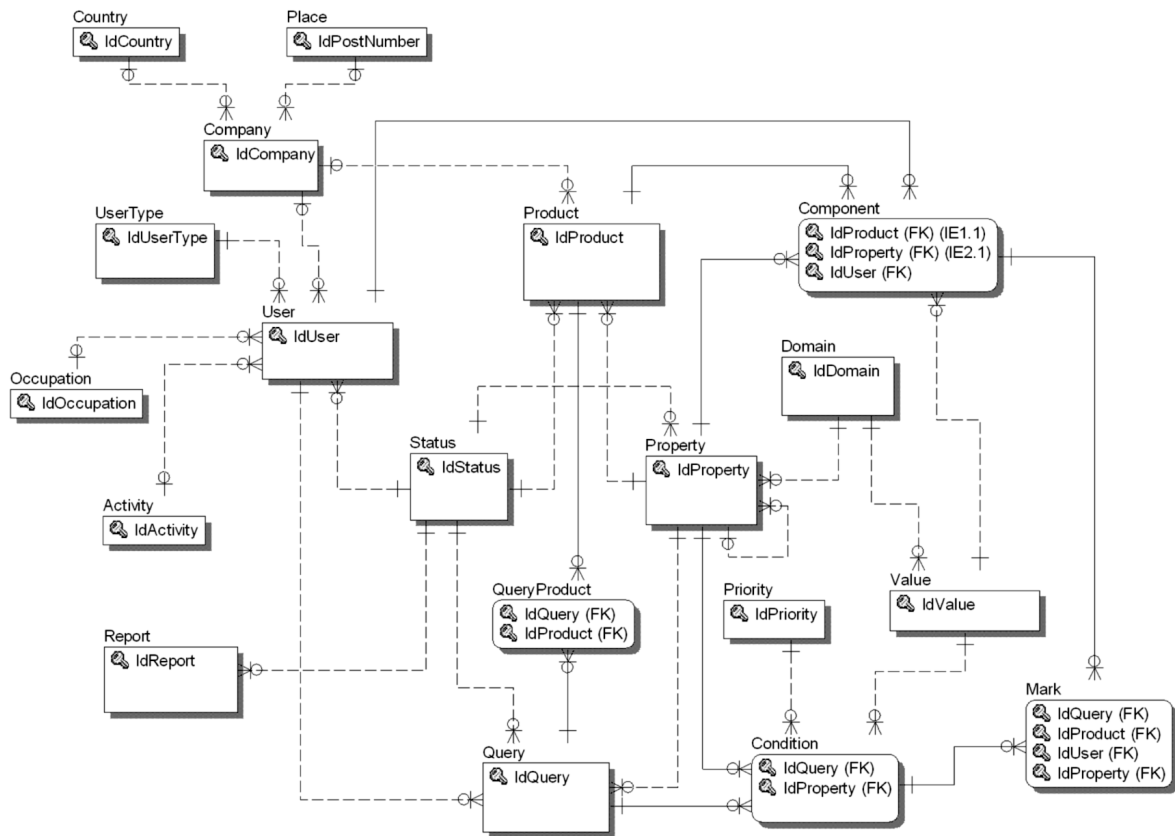


Fig. 1. Entity sets with their key attributes and relationships.

Entity set	Description
Activity	Users' activities
Domain	Domain values for properties
Report	Definition of dynamically created reports
Component	Components of a product
User	Users
Parameter	Application parameters
QueryProduct	Filtered products for a query
Product	Products and services that are evaluated
Property	Properties that are estimated
Country	List of countries
Place	List of places
Status	List of possible statuses
Priority	List of possible condition priorities
Value	List of possible property values
UserType	List of user types
Occupation	Users' occupations
Company	Users' companies and suppliers
Query	Feasibility query
Condition	Conditions within a feasibility query
Mark	Value of a component, based on the query condition

Table 1. Brief description of entity sets.

That would degrade our system to free commercial advertising. While being evaluated by other users, such software will attain realistic scores because any users that have had a chance to use particular software can contribute with their comments and evaluate particular software properties. On the other hand, this can become a major disadvantage because a user may make negative evaluation of a competitor's product or use the opportunity to express his/her personal dislike. To make a compromise between these two opposing extremes, we decided to divide the users into several categories: non-registered

users, registered users, evaluators, suppliers and administrators. In this way, only evaluators and suppliers can evaluate products (with the difference that suppliers can add new products), and administrators can correct their marks. Registered users are promoted to suppliers and evaluators manually, so we have some kind of control over reviews and scores to be fair and objective.

