A Flexible Web-Based Publication Database

Karl Riedling[1] and Siegfried Selberherr[2]

[1] Institute of Sensor and Actuator Systems

[2] Institute for Microelectronics

Technische Universität Wien
Gusshausstrasse 27-29
A-1040 Wien, Austria

Email: {Karl.Riedling | Siegfried.Selberherr}@TUWien.ac.at

To allow allocation of resources dependent on reliably determined publication output, the Faculty of Electrical Engineering and Information Technology of the TU Vienna decided to custom-design a publication database, which later was adopted by the entire university. To make full use of the wealth of information stored in such a database, the chosen concept includes features that support its application as a knowledge management and research documentation system. The development of the database over a period of seven years from a stand-alone tool for collecting evaluation data to its current state as one of the university's knowledge bases is described.

Introduction

The scientific community commonly measures the quality of scientific work by judging the resulting published output. Reliably determining this output is, however, not a straightforward task: The quality of publications may vary widely, data provided by researchers are not in all cases accurate, 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 supports diverse queries at any time without involving the actual researchers.

In some scientific areas, there are internationally recognized publication collection systems that entirely cover their respective areas. In engineering sciences with interdisciplinary aspects, this is frequently not the case, and often no publication collection system can be identified 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, e.g., publication counts and lists for a group of scientists or an entire organizational unit, which frequently is a requirement 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 our university decided to custom-design a publication database to permit allocation of resources dependent on reliably determined publication output. Since a quick solution was required, *Microsoft Access* was chosen for the prototype version of this database. The *Access* prototype consisted of two modules: A GUI front-end with a number of VB script modules as a user interface that allowed easy upgrading, and a back-end that held the publication data. This database became operational after only a couple of months of development. After a few more months of test operation at the authors' respective institutes it was introduced faculty-wide in late 1999, first with separate copies of the database for each of the about fifteen institutes of the faculty, later with one common server-based installation. Some severe drawbacks of using *Access* for a multi-user application 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 code of the Web-based database, which took more than one year due to the complexity of the task. Hence, their version became available only in mid-2001; almost two years after the *Access* prototype had been ready for use. After 13 version releases of the *Access* database, the Web-based database took over the more than 3600 publication records and the tasks of the prototype publication database.

Since the very beginning of this project, one single person, the first author of this paper, has been executively in charge of the architecture and implementation of the publication database. A wealth of additional functions and improvements have been implemented meanwhile. In close to 60 major and minor releases of the database, its PHP program code size has grown by a factor of five during the five years since the introduction of the Web-based version. This growth was partly due to additional evaluation functionality required by the 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, it was adopted by the entire university in mid-2002, and provides all publication-related evaluation data of the university since.

The Basic Concept of the Publication Database

The design of a system like the publication database has to take into account two possibly conflicting requirements: Information in the database has to be as complete and detailed as possible to allow for all conceivable queries which should not only result in a simple count of publications but should also 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. However, entries into the database are often made by persons not familiar with the detailed aspects of bibliography. This precludes a "full-blown" bibliographic system and demands a flexible approach where only the 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.

In general, instruments that only serve the purpose to collect statistical data are not well accepted. In order to improve the acceptance of the publication database, it has to provide sufficient additional benefit to its users. Apart from the financial implications, at least for the successful groups, which result from a publication-dependent allocation of resources, an additional advantage is that, e.g., everybody is able to extract their own publication lists or have them created dynamically for use on a web site. A standardized reference format, which greatly facilitates the preparation of project applications or departmental reports, is one of the important benefits in creating publication lists from a database. Furthermore, external users 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, the publication database must serve as a knowledge base to provide the desired benefits.

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 feature simple extraction of counts and lists of publications based on a variety of query criteria. It must be possible to select, group, list, and rate publications according to their types and properties, and according to various attributes of their publication media. This implies a genuine database structure, where each item of a publication entry is located in an individual field of a database table.

Several authors affiliated to different organizational units may jointly have written a publication, which is supposed to appear in the publication lists or evaluation data of each of its authors, and of each of the units to which its 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, linked to the table of publications, with references to the groups and institutes to which these persons belong (Fig. 1).

