DBLP — Some Lessons Learned *

Michael Ley
Universität Trier, Informatik
D–54286 Trier
Germany
ley@uni-trier.de

ABSTRACT

The DBLP Computer Science Bibliography evolved from an early small experimental Web server to a popular service for the computer science community. Many design decisions and details of the public XML-records behind DBLP never were documented. This paper is a review of the evolution of DBLP. The main perspective is data modeling. In DBLP persons play a central role, our discussion of person names may be applicable to many other data bases. All DBLP data are available for your own experiments. You may either download the complete set, or use a simple XML-based API described in an online appendix.

1. INTRODUCTION

In June 2009 the DBLP Computer Science Bibliography[1] from the University of Trier contained more than 1.2 million bibliographic records. For computer science researchers the DBLP web site is a popular tool to trace the work of colleagues and to retrieve bibliographic details when composing the lists of references for new papers. Ranking and profiling of persons, institutions, journals, or conferences is another sometimes controversial usage of DBLP.

The DBLP data may be downloaded. The bibliographic records are contained in a huge XML file[2]. We are aware of more than 400 publications which mention the usage of these data for an amazing variety of purposes. Many researchers simply need non-toy files to test and evaluate their algorithms. They are interested in XML, but not in the semantics of the data. Others inspect the DBLP data more closely: It is easy to derive several graphs like the bipartite person-publication graph, the person-journal or person-conference graphs, or the coauthor graph, which is an example of a social network. Methods for analysis and visualization of these medium sized graphs are reported in many papers. Bibliometric studies are a third group of publications. They make use of the full semantics of the data. The main disadvantages of DBLP for this purpose are the lacking citation information and the varying coverage for different subfields of computer science[3]. The main advantages are the free availability and the inclusion of many conference proceedings which play an essential role for many branches of CS and are poorly covered by other bibliographic data bases. A fourth group of papers deals with person name disambiguation, a special aspect of data quality.

DBLP is a (very imperfect) "authority file"[1] for computer science researchers. We try to identify the persons behind the research papers and to treat synonyms and homonyms as precise as possible. Incomplete and inconsistent information, imperfect software, lack of time, and our own inability are the limiting constraints for this task. The DBLP web server lists all known papers published by a person on her/his "person page". This simple mapping becomes tricky, as soon as a person has several names (synonyms) or if there are several persons with the same name (homonyms). The main obstacles are the bad habit to abbreviate (given) names beyond recognition and spelling errors. The main algorithmic idea we use to identify names we should check more precisely, is to look at person pairs which have the distance two in the coauthor graph and have a "similar" name[6].

DBLP isn't a well designed project. It grew from a small scale experimental server which was set up at the end of 1993 to test web technology. In retrospect many ad hoc solutions are poorly designed. Nevertheless, our policy is to keep DBLP as stable as possible. Data formats, URLs etc. are only changed if they prevent important functionality, and not if we simply recognized them as unaesthetic.

Section 2 of this paper describes `dblp.xml`. Beyond the syntactic framework given by the DTD, there are a lot of conventions and micro-syntax rules. We already commented on the special treatment of person names in DBLP. In section 3 you may find more details. In section 4 we sketch the remaining HTML-style part of DBLP. The new DBLP XML services are explained in the online appendix. Our example application is a simple crawler which finds the shortest path between two DBLP authors in the coauthor graph. In addition the appendix lists code to map person names to DBLP URLs.

2. DBLP RECORDS

The DBLP data set is available from the location

`http://dblp.uni-trier.de/xml/`

*An early version of this paper was titled "dblp.xml — A Documentation". Version: June 18, 2009

[1] `http://dblp.uni-trier.de`

[2] 2009/06: sizeof(dblp.xml) = 532MB, sizeof(...gz) = 93MB

VLDB '09, August 24-28, 2009, Lyon, France

The file `dblp.xml` contains all bibliographic records which make DBLP. It is accompanied by the data type definition file `dblp.dtd`. You need this auxiliary file to read the XML-file with a standard parser[4]. `dblp.xml` has a simple layout:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE dblp SYSTEM "dblp.dtd">
<dblp>
  record 1
  ...
  record n
</dblp>
```

The header line specifies ISO-8859-1 ("Latin-1") as the encoding, but in fact the file only contains characters <128, i.e. pure ASCII. All non-ASCII characters are represented by symbolic or numeric entities. The symbolic entities like `&eacute;` for the character 'é' are declared in the DTD. Numeric entities like `&#233;` should be understood by any XML parser without declaration.

In practice there are some obstacles in parsing the large XML file which cost us a lot of time: The SAX parser contained in the Java standard distribution has a limit for handling symbolic entities. When starting the Java virtual machine the option 'DentityExpansionLimit' has to be set to a large number. The parser contained in Java 1.6 distribution is not able to handle large XML files. You should install an alternative SAX Java parser. We now use Xerces-J from the Apache XML project, which reads `dblp.xml` without any problem. The DBLP FAQ[3] reports more details.

The XML root element <dblp> contains a long sequence of bibliographic records. The DTD lists several elements to be used as a bibliographic record:

