A Web-Based Publication Database for Performance Evaluation and Research Documentation

Karl Riedling[1] and Siegfried Selberherr[2]

Abstract - **Established publication collection systems are not well suited for evaluation purposes, particularly for technical sciences with interdisciplinary aspects. To support allocation of resources dependent on publication output, the Faculty of Electrical Engineering and Information Technology at the Technical University Vienna decided to custom-design a publication database which later was adopted by the entire university. This Web-based database supports a wide range of publication types and features simple extraction of counts and lists of publications based on a variety of query criteria. The database allows that the institutes themselves enter the publication data; various algorithms and workflow procedures maintain data quality. Currently, the database supplies all publication-related evaluation results of our university, which also affect resource allocations. In addition, it provides public search facilities and web services that dynamically create publication lists and records, and exports its contents to the university library system. Therefore, the database also serves as an important research documentation tool.**

Index Terms - Publications, Research documentation, Research evaluation, Web-based database.

INTRODUCTION

The scientific community commonly judges the quality of scientific work based on the resulting publication output. However, it is often not straightforward to determine this output reliably: Data provided by researchers are not in all cases accurate, the quality of publications may vary widely, and in some cases, the quality assessment given by the researchers themselves might not be realistic. Therefore, it is desirable to obtain evaluation data from a database with some kind of built-in quality control, which also allows various queries at any time without involving the researchers again.

In some scientific areas, there are internationally recognized publication collection systems that completely cover their respective areas. In engineering sciences with interdisciplinary aspects, this is frequently not the case, and often no single publication collection system exists that is appropriate for a comprehensive evaluation. Furthermore, the publication collections permit to search for publications of a particular author, but usually they have no provisions for extracting publication counts and lists for groups of scientists or entire organizational units, which frequently is required in evaluation schemes. Finally, a complete representation of the work performed at a university also comprises less "official" publications like academic theses or reports, which are by design ignored by standard publication collections.

In spring 1999, therefore, the Faculty of Electrical Engineering and Information Technology at the Technical University Vienna decided to custom-design a publication database to support the allocation of resources dependent on reliably determined publication output. Since a quick solution was required, we chose *Microsoft Access* for the prototype version of this database. This prototype became operational after only a couple of months of development; it was introduced faculty-wide in late 1999. Some severe drawbacks of *Access* in a multi-user environment and the prospect of a much more powerful system led to the development of a Web-based database solution with a LAMP (Linux – Apache – MySQL – PHP) approach. Based on the concept of and the experience gathered with the *Access* prototype, and under the first author's supervision, a group of four students developed the initial code of the Web-based database, which took more than one year due to the complexity of the task. Hence, the Web-based version became available only in mid-2001; almost two years after the *Access* prototype had been ready for use, and after 13 version releases of the *Access* database. More than 3600 publication records were migrated from the *Access* prototype to the Web-based publication database.

From the very beginning of this project, one single person, the first author of this paper, has been executively in charge of the architecture, implementation and programming of the publication database. In close to 60 major and minor releases the size of its PHP program code has grown by a factor of five meanwhile caused by the implementation of a wealth of additional functions and improvements. This growth was partly due to additional evaluation functionality required by law or the university authorities, but to a greater degree because of enhanced usability and "added value" functions. Because the software met the expectations of the university authorities, the entire university adopted it in mid-2002. It provides all publication-related evaluation data of the university since.

THE CONCEPT OF THE PUBLICATION DATABASE

The design of a system like the publication database has to take into account two possibly conflicting requirements, the

[1] Karl Riedling, Institute of Sensor and Actuator Systems, Technical University Vienna, Vienna, Austria, karl.riedling@tuwien.ac.at
[2] Siegfried Selberherr, Institute for Microelectronics, Technical University Vienna, Vienna, Austria, siegfried.selberherr@tuwien.ac.at

completeness of the data held in the database, and ease of use for the intended users. Information in the database has to be as comprehensive and detailed as possible to allow for all conceivable queries, particularly, because these queries have to take into account the types and quality of the publications. Allowing the researchers or their secretaries to enter their publication data themselves results in total freedom with regard to which details of publications, and which publication types, the database can hold. This implies, however, that most users who create entries into the database are not familiar with the advanced aspects of bibliography, which precludes a "full-blown" bibliographic system. It demands, in contrast, a flexible approach where only self-explanatory fields essential for identifying and verifying a publication need filling in, while optional fields are available for additional information such as abstracts, keywords, or links to electronic versions of the publications.