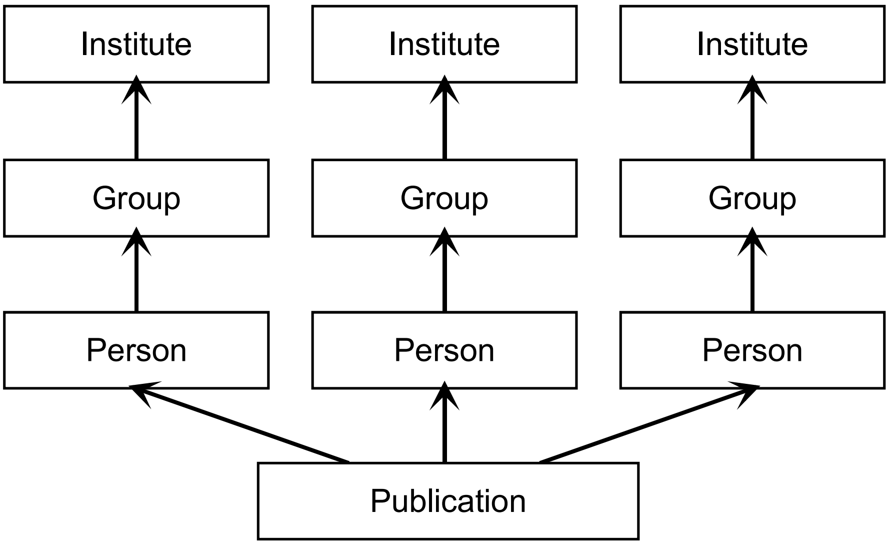


Fig. 1: Hierarchic organization of persons linked to a publication

As a consequence of this approach, 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 editors of books or conference proceedings, the reviewers or supervisors of doctor’s or diploma theses, and other persons involved in publications of some special types. Obviously, it must be possible for users to add new names to the name table in the course of creating a publication entry.

“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 Fig. 2). Similar to the names of authors, publication media have to be selected from a list, 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 a comparable quality and regard them as belonging to one 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. Since journals and, e.g., conferences obviously cannot share media types, they 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.

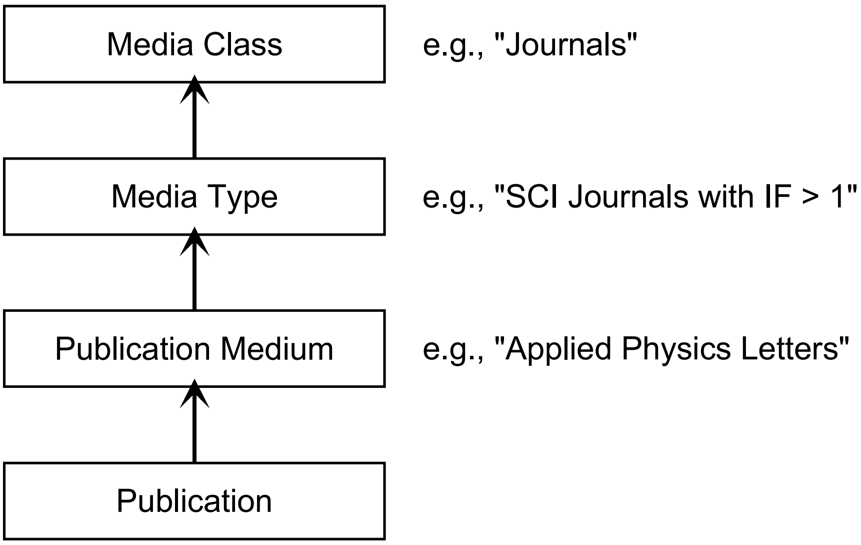


Fig. 2: Hierarchic organization of publications in the publication database

The publication media concept greatly facilitates a sanity check of the data entered: Instead of looking at the classifications in hundreds of publication entries, only the classifications of the publication media need 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 require different information items to be kept in their database records, and different output formats. It makes therefore sense to define “publication types”: A publication type determines not only the data format; it also determines the media class to which the publication media offered for selection must belong.

This structure results in the ER diagram shown in Fig. 3, which is a simplified representation of the actual table structure of the publication database. Figure 3 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 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.