```
<!ELEMENT dblp (article|inproceedings|
  proceedings|book|incollection|
  phdthesis|mastersthesis|www)*>
```

These tags correspond to the entry types used in BibTeX[5]. DBLP records may be understood as "BibTeX records in XML syntax $+\epsilon$":

```
<article key="journals/cacm/Szalay08"
     mdate="2008-11-03">
  <author>Alexander S. Szalay</author>
  <title>Jim Gray, astronomer.</title>
  <pages>58-65</pages>
  <year>2008</year>
  <volume>51</volume>
  <journal>Commun. ACM</journal>
  <number>11</number>
  <ee>http://doi.acm.org/10.1145/
      1400214.1400231</ee>
  <url>db/journals/cacm/
      cacm51.html#Szalay08</url>
</article>
```

Record Attributes

This record describes an article from CACM. The enclosing `article` element has two attributes:

- `key` is the unique key of the record. DBLP keys look like slash separated Unix file names. The most important sub-trees in the key namespace are `conf/*` for conference or workshop papers and `journals/*` for articles which are published in journals, transactions, magazines, or newsletters. The second part of a DBLP key typically designates the conference series or periodical the papers appeared in. The last part of the key may be any sequence of alphanumerical characters, in most cases these IDs are derived from the authors' names and the year of publication, sometimes a letter is appended to make this key part unique. Keys are not changed if we correct misspelled author names or year information. Obviously keys are not functional dependent from the contents of their records, you should not make any assumption about the last part of a key.

  The DBLP key namespace layout is coarse and sometimes fuzzy: periodicals or conference series often are renamed, sometimes they are split or they are joined. The world of publications does not form a hierarchy, the mapping into the DBLP namespace often is a very helpful sub-categorization, but sometimes it may be wrong or controversial. A typical problem is the decision when to treat a satellite workshop as a self-contained event or as a part of the hosting conference.

- `mdate` is the date of the last modification of this record. The format complies with ISO 8601, i.e. `YYYY-MM-DD`. We have no explicit change log for DBLP, but the mdate attribute makes it easy to load only recent additions into your application. The file `dblp.xml` contains only the current version of the records. `dblph.xml` is an extended version of the XML file which additionally contains old versions of the records. These information were extracted from the daily backups of the last years. If may be useful for the analysis of the evolution of DBLP. `dblph.xml` is only updated occasionally.

In our example the elements `author`, `title`, `pages`, `year`, `volume`, `journal`, and `number` are used similar to the corresponding BibTeX fields.

Author

In BibTeX there is at most one **author** field which may contain an 'and'-separated list of names. In DBLP records there is an `author` element for each author. The order of the `author` elements inside a record is significant, it should be the same as on the head of the paper. In BibTeX a name may be entered as `"Michael Ley"` or as `"Ley, Michael"` (last name, first name). In DBLP we always use the first form, there should be no comma inside of `author` elements. If name parts are abbreviated, each initial should always be followed by a dot, i.e. we write `H. P. Smith` and not `H P Smith` or `HP Smith`. Behind a dot, there should always be a blank or a hyphen. Names may be composed of a list of name parts. We do not make any statement about the role of the name parts. For western names, the sequence usually starts with given names and ends with family names. But even in Europe there is large diversity in local naming traditions. In many situations person names are outside of this simplistic schema:

- The name `"Luqi"` is complete, the categories first/last name simply do not apply.

- `Ingibjörg Sólrún Gísladóttir` is a traditional Icelandic name. `Gísladóttir` means daugther of Gísla,

[3] `http://dblp.uni-trier.de/db/about/simpleparser/`

there is/was no family name inherited over many generation. Because of these patronyms the Icelandic telephone directory is sorted by first names.

- In `Peter van der Stok` only Peter is the given name.
- `Yi-Ping Phoebe Chen` is the transliteration of a Chinese name. `Phoebe` is an additional western name.
- 'Jr.' is a postfix often appended to names: `Ola R. Snøve Jr.`. We separate postfixes by a space and not by a comma from the other name parts.

Section 3 describes more details of person names in DBLP. If the author of a publication is unknown, the DBLP record does not contain any `author` element.

Title

The only element which has to exist in every DBLP publication record is the **title** element. It may contain `sub` elements for subscripts, `sup` elements for superscripts, `i` elements for *italics*, and `tt` for `typewriter` text style. These elements may be nested.

Pages

Our preferred style to fill the **pages** element is "*from-to*" (unlike the `--` of BibTeX). If the number of the last page is unknown, we write *from-*. If it is a single paged paper, we just write the page number without hyphen. For split articles in magazines we occasionally use a comma separated list of page numbers or page ranges. In rare cases the `pages` element may contain any character sequence. Whenever available we prefer the conventional page numbering which is established for paper media. It is a simple addressing scheme inside of volumes and a length specification at the same time. Unfortunately many electronic publications abandon page numbering. In these cases the `pages` element may contain a paper number or the length of the paper.

Years

The **year** element should always contain a four digit number to be interpreted according to the Gregorian calendar. For journal articles, we assume that there is always a definite date of publication of the issue.

For papers published in conference proceedings the year specification may become tricky, if the proceedings are are not published in the same year as the conference was hosted. In DBLP the `year` field of conference papers specifies the date when the conference took place, the `year` field of the enclosing proceedings specifies the publication date. Our example shows the situation for a typical post-proceedings:

```
<inproceedings key="conf/naa/Xiang08" ... >
  <title>Numerical Quadrature ...</title>
  <year>2008</year>
  <booktitle>NAA</booktitle>
  <crossref>conf/naa/2008</crossref>
  ...
</inproceedings>

<proceedings key="conf/naa/2008" ... >
  <title>NAA 2008, Lozenetz, ...</title>
  <year>2009</year>
  <publisher>Springer</publisher>
  ...
</proceedings>
```

Crossref

Like in BibTeX, we use a `inproceedings` record for the paper and a `proceedings` record for the volume. The `crossref` field in the inproceedings record contains the key of the proceedings record. The conference took place in 2008, the proceedings were published in 2009. In rare cases cumulative proceedings with papers from two or more years of a conference series are published. DBLP is not able to model this situation precisely, for an example look at LNCS 5402.

For article records the `journal` field contains the name of the journal. The `volume` and the `number` field are used to specify the issue the paper appeared in. For inproceedings records the `booktitle` field gives the short name of the conference or workshop. In many cases conference acronyms are only self-explanatory for researchers who work in the subfield of computer science the conference addresses. The corresponding proceedings record should contain more detailed information about the conference and the proceedings volume. Unfortunately the proceedings records (and the `crossref` fields) are missing for a lot of legacy inproceedings records.

URL and EE

A DBLP record may contain up to two URLs in the fields `url` and `ee`. Both URLs may be either global or local. A global URL is standard internet URL, it allways starts with protocol specification of the form "*letter*$^+$:" (`http:`, `ftp:`, ...). If the `url` or `ee` contents does not start with protocol name followed by a colon, it is a local URL pointing inside the DBLP web site. To get valid URL, you simply have to add a base URL of a DBLP server as a prefix. The base URLs (DBLPbURL) of the most stable DBLP servers are

- `http://www.informatik.uni-trier.de/~ley/db/` for the origin DBLP server at the University of Trier, Germany.
- `http://dblp.uni-trier.de/db/` for our alternate server at Trier. The "author pages" of this server intentionally do not include the Complete Search facility.
- `http://www.sigmod.org/dblp/db/` for the DBLP mirror hosted by ACM SIGMOD.

In retrospect, the field names `url` and `ee` are misnomers, but there is simple "translation":

- `url` = position inside the table of contents. When we started DBLP end of 1993, nearly all publications were only available on paper. A citation of a paper, for example on an author's page, should be linked upwards to its formal context, i.e. the table of contents of the proceedings or journal volume were the paper was published in. For proceedings entries there should be a link downwards to it's table of contents. In both cases the `url` field contains the location of the table of contents. The `url` field is available for almost all DBLP records. Usually it is a "local URL".
- `ee` = position of the electronic edition. During the 1990s electronic versions of most formally published

papers became available on the servers of ACM, IEEE, and the large commercial publishers. It was obvious to extend the DBLP bibliography to a portal which makes it easy to find publications on the publishers' servers. `ee` contains the required link information. Usually the `ee` fields are "global URLs". Local URLs are only used, if DBLP has supplementary information about a paper — this facility was used for the ACM SIGMOD Anthology.

For some publishers it took several years to learn how to organize their digital libraries. Today most of them provide DOIs to address publications. Unfortunately some of the publishers do not map old URLs they declared as "stable" to the current ones. The most important cases are the IEEE Computer Society and Springer. DBLP still contains thousands of broken URLs which are the result of reorganizations by publishers, which did not take care of backward compatibility.

To store the table of contents location in (nearly) every record and to use complete URLs for the external links may seem cumbersome. Definitely it is possible to represent these information more compact by reusing the URL information of `proceedings` records if available and by using auxiliary tables with the publishers' addressing schemes. The advantage of the more redundant representation is its simplicity: DBLP records are self-contained, from each record you may produce a linked citation like on the DBLP authors' pages without any additional lookup in auxiliary tables.

(In)proceedings

Our next example shows a record describing a conference paper published in an LNCS volume and the record of the volume:

```
<inproceedings key="conf/er/Norrie08"
            mdate="2008-10-20">
  <author>Moira C. Norrie</author>
  <title>PIM Meets Web 2.0.</title>
  <pages>15-25</pages>
  <year>2008</year>
  <booktitle>ER</booktitle>
  <ee>http://dx.doi.org/10.1007/
       978-3-540-87877-3_3</ee>
  <crossref>conf/er/2008</crossref>
  <url>db/conf/er/er2008.html#Norrie08</url>
</inproceedings>

<proceedings key="conf/er/2008"
            mdate="2008-10-20">
  <editor>Qing Li</editor>
  <editor>Stefano Spaccapietra</editor>
  <editor>Eric Yu</editor>
  <editor>Antoni Oliv&eacute;</editor>
  <title>Conceptual Modeling - ER 2008,
      27th International Conference on Conceptual
      Modeling, Barcelona, Spain, October 20-24,
      2008. Proceedings</title>
  <volume>5231</volume>
  <year>2008</year>
  <isbn>978-3-540-87876-6</isbn>
  <booktitle>ER</booktitle>
  <series href="db/journals/lncs.html">Lecture
      Notes in Computer Science</series>
  <publisher>Springer</publisher>
  <url>db/conf/er/er2008.html</url>
</proceedings>
```

The `proceedings` records lists the editors of the volume. For conference proceedings "editor" is a fuzzy term. The editors listed on most LNCS proceedings are the leading organizers of the conferences: The chairs of the program committee and the general chairs. Additionally the copy editor may be listed, that is a person who did the work to form a consistent book from the manuscripts. Unfortunately the digital library of the IEEE Computer Society and the Xplore system of the IEEE umbrella organization do not list any editors. On the other hand ACM now lists the PC chairs and general chairs of most conference, but does not mention the copy editors. Editors are only the tip of the iceberg: To run a large academic conference requires hard work by hundreds of people. People who volunteer for these kind of service should be credited for their work. Several DBLP users suggested to extend our service towards this direction. Unfortunately we have not the human resources to maintain such a service on a reasonable level. If you plan any bibliometric analysis of editorship, you should be aware that these information are very incomplete and that there exists no consensus, who should be listed as the editors of a proceedings.

The `title` field of proceedings records requires same comments, too. Most LNCS volumes are ideal: After the main title, they list the exact name of the conference, the location of the event, and the exact date. Whenever possible, we try to imitate this style. Some publishers give only very incomplete information about conferences or workshops, a bad example is set by Xplore.

The `series` field specifies the book series the volume is part of. If there is a `series` field and a `volume` field, the volume number is interpreted as the numbering of the series. Sometimes two volume fields with different roles are required: LNCS Volume 5358 is the first volume of ISVC 2008 and LNCS Volume 5359 is the second proceedings volume of this conference. We refrained from inventing a second volume field. We use a simple convention: The local number is appended in braces to the `booktitle` of the `proceedings` and the `inproceedings` records, in our example "ISVC (1)" / "ISVC(2)". The optional attribute `href` in the `series` field contains the local URL of the main page of the series.

LNCS Journals

Our last example of this section looks somewhat exotic:

```
<article key="journals/jods/HurtadoPW08"
     mdate="2008-04-15">
  <author>Carlos A. Hurtado</author>
  <author>Alexandra Poulovassilis</author>
  <author>Peter T. Wood</author>
  <title>Query Relaxation in RDF.</title>
  <pages>31-61</pages>
  <year>2008</year>
  <volume>10</volume>
  <journal>J. Data Semantics</journal>
  <ee>http://dx.doi.org/10.1007/
      978-3-540-77688-8_2</ee>
  <crossref>journals/jods/2008-10</crossref>
  <url>db/journals/jods/
```