Instruments that only serve the purpose to collect statistical data are generally not well accepted. Therefore, the publication database has to provide sufficient additional benefit to its users to improve its acceptance. A financial profit resulting from a publication-dependent allocation of resources is certainly a benefit, at least for the successful groups. An advantage for all users is the possibility that everybody can extract their own publication lists, even dynamically for use on a web site. A standardized reference format that greatly facilitates the preparation of project applications or departmental reports is an additional important benefit in creating publication lists from a database. Furthermore, external visitors must be able to freely search for information in the database, and data export must be possible into other research documentation or library systems. In fact, in providing the above benefits the publication database serves as a knowledge base.

To allow both to determine evaluation data and an operation as a knowledge base with the possibility to search for information, the database must support a wide range of publication types, including less "official" publications like internal reports or academic theses, and permit a simple extraction of counts and lists of publications based on a variety of query criteria. It must allow selection, grouping, listing, and rating of publications according to their types and properties, and according to various attributes of their publication media. This implies a genuine relational database structure, where each item of a publication entry is located in an individual field of a database table.

Publications jointly written by several authors affiliated to different organizational units are supposed to appear in the publication lists or evaluation data of each of its authors, and of each of the units to which the authors belong. To allow the selection of all publications of a particular group or institute, the names of persons must reside in a separate table of a relational database, with references to the groups and institutes (kept in other tables) to which these persons belong. In turn, the table of names is linked to the table of publications.

A consequence of this approach is that users must select the names of the authors from a list during the creation of a publication entry. For reasons of uniformity, the same applies to the names of editors of books or conference proceedings, of the reviewers or supervisors of doctor's or diploma theses, and of other persons involved in publications of some special types. Obviously, users must be able to add new names to the name table when creating a publication entry.

Maintenance of the information required for "weighing" publications should be as easy as possible: It simply would not make sense to have information such as the SCI status of a publication or the impact factor of the journal in which it appeared entered separately for each publication. These are properties of the "publication medium" (e.g., the journal), which properly belong into a publication medium record (see Figure 1). Hence, publication media have to be selected from a list similar to the names of authors, and are added to this list if they are not yet in the database. It should also be possible to tie together publication media with comparable properties and regard them as belonging to a specific "media type" that, in turn, determines their "weight" in an evaluation. For example, "journals listed in the SCI with an impact factor greater than 1" may constitute a particular media type. Journals and, e.g., conferences obviously cannot share media types; they therefore constitute different "media classes". The media classes recognized in the publication database are journals, publishing houses (for books and contributions to books or proceedings volumes), events (for talks or poster presentations at conferences or other scientific meetings), and patents. The publication media concept is not used for some publication types like academic theses or internal reports.

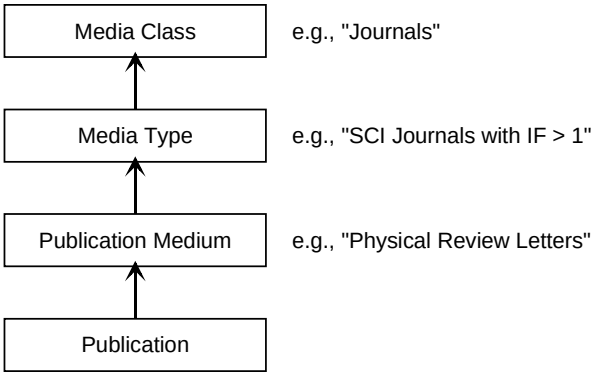

FIGURE 1
HIERARCHIC ORGANIZATION OF PUBLICATIONS IN THE PUBLICATION DATABASE

Our publication media concept greatly facilitates the quality control of the data entered: Instead of looking at classifications in hundreds of publication entries, only the classifications of the publication media require checking. Particularly in the case of journals, the number of publication media grows only slowly after an initial phase, and it is easy to look up these newly added journals in the proper databases.

Different types of publications imply different items of information in their database records, and different output formats. It makes therefore sense to define "publication types": A publication type determines not only the number and

meaning of data fields and the output data format; it also determines the media class to which the publication media offered for selection must belong.

This structure results in the ER (*entity-relationship*) diagram shown in Figure 2, which is a simplified representation of the actual table structure of the publication database. Currently, the database comprises more than 50 tables most of which are related with one another. Figure 2 does not show the numerous tables that hold auxiliary information such as the formatting of the reference output, the grouping of publication types in publication lists, or the evaluation queries and results, and it also shows only one relation that determines the "owner" of a publication entry (i.e., the person who made the entry). All tables that regular users can modify hold, in addition to "owner" fields, similar fields that permit to determine who the last person to change the entry was.