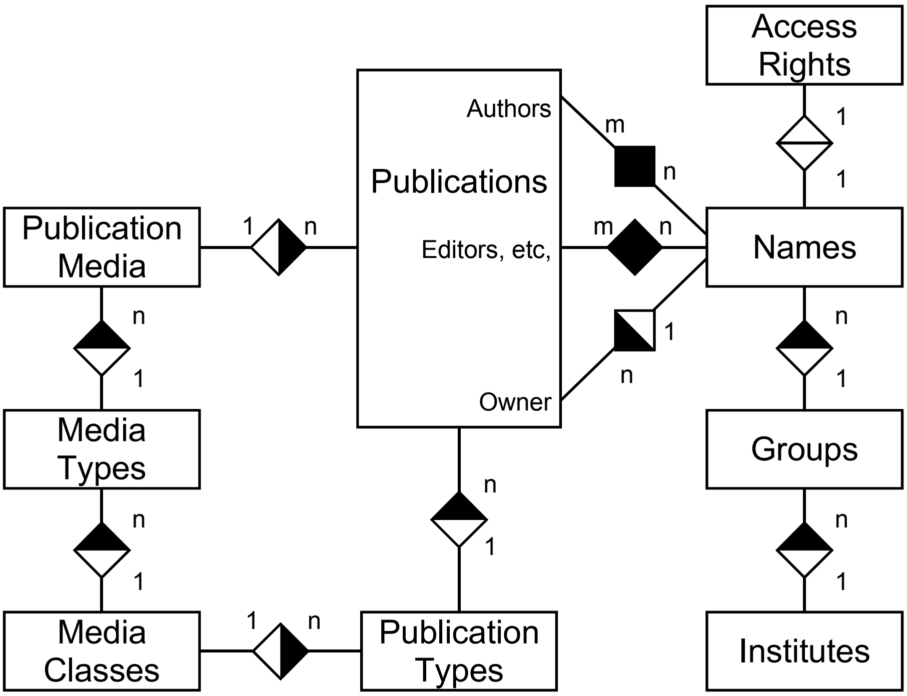


Fig. 3: Simplified ER diagram of the publication database

The concept already introduced in the *Access* prototype, namely, to keep as much configuration information as possible in database tables, proved to be exceedingly beneficial: No changes of the program code are necessary to introduce, e.g., new publication types; this requires only adding records to the publication type and the formatting tables. The core table structure as shown in Fig. 3 has remained unchanged through the life of the database; however, many new fields were added to some of these tables, and added functionality required additional auxiliary tables.

The Implementation of the Publication Database

While the shortcomings of *Access* in a multi-user environment dictated a different solution in any case, other constraints favored a Web-based solution over any other client-server concept:

- We were looking for a sustainable solution that should exceed the lifetime of common client software applications.
- At a university, one has to deal with a wide range of hardware and operating system platforms. This precludes dedicated LAN-based clients.
- Maintenance should be easy. No software need be distributed to the clients when a Web-based system is upgraded.
- Using the database as a knowledge base, which implies providing external access to the publication information, requires a web interface anyway.
- A web interface allows the implementation of web services, which can help to integrate the database with other related systems.

In general, using conventional web browsers as clients and the HTTP or secure HTTP protocols for transport makes the database platform-independent and worldwide accessible. Since university members tend to use a variety of browsers, including some “exotic” species, browser-independent programming is mandatory.

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

The program structure chosen keeps most of the processing in the server-based PHP code. This facilitates software management and provides a secure and reliable processing environment. All potentially security-related functionality resides in server-side PHP. Most of the JavaScript code in the publication database is only there to enhance the usability of the user interface. One example is presetting certain form elements after modifications of other elements. Other important 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 necessitates repeated testing of the JavaScript and HTML rendering functionality, and occasional code modifications in the case of a non-standard browser behavior or browser bugs. With one exception – the display of Greek characters –, no browser-dependent programming is used, though.

There are various access points to the publication database:

- An authenticated access for data entry and maintenance (the “administration module”);
- Several interactive public interfaces that allow searching for publications and/or creating tailored publication lists of persons, groups, or institutes;
- A number of functions that dynamically create HTML pages with publication lists in a custom design for inclusion on other web sites;
- Features to export publication data in HTML, ASCII text, TeX or XML format; and
- Web services presented by the publication database that prepare on demand data output in various formats, based on diverse dynamically chosen selection criteria. Likewise, the database invokes web services provided by other systems. This approach allows platform-independent and portable real-time data exchange with other databases of the university, and results in a *de facto* integration of all research-related data collections.