```
      jods10.html#HurtadoPW08</url>
</article>

<proceedings key="journals/jods/2008-10"
         mdate="2008-04-15">
  <editor>Stefano Spaccapietra</editor>
  <title>Journal on Data Semantics X</title>
  <booktitle>J. Data Semantics</booktitle>
  <series href="db/journals/lncs.html">Lecture
     Notes in Computer Science</series>
  <volume>4900</volume>
  <publisher>Springer</publisher>
  <year>2008</year>
  <isbn>978-3-540-77687-1</isbn>
  <url>db/journals/jods/jods10.html</url>
</proceedings>
```

Springer publishes several journals inside of the LNCS series. Each volume of these journals is an LNCS volume. Rightly authors of papers from these journals asked us to classify their papers as articles. On the other hand the volumes of these journals are more self-contained than usual volumes of a journal: They have an ISBN and a series volume number. The editors change from volume to volume. Our ad hoc solution is to describe the volumes by `proceedings` records or `book` records, but this is incorrect and shows the limitations of classic BibTeX records to model the world of scientific publications.

More Record Types

A future DBLP version should have new record types for journal volumes. In addition it makes sense to make journals as a whole "first class citizens", i.e. to model them by explicit by records of a new type. Information like the ISSN, the publisher of the journal, and the journal home page can be stored in such records. The same holds for conference series like "the annual VLDB conference", or book series like LNCS. The modeling will become complicated because these objects do not form a hierarchy: A proceedings volume of a joint conference is a member of several conference series. The SIGMOD 2000 proceedings volume is a member of the SIGMOD conference stream and it is a number (not a volume) of SIGMOD Record volume 29.

We are aware of many shortcomings of the current data model, but nevertheless a fundamental revision has low priority. We fear that a "perfect" data model becomes very complex and makes data acquisition too expensive. In DBLP we use plain hypertext pages to describe situations not covered by our data model. We discuss this "escape mechanism" in section 4.

To collect citation links is another feature which is very attractive, but which was abandoned because it is too costly to be done manually. The `cite` fields in the DBLP records were entered for a small subset of the ACM SIGMOD Anthology papers, the contents of these fields are DBLP keys of cited papers. For large scale citation link information you should look at CiteSeer and the ACM Guide to Computing Literature.

3. PERSONS

Humans are social beings. Language requires to name persons you want to speak about. As long as life was organized in small groups and there was little exchange with other groups, ad hoc naming of persons works perfectly. It was sufficient to name a person by her/his function ("Smith") or by the parents ("Gísladóttir"). A large variety of naming conventions developed in different cultures[2].

DBLP is a global person register for researchers of computer science and neighboring sciences. Our collection now contains about 700000 different person names. On this scale, which is at most "medium" compared to the set of all living humans, the traditional naming conventions reach their limits. Synonyms and homonyms become a main problem.

Author Pages

When we started DBLP in 1993 with the papers of a few hundred persons from the data base systems community, we did not take care of any scaling problems or the complexity of naming conventions. Each person should have her/his own "**DBLP author page**". The papers by David Maier are listed on

DBLPbURL`indices/a-tree/m/Maier:David.html`

and the papers by Laura M. Haas on

DBLPbURL`indices/a-tree/h/Haas:Laura_M=.html`

(*DBLPbURL* = `http://www.informatik.../db/`, a DBLP base URL — see section 2). All pages for persons whose last name starts with an 'm' are stored in the directory `.../indices/a-tree/m/`, and all 'h'-names are stored in `.../indices/a-tree/h/`, etc. The filenames inside these directories are formed

last-name:*first-name*`.html`

Blanks are mapped to underscores, all other characters which are not alphanumeric are mapped to '='. This avoids illegal URLs.

Today this primitive mapping produces huge directories. For example there are now more than 700000 persons with last names starting with 's'. A few years ago this was a performance problem. Today there is no good argument to change the URLs of thousands of web pages which are referred by numerous other web servers and search engines. We regard URL stability as a very important virtue to make a service reputable. There are (at least) two solutions for implementation:

- We continue to materialize DBLP author files as static HTML files. The files are generated daily. Fortunately a lot of data base system technology has been moved into standard file systems. Contemporary file systems have no performance problem to handle huge directories, because they use variants of B-trees or hash tables (ext2/3) as access paths. For the file system with the author pages, we switched off the logging into a journal — if the system crashes, we produce new author pages from `dblp.xml`. The journaling was the main bottleneck when we refreshed all author pages.

- Today you can choose from a large palette of technologies to implement dynamic web pages. Even the oldest of these technologies, the CGI interface between web servers and programs written in nearly any language, allows realization of any legal URLs by software. For example the DBLP BibTeX pages look like static HTML files, but they are produced on demand

by a little (and still imperfect) C program from the XML records.

In retrospect it was a design error to split the person names into first names and last names. A correct name splitting is not computable with feasible costs. To make our algorithm reproducible, we nevertheless document the source code in the appendix. The problem becomes evident, if you apply it to a Spanish name like "Juan Antonio Holgado Terriza": The algorithm says that Terriza is the last name, but in fact Holgado Terriza is the correct answer. Certainly it is possible to codify a lot of knowledge about naming into an algorithm, but in practice any algorithm remains incomplete because our knowledge is incomplete. To tag name parts by more detailed markup fails for the same reason. For some users of DBLP the incorrect splitting of their names is annoying, but we do not claim any longer that the name parts are first/last names.