As a product can be evaluated by multiple referees, we had to choose what score values should be used as resulting values for such a multiply evaluated product. Without a proper measure for weighting each of the different evaluations of the same product, we dismissed the idea of calculating the average. Instead, we maintained the pairs (product, evaluator). While using the word "product" in the context of results of a feasibility query, we assume these pairs as entities. In this way the same product may appear in a list several times, each time with an annotation who the evaluator was for the particular instance.

2.3. Feasibility Query

The feasibility query is made within a particular evaluation category. After choosing a desired category, a hierarchical list of properties belonging to a chosen evaluation category is displayed. To each product property, the user can assign a desired score value and priority. Currently available priorities are: irrelevant, desirable, important and necessary with the respective values 0, 1, 2 and 3. By setting priority of a property to irrelevant, the user will exclude that property from the calculation algorithm.

In this way, a user creates one or more conditions in the feasibility query. The user can mark a product as not feasible if a particular condition is not satisfied. We say that such property is marked as a disqualifying property. If the disqualification is not set, then the products that do not match desired values will only achieve lower matching percentage.

The result of each feasibility query is a list of products satisfying desired conditions together with matching percentage and product mark. By varying the priorities assigned to conditions a product's score is changed accordingly after the calculation algorithm to be described in the next section. A rank list of considered products is produced to make the comparative analysis easier.

At any moment, either before submitting the conditions or after the results are displayed, the user can filter the products to be considered in the feasibility query.

2.4. Product Evaluation

Product properties are presented in form of a tree data structure. The root property of a particular product is the property that is equal to the category name of that product. Every product can be evaluated by any evaluator so let v_s be a value of property S in a product evaluated by some evaluator. Since the whole tree can contain properties with values from different domains, the values have to be aggregated and normalized. A way to normalize the values given to some property can be to divide them with the maximum value from that domain in order to obtain the relative property value. This will lead to the situation that the products with higher values will have better final values, like it was the case in [2]. However, we do not believe that such normalization would be appropriate for this model. We do not want to appreciate the excess of some property if the user does not need it. As the user assigns desired value to each property, we assume that all the products bearing values which are equal or higher than the desired one, are equally valuable. Therefore, we divide the value v_s with the desired value (rv_s). Using this kind of normalization we avoid higher ranking of products that excel in some unnecessary aspects. In case the desired value is 0, the relative value is set to 0 by definition. Now we can precisely define the relative value of a property S as:

```
if  $rv_s > 0$  then
   $Rel(S) = \min\{v_s / rv_s, 1\}$ 
else  $Rel(S) = 0$ 
```

Besides assigning the desired value, the user also assigns the priority (weight) to each property (p_s). The relative value of property S multiplied with the priority of S and divided by the sum of weights for properties sharing the same parent, yields the normalized value:

$$NO(S) = \frac{p_s * Rel(S)}{\sum_{i \in Children(Parent(S))} p_i}$$

Another computational problem is how to determine the value for a property that has children.

The dilemma is whether the entered value or the average of children values should be used. Our decision was to use the average value of children, rounded up to the nearest discrete interval in order to satisfy database model constraints for integer domain. For this purpose we define a new value V_s to be calculated in the following way:

$$V_s = 0$$

if S is not leaf then: $V_s = \sum_{ds \in Children(S)} NO(ds)$

if V_s is equal 0 then $V_s = v_s$

Using the above definition, the relative value of property S is calculated as:

```
if  $rv_s > 0$  then
   $Rel(S) = \min\{V_s / rv_s, 1\}$ 
else  $Rel(S) = 0$ 
```

After defining these values we can define the total product score as V_{RP} where RP is the root property for a particular product evaluation.

2.5. Evaluation Categories and Value Domains

As mentioned in the introduction, BISF as a new system was not limited to existing ERP properties. Preferred ERP characteristics have been enumerated, restructured and updated by using various data sources [5][7][8][9][10]. The following issues have been defined:

- Criteria for software selection,
- Criteria for selection of software vendor,
- Risk estimation factors,
- Elements for estimation of a project feasibility,
- Elements for project evaluation.

In ERP2000 evaluation category was simplified to have maximum depth of one, and all the properties had YESNO domain covering values with either "yes" or "no" (see Table 2). In BISF, several new domains have been defined, so that ERP2004 currently contains a hierarchical list of properties with levels nested deeper than before and with possible values in functional domain. Currently, the possible values range from zero to five and, together with their description, they are made visible to the users.

The complete list of domains is presented in Table 2.