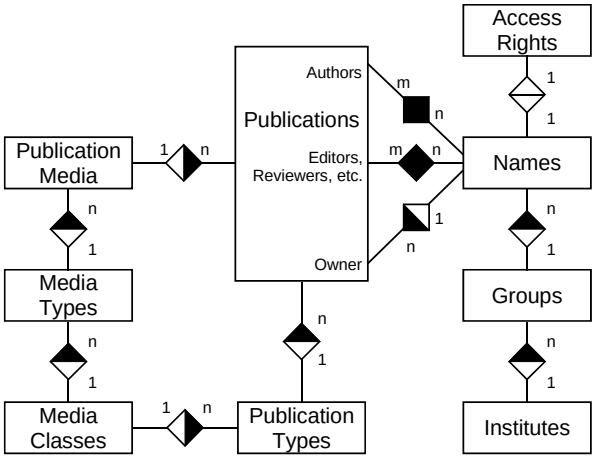

FIGURE 2
SIMPLIFIED ER (ENTITY-RELATIONSHIP) DIAGRAM OF THE PUBLICATION DATABASE

The concept to keep as much configuration information as possible in database tables, which had already been introduced in the *Access* prototype, proved to be exceedingly beneficial. No changes of the program code are necessary, e.g., to introduce new publication types or change the publication reference format; this requires only adding or modifying records in the publication type and the formatting tables. The core table structure as shown in Figure 2 has remained unchanged through the life of the database; however, added functionality necessitated many new fields in some of these tables, and additional auxiliary tables.

THE IMPLEMENTATION OF THE PUBLICATION DATABASE

For the replacement of *Access*, whose shortcomings in a multi-user environment dictated a different solution in any case, a Web-based solution appeared favorable over any other client-server concept:

- One has to deal with a wide range of hardware and operating system platforms at a university. This rules out dedicated LAN-based clients.
- The publication database should be a sustainable solution that should exceed the lifetime of common client software applications.
- Maintenance should be easy. Upgrading a Web-based system requires no software distribution to the clients.
- Using the database as a knowledge base implies external access to the publication information. This calls for a web interface anyway.
- A web interface allows implementing web services which can contribute to an integration of the database with other related systems.

In general, using the HTTP or secure HTTP protocols for transport and conventional web browsers as clients makes the database platform-independent and worldwide accessible. Browser-independent programming is mandatory since university staff tends to use a variety of browsers, including some "exotic" species.

For primarily financial, but also for technical reasons, we chose a LAMP structure for the database server, with client-based JavaScript for local pre-processing.

Despite the use of client-side programming, the implementation of the database keeps most of the processing in the server-based PHP code. This facilitates software management and guarantees a secure and reliable processing environment. All potentially security-related functionality resides in server-side PHP code. Most of the JavaScript code in the publication database serves only for enhancing the usability of the user interface, for example by presetting certain form elements after modifications of other elements. Other important JavaScript-based features are a quick search through long lists of person or media names, or checking the completeness of an input form. Although the client-side code uses only the most established JavaScript features, problems with some browsers made it advisable to convert the initially rather extensive client-side JavaScript data pre-processing code into PHP code wherever possible. The introduction of new browsers calls for repeated testing of the JavaScript and HTML rendering functionality. Occasionally, browser bugs or a non-standard browser behavior made code modifications necessary. With one exception – the display of Greek characters –, no browser-dependent programming is used, though.

Various entry points permit access to the publication database:

- An authenticated "administration module" for data input and maintenance;
- Several interactive public interfaces that allow searching for publications and/or creating publication lists of persons, groups, or institutes covering arbitrary time ranges, publication types, or publications that match some other criteria;
- Functions that dynamically create HTML pages with similarly tailored publication lists in a custom design for the inclusion on other web sites;
- Features to export publication data in HTML, ASCII text, T_eX or XML formats; and

- Web services that prepare on demand data output in various formats, based on diverse dynamically chosen selection criteria. Complementary to the web services presented by the publication database, the database itself invokes web services provided by other systems. This approach allows portable and platform-independent real-time data exchange with other databases and results *de facto* in an integration of research-related data collections.