Homonyms and Synonyms

David and Maier are popular names, but at the moment there seems to be only one person who publishes computer science papers — the well-known professor at Portland, Oregon. Other names are less unique. In DBLP you may now find papers by at least seven different persons with the name Chen Li. In these cases, we have to add some small "mystical" number to make the homonym persons distinguishable:

<author>Chen Li</author>
<author>Chen Li 0002</author>
<author>Chen Li 0007</author>

We optionally append a space character and a four digit number to names. For the naïve name split algorithms the number is a postfix to the last name. The number is part of the URL, but not printed with the name:

`.../indices/a-tree/l/Li_0007:Chen.html`

The hard problem is to detect homonym persons and to assign the correct ID numbers. We have no solution for this problem, but simple heuristics proved to be very helpful: The coauthor index at the bottom of the DBLP authors' pages is colored. If persons have jointly published they get the same color. If there is no direct connection between two persons or transitively via another member of the coauthor list they are assigned different colors. If a coauthor list is monochrome, we are quite confident that the main name entry represents a single person. There are two main reasons for multicolored coauthor lists: If DBLP only contains a small sample of the publications of a person, our information is simply too incomplete — this often happens to senior researchers working in an area only partially covered by DBLP. The other reason is a "split personality". If the coauthors can be separated in disjunct groups, the main entry may represent several persons or a person who works in several distant research areas with unconnected colleagues. The reason for this may be the change of the affiliation. An open challenge is to develop a clustering algorithms which indicates homonyms with a better precision. This algorithms should not only look at the structure of the coauthor graph, but also to the conferences, journals, title key words, publication years etc.

At the moment the splitting of DBLP author pages is either triggered by requests of authors who find their publications mixed with other persons writings or if we can prove our own strong suspicion that there are several persons behind an entry. In many cases homonyms remain undetected.

The problem is not new. The library community works on it for a long time. An important project is VIAF, the Virtual International Authority File, their web interface is available on `http://viaf.org/`. If you enter my name, you get a list of some of my old papers mixed with publications of several unrelated homonyms. I share the year of birth (1959) with at least one of these homonomys.

After an entry has been split and hidden numbers have been added to the `author` and `editor` fields in the bibliographic records we may miss to assign the correct name ID to new entries. Again we have no perfect solution, but a heuristics which works in many cases: There is a daily job which looks for homonym persons with different ID numbers and a shared coauthor. Until now there were no homonyms close enough to share a coauthor. All alerts produced by our program were caused by input errors, we are still waiting for the first "false hit".

The job to alert for likely incorrect homonym IDs is a special case of a more general framework[6]: In a first step we select a promising subset of the huge product space of the $> 700000^2$ DBLP person pairs. It is often senseless (e.g. for the homonyms) or too expensive to consider all person-pairs. A "blocking function" (this notion was introduced by the census community) selects a subset of the product space. Important blockings are (1) all person pairs with the distance two in the coauthor graph, (2) all person pairs who have published in the same conference series or journal, (3) all person pairs who have published papers which contain the same rare title (key-)word. The person-pair-streams provided by the blockings may be filtered. For example we may be interested only in persons with overlapping publication years. Other filters look at the `mdate` attribute or the coauthor colorings. The most important filter applies a string distance to the names of the two persons. Typical string distances are the classical edit-distance, diacritic/case insensitive comparisons (`René` $\sim$ `Rene`), comparators insensitive to permutations of name parts (`Li Chen` $\sim$ `Chen Li`), comparators which are able to expand initials (`M. Ley` $\sim$ `Michael Ley`), or the comparator mentioned above which ignores the hidden ID (`Chen Li 0003` $\sim$ `Chen Li`). We use this software to find **likely synonyms**, e.g. if there are entries `Michael J. Carey` and `M. Carey` in the same journal, we should check if they belong to the same person. It is easy to produce long lists of candidates, but fine tuning to get a good precision remains a open problem. We implemented more than 20 distance functions for person names. But there remain many cases which are hard to capture by general rules and require a specialised knowledge base. For an example of an "hard case" look at

A. Kourtis = Anastasios Kourtis
= Tasos Kourtis = T. Kourtis

The software sketched above works retrospective, it helps to find errors in the data base. In addition we experiment with software which tries to avoid input errors. The basic idea is to consider the input of a new multi-authored publication as a graph matching problem. We have to find a good position for the small graph (the new entry) in the huge graph (DBLP). First we look for exact matches of person names, then we use distance functions insensitive to diacritics and initials. If we find some of the names to be entered, we do a local search in the neighborhood of the hits with

more expensive distance functions. Again the fine tuning of this algorithm remains the challenge. Often the human user "sees" partial hits our algorithms are not able to locate.

Person Records

Many DBLP authors maintain their own personal "home pages" on the web. It soon became obvious to add links from the DBLP author pages to personal home pages. Home pages were modeled as special web publications:

```
<www key="homepages/m/DavidMaier" ...>
  <author>David Maier</author>
  <title>Home Page</title>
  <url>http://web.cecs.pdx.edu/~maier/</url>
</www>
```

The key always starts with `homepages/`, the `author` field specifies the name of the person, the `title` field always has the value "Home Page", and the `url` field contains the location of the home page. Later it became clear that this modeling was shortsighted: We should be able to store more information about a person, we need "person records". To enable a smooth upgrading of running software, we extended the "home page records" to "person records". The most important addition are secondary names for persons:

```
<www key="homepages/h/AlonYHalevy" ...>
  <author>Alon Y. Halevy</author>
  <author>Alon Y. Levy</author>
  <title>Home Page</title>
  <url>http://alonhalevy.googlepages.com/</url>
</www>
```

Persons may change their names for several reasons, marriage is the most important one. If a person records contains more than one `author` field, the additional names are interpreted as a list of **secondary names for the same person**. On the DBLP web server we simply produce redirections from the secondary name URLs to the primary name URL. The primary name person page lists the publications of any spelling variant of the person. Secondary names are not only useful to model name changes, but they also enable us to treat with synonyms which are used at the same time:

```
<www key="homepages/r/CJvanRijsbergen" ...>
  <author>C. J. van Rijsbergen</author>
  <author>Cornelis Joost van Rijsbergen</author>
  <author>Keith van Rijsbergen</author>
  <title>Home Page</title>
  <url>http://www.dcs.gla.ac.uk/~keith/</url>
</www>
```

This famous IR pioneer is known under three name variants.

To identify people, it may be helpful to store additional information like their affiliation or their name in an alternative writing system. In person records, there is an optional `note` field. The contents of this field is printed out at the heading of the corresponding person page. Currently the note field of person records is the only place where DBLP extends above the Latin-1 character set.

```
<www key="homepages/m/AtsuyukiMorishima" ...>
  <author>Atsuyuki Morishima</author>
  <title>Home Page</title>
  ...
  <note>&#x68EE;&#x5D8B;&#x539A;...</note>
</www>
```

The last extension of person records for now is still populated only by a few instances. `cite` fields inside of person records are interpreted as biographical citations in a broader sense, we intend to associate Festschrift publications, obituaries, etc. with the honored person.

```
<www key="homepages/k/ParisCKanellakis" ...>
  <author>Paris C. Kanellakis</author>
  <title>Home Page</title>
  <url>...</url>
  <note>Dec. 3, 1953 - Dec. 20, 1995</note>
  <cite>conf/birthday/2003pkc</cite>
  <cite>journals/csur/AbiteboulKMSV96</cite>
  <cite>conf/pods/AbiteboulKPV96</cite>
  ...
</www>
```

Person IDs

DBLP is an (very imperfect) "authority file" for computer scientists. Conference servers, conference management systems, preprint servers, publishers, and universities refer to DBLP. Several of these members of the "publication chain" asked us to provide a stable mechanism to point to persons in DBLP. To keep the URLs of author pages stable was a first important step. But this policy is limited by the dependence of the URL from the exact spelling of the person name. For most persons the spelling converges to a stable and hopefully correct state after a while. But sometimes our information about a name remains incomplete for a long period. Nevertheless it is necessary to provide stable person IDs to enable other applications to exchange information with DBLP.

A global person ID does not exist and is very controversial. Even for scientists it is not well established yet. Until there is a more general mechanism, we should introduce or own DBLP person IDs. Again we do this by extending an existing mechanism: We simply use the IDs of person records as person IDs. The IDs of the existing person records remain unchanged. For any person in DBLP who has no person record we generate a new record. The new IDs are consecutively assigned integers. A generated person record only contains a mapping from an ID to a name:

```
<www key="homepages/45/123" ...>
  <author>C. Ley</author>
  <title>Home Page</title>
</www>
```

We may change "C. Ley" to "Carola Ley" as soon as we get additional information about the name. The person ID remains stable, the URL of the author page changes.

Two situations require further explanation: Author pages may be joined or splitted.

If there is already an author page and person ID for Carola Ley, we may have assigned two or more IDs to the same person. We should not invalidate an redundant ID, but register that it is equivalent to another one:

```
<www key="homepages/45/123" ...>
 <crossref>homepages/55/1002</crossref>
</www>
```

The hard case is splitting of author pages. On the page of "C. Ley" we may have collected publications by "Carola Ley" and "Christoph Ley". The ID of "C. Ley" should become invalid because it is not a single person, but a set of similar named persons. In practice this strict interpretation of splitting may have absurd side effects: Assume "C.

Ley" (alias Carola) has published papers listed in DBLP for several years with her initial only. Now the first paper of Christoph is added to the author page of "C. Ley". In this asymmetric sitation it makes sense to keep the ID for Carola stable and to assign a new ID to Christoph, even if his publications temporarily were merged with Carola's publications by mistake. There remains a gray zone in the decision when a splitting is asymmetric or symmetric.

To make Person Ids operational we provide a simple mapping service. The URL

`http://dblp.uni-trier.de/rec/pid/`*pID*

redirects to the author page with the specified *pID* (person record ID without "homepages/" prefix). For example try

`http://dblp.uni-trier.de/rec/pid/00/7`

4. ESCAPE TO HYPERTEXT

DBLP was started in 1993 as a small collection of HTML files which were directly entered using a standard text editor. Very soon the bibliographic records were extracted from the HTML files. For the records we used a format very close to XML, later it was adjusted to conform with the standard. We did not envision the details of XML in 1994, but it was a natural choice for lazy programmers. Our first parser was trivially derived from the parser of the xmosaic browser, to use a small customized markup language was very simple to implement.

Not all information from the origin HTML tables of contents (TOCs) went into the bibliographic records. `dblp.xml` is sufficient to generate the DBLP person pages, but not the TOC pages and the navigation pages. The most notably information missing for the TOCs are session titles. The source for a typical TOC page now looks like this:

```
...
<cite key="conf/vldb/2006">
<h2>Keynote Addresses</h2>
<ul>
<li><cite key="conf/vldb/Jhingran06" style=ee>
<li><cite key="conf/vldb/Sikka06" style=ee>
</ul>
<h2>Ten-Year Best Paper Award Talk Session</h2>
<ul>
<li><cite key="conf/vldb/HalevyRO06" style=ee>
</ul>
<h2>Research Sessions</h2>
<h3>Continuous Query Processing</h3>
<ul>
<li><cite key="conf/vldb/LiCTACH06" style=ee>
...
<footer>
```

This is HTML with a few additional customized elements: `cite` includes the bibliographic record specified by the `key` attribute. The optional attribute `style` is used the choose from several formatting options. `footer` produces the standard DBLP footer, `ref` is used for hyperlinks inside of the DBLP web pages, etc. We named this slightly extended HTML "bibliography hypertext" (BHT).

In 1994 we implemented a primitive program which produces HTML from these BHT-files. The program imitated the idea of the C preprocessor, later "Server Side Includes" and several more advanced mechanisms to compose HTML from scattered building blocks were introduced. Contrary to "server side includes" the HTML pages are not produced on demand but they are composed daily in advance. We use this archaic mechanism until today because it gives us a lot of flexibility to circumvent the limited modeling power of the bibliographic records. In essence

DBLP = bibliographic records + BHT-files.

Even it we often get to limits of the bibliographic records, we are very hesitant to extend the model because we try to keep it simple and manageable. The hypertext part of DBLP gives us the freedom to describe any irregularity in the publication world without inventing a mechanism for a singular case. In a few cases we extended the power of the bibliographic records because a not anticipated phenomenon started to occur more frequently. Homonyms/synonyms and their representation in person records are the most important examples of a DBLP-feature which was moved from hypertext to records.

For the TOC pages the BHT files often are not more than a skeleton with the enumeration of the papers which constitute the volume. In many cases they are upgraded by session titles. Sometimes we have added editorial comments. TOC BHT files for proceedings may be viewed as appendices to the corresponding `proceedings` records. The `url` field is the connecting link. On the level above, for journals or conference "streams", the BHT files often become much more irregular. Here the escape mechanism to hypertext may be more justified.

The original BHT files are still not consistent with the XML syntax. In `.../xml/dblp_bht.xml` you may find a version of them which is "XMLlized" by a script.

5. REFERENCES

[1] Authority control. *Wikipedia*, 2009.

[2] Personal name. *Wikipedia*, 2009.

[3] A. H. F. Laender, C. J. P. de Lucena, J. C. Maldonado, E. de Souza e Silva, and N. Ziviani. Assessing the research and education quality of the top Brazilian Computer Science graduate programs. *SIGCSE Bulletin*, 40(2):135–145, 2008.

[4] T. C. Lam, J. J. Ding, and J.-C. Liu. XML document parsing: Operational and performance characteristics. *IEEE Computer*, 41(9):30–37, September 2008.

[5] L. Lamport. *LaTeX User's Guide and Document Reference Manual*. Addison-Wesley, 1986.

[6] M. Ley and P. Reuther. Maintaining an online bibliographical database: The problem of data quality. In *EGC'2006, Actes des sixièmes journées Extraction et Gestion des Connaissances, Lille, France, 17-20 janvier 2006, 2 Volumes*, pages 5–10, 2006.

APPENDIX

If your software needs only in a few facts from DBLP, downloading the entire `dblp.xml` file may be a too costly burden. The web pages are intended for humans, wrappers are always exposed to the risk of formatting changes. In the online appendix we describe a very basic API for DBLP. It is available from

`http://dblp.uni-trier.de/xml/docu/dblpxmlreq.pdf`