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. While the administration module requires client-side JavaScript and at least *Netscape* 4 or *Internet Explorer* 4, the interactive public interfaces can also operate without JavaScript (although they have a smoother user interface on JavaScript-enabled browsers); in fact, even *lynx* can display the public interfaces.

While the interactive public interfaces generally create human-readable lists in HTML format of those publications that match the query conditions, the administration module and the web service functions also support ASCII, TeX, or XML-based output.

In the interactive interfaces, various query functions 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. For most publication types, only the affiliation of the authors is taken into account; 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 process the entire record including abstracts etc., or only certain fields of the record.

The authenticated administration module of the database uses a multi-level access privilege scheme: At the lowest level, users may create publication entries and edit their own entries (where "their own" means those that they entered themselves, plus all entries in which they appear in the list of authors). The next level extends the editing rights to all publication entries created by members of the group the user belongs to, or with authors belonging to this group. The third level analogously extends these rights to the user's institute. At the highest level is the administrator who can edit any entry in the database, including administrative parameters. Separate privilege attributes permit users to change evaluation-specific parameters or perform complex (and therefore resource-consuming) 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 authors and other persons shown in publication entries (see Fig. 3).

The database supports two different schemes for obtaining statistics and evaluation data: one that accounts for the "official" evaluation algorithms, which are based on simple counts of publications in specific categories, and an experimental one that, among others, takes also the page count of publications into consideration, giving therefore greater weight to the larger of two publications in comparable media[1]. The experimental algorithm is not regularly used, though.

The statistics and evaluation queries are frequently rather complex and must be repeated reproducibly for a large number of different queries and organizational units. To facilitate their management, they are not hard-coded, but can be dynamically created and edited through an interface in the administration module. Special database tables accommodate the query information. Simple queries may contain 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 attributes of the publication, the publication media, or the authors. Complex queries are an OR-combination of any number of 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. A special page is available to selected users that allows executing a set of queries applied to a number of organizational units in a bulk mode; the results of such queries can be exported 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 Publication Database was originally designed for use by one faculty only. 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 and are accessed via the virtual web server concept of Apache. Although the maintenance of ten separate databases requires more effort, compared to one database for the entire university, several reasons favored the solution chosen:

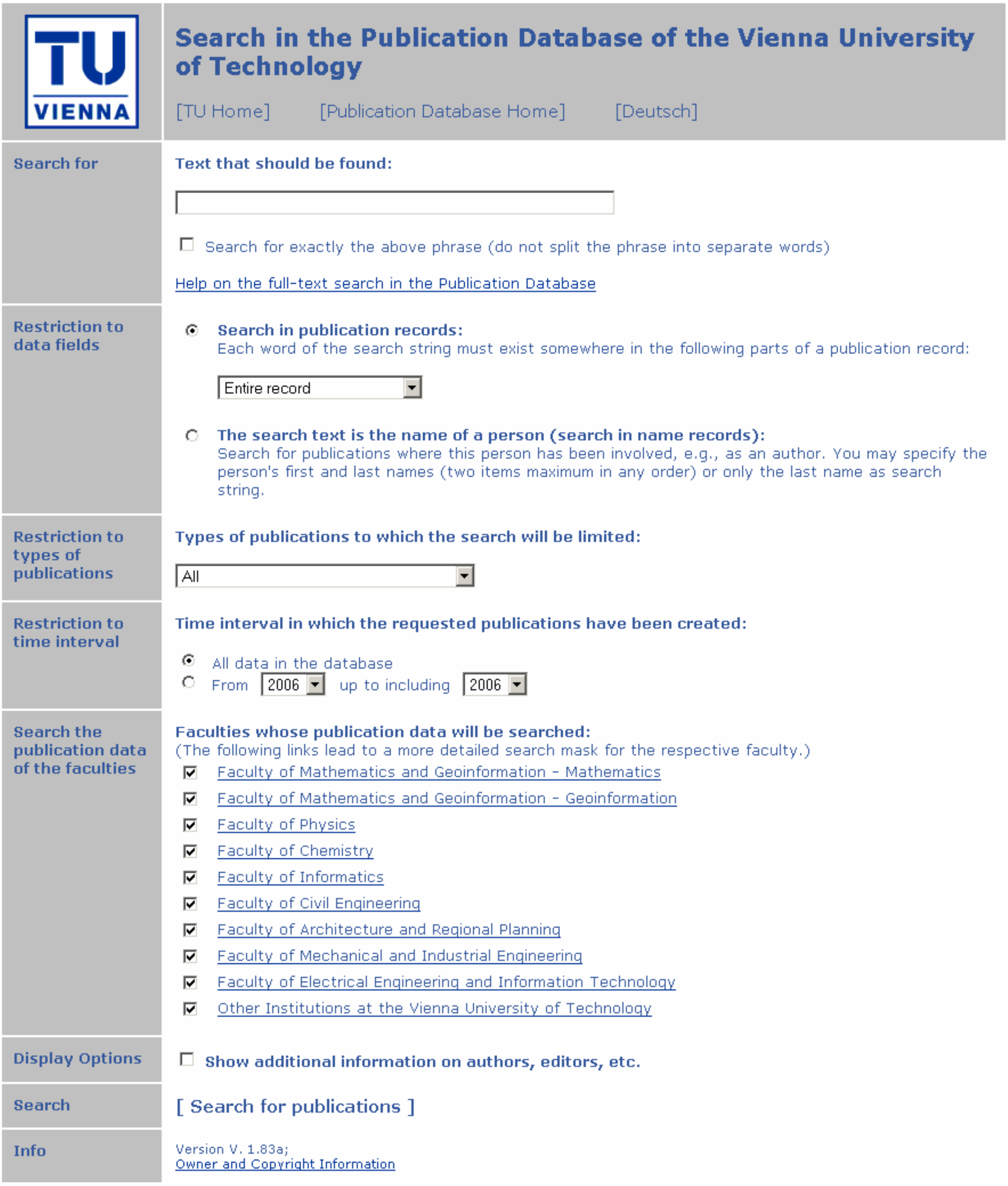
- It does not make a difference for people entering publication data, whether they log into a university or a faculty publication database.
- Evaluation data are primarily gathered on a faculty base. Splitting the database in the way chosen does not constitute a problem for evaluation schemes.
- Faculties may want to use individual configurations of the database. This is much easier to implement in separate copies.
- The lists of already registered authors and of the publication media with a suitable media class from which users must select entries 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 of such a size impractical.
- The drawback that external visitors would have to search in several databases, where the publications they were looking for might appear, could easily be resolved by introducing a portal that transparently searches all databases in turn (see Fig. 4).

Apart from one configuration file which defines configuration parameters specific for the particular faculty database, all copies of the database use the same set of PHP, HTML and image files. This makes software updates rather straightforward, although PHP and Linux do not permit to install the common file tree only once and access it via symbolic links.

Operation of the Publication Database at TU Vienna

As was to be expected, users initially met the publication database with (at least) suspicion, as an instrument designed to increase their workload. We could alleviate their objections by promising that all publication-related evaluation data would come from the database, without bothering them with such surveys in the future, and by pointing out the additional benefit of on-line publication lists and queries and the increased visibility of their work. A financial bonus for institutes and first authors of high-quality publications derived from the database data, introduced at the Faculty of Electrical Engineering and Information Technology, 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.

Particularly after its university-wide introduction, there were widely differing user expectations in the database: At the same day, two researchers claimed that there were too many and to few data fields, respectively. Because of urgent requests from institutes that already had publication collections of some kind, an import function for publication collections in a variety of formats was developed, which, however, hardly was used when it finally became available.