The administration module requires client-side JavaScript and at least *Netscape* 4 or *Internet Explorer* 4. In contrast, the public interactive interfaces are also functional without JavaScript (although they have a smoother user interface on JavaScript-enabled browsers); in fact, even *lynx* can display the public interfaces. Currently, the administration module is in German only, but it permits to create publication lists in English and German. The public interfaces are available in English and German, and the web services likewise provide bilingual data where necessary.

The interactive public interfaces permit to set a variety of query conditions, and generally create human-readable lists in HTML format of the matching publications. In addition, the administration module and the web service functions also support ASCII, TeX, or XML-based output.

Various query functions in the interactive interfaces permit restricting a search to entries meeting certain conditions, e.g., the affiliation of at least one author or essentially involved person to a particular organizational unit; publication years; publication types, and many more (Figure 3). For most publication types, only the affiliation of the authors matters; for some, such as academic theses, an entry is selected if either the author or the supervisor of the thesis belongs to the unit chosen. All interactive interfaces provide full-text search functions, which may optionally take into account the entire record including abstracts etc., or only certain fields of the record.

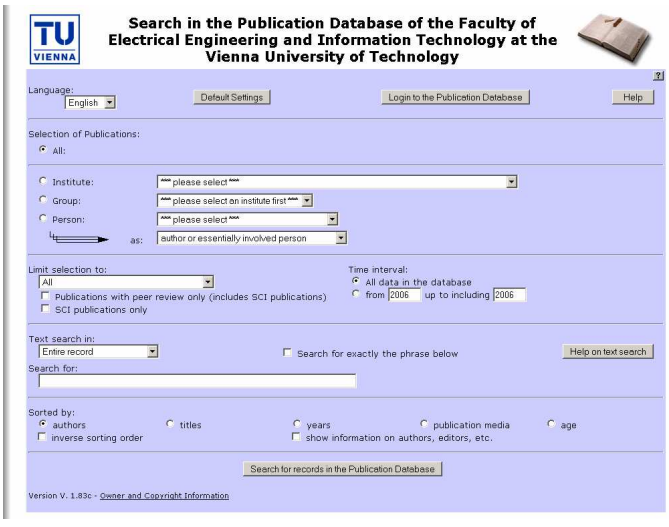

FIGURE 3
ONE OF THE INTERACTIVE SEARCH FUNCTIONS OF THE PUBLICATION DATABASE

The authenticated administration module of the database uses multi-level access privileges: Users can have permissions to edit their own publication entries, those belonging to their groups, or the entries of their institutes. "Their own" means entries created by the particular user, plus all entries in which this user appears as an author. These rights may extend analogously to the publications of group or institute members. Administrators can edit any entry in the database, plus administrative parameters. Separate privilege attributes permit users to change evaluation-specific parameters or to perform very resource consuming complex evaluation queries. Since permissions for editing a publication also depend on the relation of the user of the administration module to at least one of the authors, the table where the access rights are stored is closely linked to the table that holds the names of persons shown in publication entries (see Figure 2).

The database creates statistics and evaluation data according to two different schemes. One scheme accounts for the "official" evaluation algorithms, essentially simple counts of publications in specific categories. An experimental algorithm [1] allows a more detailed weighing of publications; it is not regularly used, though.

The statistics and evaluation queries are frequently rather complex and must be repeated easily and reproducibly for a large number of different queries and organizational units. A special function in the administration module allows easy creation and modification of the queries; they are stored in special database tables. Simple queries may consist of an arbitrary number of close to 30 conditions, which are AND-combined, and select publications that belong to one of a set of specified publication and media types. The conditions may pertain to properties of the publications, the publication media, or the authors. Complex queries are an OR-combination of several simple queries. Only administrators may edit the queries, but any user of the administration program can inspect them and carry them out one by one. For selected users, a special page is available that allows bulk execution of a set of queries applied to a number of organizational units; the results of such queries are available in a CSV format compatible with, e.g., *Microsoft Excel*.

Additional functions of the administration module comprise various database maintenance and integrity testing functions; functions for extracting evaluation data; and a tool to create URLs for inclusion on other web sites that request a certain selection of publication data from one of the web services of the database. While the URL generator is available to all users of the administration module, only administrators or specially privileged users may access the other functions.