Domain Id.	Domain Name
PRICESEAT	Price per end user (workstation).
PRICEPACK	Price of the whole software package or price of an IS installation.
PRICEREL	Costs per year, relative to initial costs, e.g. yearly maintenance costs 20% of the purchase price.
PRICEHR	Price per hour divided in discrete intervals.
YESNO	No, Yes [0, 1], where Yes stands for better.
FUNC	Functional characteristics [0-5], Not implemented, Implemented via partner product, ..., Fully implemented, Sophisticated features.
CHECK	Checkbox domain: 0 used for No, and 1 for Yes.
QUANTITY	Rank of quantity, 0, 1 to 1000, 1001-5000, ... [0-4], e.g. Number of users.
NOYES	Yes, No [0, 1], where No stands for better.
NONFUNC	Non-functional properties: Not existent (0), poor (1), good (2).
HUMANRES	Number of human resources divided in four discrete intervals [0-3]. Zero is used when there are no employees.
REPEAT	Repeating, none to many times [0-3], e.g. Number of post-delivery integrations.
PERCENT10	Percentage divided into discrete intervals with step size of 10% [0.0-1.0].
LEVEL	Level: none, low, intermediate, high [0-3], e.g. Level of technical expertise required for maintenance.
RESOURCE	Resource consumptions (less is better) divided in four discrete intervals [0-3].
RISK	Risk: none, low, intermediate, high [0-3]. The values diminish with risk increase.
TIME	Time of delivery: unlimited or unknown, year to day [0, 5], shorter is better.
DURABILITY	Durability or stability, divided in four discrete intervals [0-3]. Zero is used when no durability is defined.
IMPORTANCE	Importance: unimportant (0), desirable (1), Important (2), very important (3).
DURATION	Duration: not estimated, hour to days [0-3], shorter is better, e.g. Time of recovery after failure.

Table 2. Domains for property values.

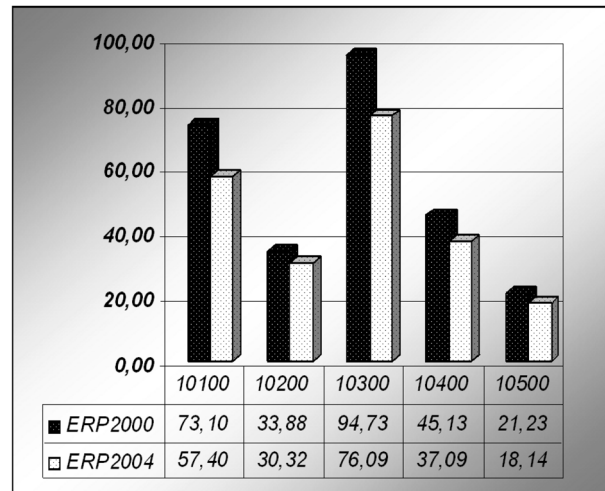


Fig. 2. Product scores in comparative analyses of five products in two different ERP evaluation categories.

After the restructuring of ERP characteristics we performed a feasibility query on the same products that had existed in the former project. Product names were coded as 10100, 10200, 10300, 10400 and 10500. The graph in Fig. 2 shows the products' scores in both comparative analyses.

As we can see from the graph, total scores for all the products were lower in the ERP2004 model.

The reason for this was a change of the domain type. All values from ERP2000 model that had belonged to YESNO domain were translated into the functional domain (see FUNC in Table 2) ranging from 0-5 with initially transformed values of "no" into 0, and "yes" into 4. In this way we degraded all the products in terms of the final score, but their relative proportions remained the same. By doing so, we have shown that final results are nearly invariant to the evaluation model and to the evaluation process. The product property scores decide about the outcome. It is not only the final score that matters, but also the products' relative scores.

3. Conclusion

The hardest question any decision maker asks himself/herself when choosing an information system software is which product is appropriate for his/her needs. In most cases, there is no exact answer. Manual comparison of full sheets of properties, usually bearing different types of

values, can cause frustration and increase the risk of choosing a wrong product.

Our system is not a magic wand or a magic mirror that would give all the answers. However, it can help to make a correct decision. The best software for a user is the one that suits him/her best and he/she will find it by posing the right questions. By providing more than 4000 properties divided into several evaluation categories we have provided a potential user with an applicable tool that allows feasibility queries with a possibility to vary the parameters and present the results either in numeric or in graphic form.

Of course, different users may have different results because priorities and requests are subjective and different for every user, but the tool will give the most objective result based on input parameters.

Our optimism is based on the interest shown in the former project and we hope that the users will recognize the advantages and possibilities of this software. If so, it could become a remarkable knowledge base for business systems in Croatia.

Compared to similar systems, the advantage of BISF lies in carefully chosen evaluation categories and their properties, as we found them appropriate for the Croatian market.