TU VIENNA	**Search in the Publication Database of the Vienna University of Technology** [TU Home] [Publication Database Home] [Deutsch]
Search for	**Text that should be found:** □ Search for exactly the above phrase (do not split the phrase into separate words) Help on the full-text search in the Publication Database
Restriction to data fields	● **Search in publication records:** Each word of the search string must exist somewhere in the following parts of a publication record: Entire record ○ **The search text is the name of a person (search in name records):** Search for publications where this person has been involved, e.g., as an author. You may specify the person's first and last names (two items maximum in any order) or only the last name as search string.
Restriction to types of publications	**Types of publications to which the search will be limited:** All
Restriction to time interval	**Time interval in which the requested publications have been created:** ● All data in the database ○ From 2006 up to including 2006
Search the publication data of the faculties	**Faculties whose publication data will be searched:** (The following links lead to a more detailed search mask for the respective faculty.) ☑ Faculty of Mathematics and Geoinformation - Mathematics ☑ Faculty of Mathematics and Geoinformation - Geoinformation ☑ Faculty of Physics ☑ Faculty of Chemistry ☑ Faculty of Informatics ☑ Faculty of Civil Engineering ☑ Faculty of Architecture and Regional Planning ☑ Faculty of Mechanical and Industrial Engineering ☑ Faculty of Electrical Engineering and Information Technology ☑ Other Institutions at the Vienna University of Technology
Display Options	□ **Show additional information on authors, editors, etc.**
Search	[Search for publications]
Info	Version V. 1.83a; Owner and Copyright Information

Fig. 4: Portal for the search in all faculty publication databases

After the initial opposition had cooled down, people learned quickly to take full advantage of the database. 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 complete publication lists for their web sites.

Several automatic features and human actions guarantee high data quality, which is of equally high importance for both evaluation and research documentation purposes: Algorithms test, whether all required fields are properly set, and check for duplicates of new or existing entries. A possible duplicate is reported if at least two of four properties – lists of authors, titles, publication medium, and page count – match for two entries. It is sufficient to test author lists, which are created by selecting names from a list, and page counts for identity. However, title and media name strings, which may differ even for genuine duplicates due to typing errors or abbreviations, are compared with a Levenshtein algorithm[2,3], which returns the number of characters that have to be changed to transform one of the strings into the other. Although the Levenshtein algorithm is rather resource-consuming, it is the most efficient approach implemented in PHP to search for similar strings[4]. A smart restriction to those publication types and publication years where duplicates might perceivably exist makes the performance of this algorithm acceptable for routine use. A Levenshtein distance of less than a string length dependent limit constitutes a match. In addition, titles also match if one title string completely contains the other. Reports of duplicates are only warnings without automatic consequences; the decision whether reported possible duplicates are real ones is left to the user or administrator who initiated the check. In addition to the automated tests, a specifically assigned person validates the entries based on submitted reprints, optionally in electronic form. Finally, a group of senior researchers checks the semantic correctness of publication entries and their proper media type associations.

As the number of publication entries grows, the database is increasingly used as a source for publication lists displayed on the web sites of institutes and groups. These lists are obtained through one of the publication list web services of the database. 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 create publication references in formats not yet supported by the publication database, such as BibTeX. Furthermore, the university library periodically imports the data collected in the database into their own library system[5].

The database allows entering abstracts in English and German or keywords into the publication records, and permits uploading files of electronic versions or referencing 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 can only be accessed from within the administration module, and is used for validating publication entries with possibly copyright-protected electronic versions. In addition to the basic publication reference data, the 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 TU Wien that document various aspects of research and teaching. For historical and technical reasons, these systems are separate from one another, but not unconnected. For example, the publication database permits to associate projects, which reside in a separate database, with publications. Web pages or web services on either side allow displaying publications linked to a particular project, and vice versa. Likewise, the publication database, which has to maintain its own tables for authors and users, obtains staff IDs from the university’s staff database via a web service. Actually, more than three quarters of the person entries of the publication database belong to external authors rather than university staff; the web service is therefore 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.

The design concept that allows an unlimited number of groups of evaluation or statistics queries to be formulated, stored, and executed proved to be extremely beneficial: Not only do the legally required evaluation schemes change repeatedly, there are also several other statistical inquiries that may comprise 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.

Apart from 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, some seemingly insignificant features greatly facilitate work for the users, e.g., the possibility to sort entries by age (with the latest on top of the selection list), or to limit searches to entries that still require some kind of action. In some cases, studying of log files allowed insight into user behavior, and resulted in a re-design of some functions. It took plenty of real-user experience to find a proper strategy, e.g., when to warn users that they were using 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 of a university with exclusively technical orientation: In some cases, program messages that appeared clear enough to a large part of the users had to be reworded, because some other users consistently misunderstood them. Feedback from the users is generally taken very seriously; it has greatly contributed to the user-friendliness of the database.

Conclusions

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

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

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

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

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