The original design of the publication database supported only one faculty. When the university authorities introduced it university-wide, we decided to implement one separate copy of the database for each of the faculties. The resulting ten databases reside on the same physical server; they are accessed via the virtual web server concept of Apache. Although the maintenance of ten separate databases requires more effort, several reasons favored the solution chosen:

- Users need not enter publication data for a faculty other than their own. It does not matter whether they log into a university or a faculty publication database.
- Evaluation data are gathered primarily on a faculty base. Splitting the database in the way chosen does not constitute a problem for evaluation schemes.
- Faculties may want to use individual configurations of the database. This is easier to implement in separate copies.
- Users must select author and publication media names from lists of already registered records. These lists grow rapidly: In the EE Database, which holds the faculty's publications from 1996 on, there are currently about 6,000 name and 3,300 media entries (for more than 10.000 publications). Using only one database for the entire university would increase these numbers by a factor of 4 to 5, which makes selecting suitable name or media entries from lists impractical.
- The drawback that external visitors would have to search in several databases could easily be resolved by introducing a portal that transparently searches all databases in turn.

Apart from a single file that defines configuration parameters specific for one particular faculty database, all copies of the database use the same set of PHP, HTML and image files. This reduces software updates to a copy operation in batch mode, and hence makes them rather straightforward.

OPERATION OF THE PUBLICATION DATABASE AT THE TECHNICAL UNIVERSITY VIENNA

The quality and reliability of the data collected in the publication database depend on two important factors, user acceptance and quality control by organizational and technical means. Bad user acceptance results in careless entries, which in turn demand more convoluted quality control mechanisms. Providing sufficient additional benefit to the users to increase their acceptance of the system therefore enhances the effect of the quality control procedures.

The first reactions of the users to the introduction of the publication database ranged between suspicion and hostility; they regarded the database as an instrument designed to increase rather than reduce their workload. An important point we made was that in future all publication-related evaluation data would come from the database, thus sparing them several such surveys per year. Furthermore, there was the additional benefit of on-line publication lists and queries, and the increased visibility of their work. The Faculty of Electrical Engineering and Information Technology introduced a financial bonus for institutes and first authors of high-quality publications derived from the database data, which was not only a strong incentive for publishing and officially documenting published work, but also a tangible benefit that made the reception of the database much more favorable.

In addition to the proper fault-free operation of the publication database and the addition of new functionality required by law, by the university authorities, or due to the wish to make optimal use of the data in the database, the usability of its user interface has been the most important design issue. Often, seemingly insignificant features greatly facilitate work for the users, e.g., the possibility to limit searches to entries that still require some kind of action, or to sort entries by age (with the latest on top of the selection list). In some cases, log files allowed insight into user behavior, which resulted in a re-design of some functions. For example, it took plenty of real-user experience to find a proper strategy, when to warn users that they were using selection restrictions to the data they were operating on, and when not to bother them with a warning popup. There are obviously "cultural" differences between the faculties even at a university with exclusively technical orientation: In some cases, program messages that appeared clear enough to a large part of the users needed re-wording, because users belonging to some groups consistently misunderstood them. Feedback from the users is generally taken very seriously; it has greatly contributed to the user-friendliness of the database.

The mentioned "cultural" differences between the faculties also resulted in widely differing user expectations and wishes when we introduced the database university-wide. Common were requests from institutes that already had publication collections of some kind for an import function for their collections. We developed a tool for data import in a variety of formats, which, however, hardly was used when it became available.

After the initial opposition had subsided, people learned quickly to take full advantage of the database. As the number of publication entries grows, more and more institutes and groups use the database as a source for publication lists displayed on their web sites. The database provides these lists through a number of web services. Although only publications beginning with 2002 (1996 at the Faculty of Electrical Engineering and Information Technology) are required to be held in the database, many institutes also have entered their earlier publications meanwhile to allow the creation of their complete publication lists. Lately, the XML service has found more and more acceptance by groups who not only process the XML data for custom-designed output on their web sites, but also want to create publication references in formats not directly supported by the publication database, such as BibTeX. In addition, the university library periodically imports the data collected in the database into their own library system [2].

Several automatic features and human actions guarantee high data quality, which is of equally high importance for both research documentation and evaluation purposes: Algorithms test for the proper contents of required fields and check for duplicates of new or existing entries. They report a possible duplicate, if at least two of four properties – titles, lists of authors, publication media, and page counts – match for two entries. Author lists are created by selecting names from a list; it is sufficient to test them for identity, which also applies to page counts. Title and media name strings, in contrast, may differ even for genuine duplicates due to typing errors or abbreviations; a simple test for identity is not sufficient in this case. A naive approach that considers titles as matching if one