4. Further Analysis

Unlike the evaluation results, statistics for other types of feasibility queries currently do not exist, because the use of the system has just begun, but all the queries will be remembered, for the purpose of deeper analyses. We shall seek answers to the questions like:

- What are the most frequently requested properties?
- What is the correlation between desired characteristics and implemented characteristics?
- What are the properties of an average ERP package in Croatia?

5. Acknowledgments

The project Business Information System Feasibility was supported by the Croatian Ministry of Science, Education and Sports (Grant no. 03–115).

References

- [1] E. BERNROIDER, S. KOCH, Differences in Characteristics of the ERP System Selection Process between Small or Medium and Large Organizations. *Proc. of the Sixth Americas Conference on Information Systems (AMCIS 2000)*, Aug. 10-13; 2000. Long Beach, CA. pp. 1022–1028.
- [2] K. FERTALJ, D. KALPIĆ, ERP Software Evaluation and Comparative Analysis. *Journal of Computing and Information Technology CIT*. 2004; 12 (3): pp. 195–209.
- [3] K. FERTALJ, B. MILAŠINOVIĆ, D. KALPIĆ, *Business Information System Feasibility*, <http://www.zpm.fer.hr/ipis>.
- [4] X. FRANCH, J. A. PASTOR, On the Formalisation of ERP Systems Procurement. *In Procs. Continuing Collaborations for Successful COTS Development ICSE Workshop*, 2000 June 4–11; Limerick, Ireland.
- [5] G. HOMER, Homer Computer Services ERP Software Selection Guide. Homer Computer Services Pty Ltd. <http://www.homercomputer.com.au>.
- [6] L. HOSSAIN, L. D. PATRICK, M. A. RASHID, *Enterprise Resource Planning: Global Opportunities & Challenges*. Idea Group Publishing; 2002.
- [7] IEEE std. 1062–1998. IEEE Recommended Practice for Software Acquisition; 1998.
- [8] ISO/IEC Standard 9126. Information Technology — Software Product Evaluation — Quality Characteristics and Guidelines for their Use; 1991.
- [9] ISO/IEC Standard 14598. Information Technology — Software Product Evaluation — General Overview; Part 4, Process for Acquirers; 1999.
- [10] S. MCCONNELL, *Software Project Survival Guide*. Microsoft Press; 1998.

Received: June, 2005.

Accepted: October, 2005.

Contact address:

Krešimir Fertalj
Faculty of Electrical Engineering and Computing
University of Zagreb
Unska 3
HR-10000 Zagreb
Croatia
kresimir.fertalj@fer.hr

Boris Milašinović
Faculty of Electrical Engineering and Computing
University of Zagreb
Unska 3
HR-10000 Zagreb
Croatia
boris.milasinovic@fer.hr

Damir Kalpić
Faculty of Electrical Engineering and Computing
University of Zagreb
Unska 3
HR-10000 Zagreb
Croatia
damir.kalpic@fer.hr

KREŠIMIR FERTALJ is an assistant professor at the Faculty of Electrical Engineering and Computing, at the University of Zagreb, Croatia. Since graduation in 1988, he has been working at the same institution, where he currently lectures a couple of undergraduate and postgraduate courses in computing. His professional and scientific interest is in computer-aided software engineering and in complex information systems. He has written around 70 scientific and professional publications and participated in conferences locally and abroad. He is the principal researcher on the project "Improvement of information system development methods" sponsored by the Croatian Ministry of Science and Technology. Professionally, he is developing information systems and database applications. He participated in a number of projects for industry and administration, as project leader, technical lead, system analyst, developer and consultant. He is a member of ACM, IEEE and PMI and of several international program committees. Currently, he is a member of Board of Directors of PMI Chapter in Croatia.

BORIS MILAŠINOVIĆ received his BSc degree from the Faculty of Science, Department of Mathematics (University of Zagreb, Croatia). Currently he is graduate student and teaching assistant at the Faculty of Electrical Engineering and Computing, University of Zagreb. His main research interests are software engineering and Web development.

DAMIR KALPIĆ graduated and received his Ph.D. degree in the field of computer science from the Faculty of Electrical Engineering, University of Zagreb. Since graduation he has been working at the same institution, nowadays called Faculty of Electrical Engineering and Computing. He is full professor teaching information systems and software-related courses in undergraduate and postgraduate studies. He has mentored more than 200 graduation theses, and a number of M.Sc. and doctoral theses. He has written over 170 texts and R&D papers, mostly reflecting practical implementations of IT in many different fields. He is head of the Department of Applied Computing. Together with his colleagues, he has completed more than 12 complex projects accepted in practice by industry and administration. They cover information systems development, operational research applications and databases. He was deputy dean and vice dean, editor of the CIT Journal and member of many international programme committees. For eight years he was chairing the International Programme Committee of the Conference "Information Technology Interfaces".