title string completely contains the other was more efficient in finding duplicates than the test for identity, but still far from satisfactory. Therefore, we introduced a "similar string" algorithm for the comparison of titles and media names. The most efficient approach to search for similar strings in PHP [3] is the Levenshtein algorithm [4], [5], which returns the number of characters that have to be added, changed or removed to transform one of the strings into the other. A Levenshtein distance of less than a string length dependent limit constitutes a match. However, the Levenshtein algorithm is rather resource consuming. The publication database uses a smart restriction to those publication types and publication years where duplicates might perceivably exist to make the performance of this algorithm acceptable for routine use. Reports of duplicates are only warnings without automatic consequences; the decision whether reported possible duplicates are real ones, and which consequences have to be taken, is left to the user or administrator who initiated the check. In addition to the automated tests, a specifically assigned person validates entries of print publications based on submitted reprints, which may optionally be in electronic form. Finally, a group of senior researchers checks the semantic correctness of publication entries and their proper media type associations.

The design concept that allows formulating, storing, and executing an unlimited number of groups of evaluation or statistics queries proved to be extremely beneficial: Not only do the legally required evaluation schemes change repeatedly, there are also several other statistical inquiries that may apply to the data of one faculty only or of the entire university. Having a set of versatile queries at hand, and having the possibility to define new queries easily if required, reduces the time needed for answering specific questions from weeks to hours.

The database allows the addition of keywords or abstracts in English and German into the publication records, and permits to upload files of electronic versions or to reference them via web links. Actually, users may upload or reference two files for each publication record: A publicly visible version, which is feasible if there are no copyright restrictions to a publication, and a "hidden" version that is only accessible from within the administration module, and allows the validation of publication entries with possibly copyright-protected electronic versions. In addition to the basic publication reference data, the university library receives the contents of the abstract fields and the references to public and hidden files. Abstracts are transferred into the library system, and referenced files are copied to a literature server where appropriate. In addition to serving as a knowledge management system on its own, the publication database also acts therefore as a knowledge collection tool for the university library.

The publication database is one of several systems at the Technical University Vienna that document various aspects of research and teaching. For historical and technical reasons, these systems are separate from one another, but they are closely related. For example, the publication database permits the association of publications with projects, which reside in a separate database. Web pages or web services on either side allow displaying publications linked to a particular project, and vice versa. The publication database has to maintain its own tables for authors and users, because more than three quarters of the person entries of the publication database belong to external authors rather than university staff. However, it obtains staff IDs from the university's staff database via a web service that is only invoked for persons who were declared in the publication database to be members of an organizational unit of the university. The concept of using separate but strongly interoperating databases for separate tasks, rather than a large unified database, has the advantage that the individual databases can be uncompromisingly optimized, and, if necessary, upgraded or replaced without much adverse effect on the entire system. From the point of view of external visitors, the interoperating databases behave like a single complex database.

CONCLUSIONS

The publication database presented has been in use at the Technical University Vienna for seven years, first at the Faculty of Electrical Engineering and Information Technology only, later at the entire university. It has gradually grown during this time from a stand-alone evaluation instrument with the facility to generate publication lists to a comprehensive knowledge base for publication data that closely interacts with several other related databases. University institutes, external visitors, and, last but not least, robots of search engines increasingly make use of its facilities, which contributes to an enhanced visibility of the database contents in the scientific community, and to a growing acceptance by researchers at our university.

REFERENCES

[1] Riedling K.: "Design and Implementation of a Publication Database for the Vienna University of Technology"; Talk: VIEWDET 2003, Vienna, Austria; 11-26-2003 – 11-28-2003; in: "*Proceedings of the Vienna International Conference on eLearning, eMedicine and eSupport (VIEWDET 2003)*", E. Riedling (ed.); Institute of Industrial Electronics and Material Science, (2003), ISBN: 3-85465-013-2; 10 p.

[2] Hrusa H., Kirschner Ch., Neumayer F.: "Datenimport aus der TU Publikationsdatenbank in den Aleph-Bibliothekskatalog"; *Mitteilungen der VÖB*, **58** (2005), 2; 21 – 29 (in German).

[3] Hojtsy G. (ed.): "PHP Manual"; PHP Documentation Group, *http://www.php.net/docs.php*, 2004.

[4] Levenshtein V.: "Binary codes capable of correcting deletions, insertions, and reversals"; *Soviet Physics Dokl*. **10** (1965), 707– 710.

[5] Bogomolny A.: "Distance Between Strings"; *http://www.cut-the-knot.org/do_you_know/Strings.shtml*, 2006.