3

# Maintaining temporal integrity of World Wide Web pages

*G.F. Knolmayer, T. Buchberger*
*Institute of Information Systems, University of Bern*
*Engehaldenstrasse 8, CH 3012 Bern, Switzerland*
*Phone: +41.31.631.3809, Fax: +41.31.631.4682*
*URL: http://www.ie.iwi.unibe.ch/*
*E-Mail: {knolmayer|buchberg}@ie.iwi.unibe.ch*

**Abstract**

A vast amount of temporal information is provided on the World Wide Web (WWW). It is extremely difficult for a webmaster to maintain this information without inconsistencies. The business and competitive aspects of maintaining temporal integrity are discussed. We present a concept for supporting the maintenance of WWW pages by a Java agent which tries to identify temporal information.

**Keywords**

World Wide Web, Temporal Integrity, Competitive Effects, Maintenance Tool, Java Applet.

## 1 DEVELOPING AND MAINTAINING WEB PAGES

In late April 1997, the WWW search machine AltaVista (Digital Equipment Corporation, 1997) identified and indexed more than 30 million web pages. On average, approximately 50 pages can be found on *one* web server. For big organizations it is extremely difficult to keep the information on the large number of its WWW pages consistent. Therefore, there is a need to support the development and maintenance of web-sites by methods and tools which may have similar properties as the CASE tools recommended for developing and maintaining conventional information systems (IS) (Fisher, 1988; Holloway/Bidgood, 1991).

Integrity and Internal Control in Information Systems Volume 1
S. Jajodia, W. List, G. McGregor & L. Strous (Eds) © 1997 IFIP. Published by Chapman & Hall

Although some approaches have been suggested resp. described (Fielding, 1994; Aimar et al., 1995; Isakowitz/Stohr/Balasubraman, 1995; Wreggit, 1995; Bichler/Nusser, 1996; Graham, 1996; Millmann, 1997), the structural and navigational design of WWW applications is still far more an art than an engineering task; there are no widely accepted methodologies and tools available that support the maintenance of web pages. System development without applying engineering methodologies has resulted in legacy systems which are very hard to maintain and in which seemingly trivial requirements like making IS Year 2000 compliant result in nightmares. With regard to WWW presentation, history seems to repeat itself and systematic engineering methods are not widely disseminated in web design.

A main component of the web's success are the easy connections via links to other sites that provide somehow related information. Usually, the linked site does not know or not care that another site has established a link to it and feels free to autonomously restructure its presentation which may cover

- deleting previously existing addresses and
- changing the contents of pages.

These changes may result in inconsistencies.

In this paper we do not deal with the integrity of WWW information in general but with its *temporal* integrity. We show that much temporal information exists as well in IS as on the web. We discuss temporal integrity issues for data on the web and compare them to those of structured data. Finally, a Java applet is described which extracts temporal information by scanning web data.

## 2 TEMPORAL DATA ON THE WEB

### 2.1 The existence of temporal data

Many conventional IS make references to temporal data. On the basis of several case studies it is assumed that about 80 % of all programs resp. 60 % of all data stores process resp. contain temporal data (Rubin, 1996). There is a huge body of scientific work related to support the storage, integrity and query of temporal data in database management systems (DBMS). One of the extensions of SQL3 discussed by ANSI and ISO is SQL/Temporal (Snodgrass/Böhlen/Jensen/Steiner, 1996a; Snodgrass/Böhlen/Jensen/Steiner, 1996b) which should provide support for temporal data handling in relational database systems. Tools to support temporal data in data warehouses have been announced (Bair, 1996).

Temporal data on the web has not found much interest thus far. In April 1997, AltaVista found more than 9 million references to the string '1997' and more than 25 million references to '1996' (cf. Figure 1 for further details). Thus, temporal data exist in abundance on the WWW.

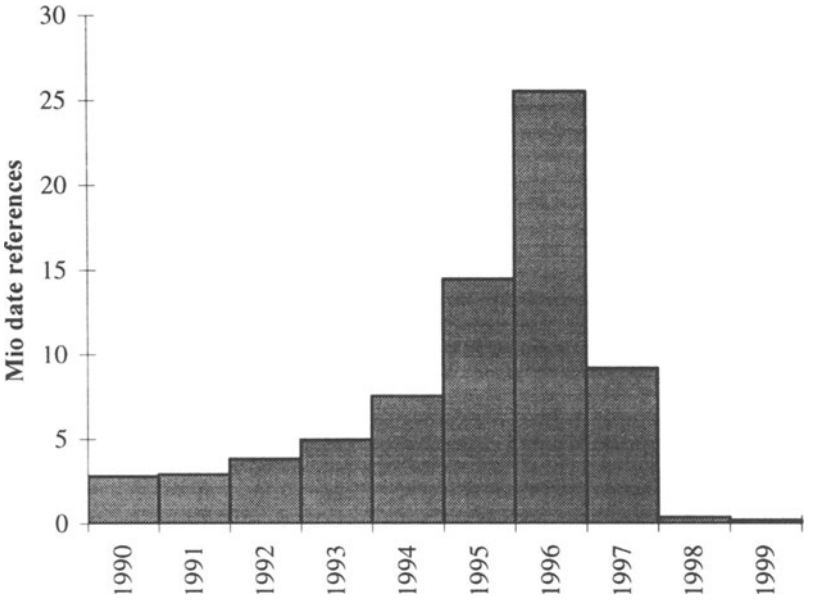


**Figure 1** Number of date references on the WWW (as of 1997-04-16)

When surfing the web, one passes many sites making references to past events or states while speaking in future tense. This means that the persons who are responsible for the WWW pages have not updated their contents according to the everlasting flow of time. For ease of formulation, in the following these persons are called webmasters although in some cases individual users are in charge for maintaining WWW pages (e.g. private home pages).

## 2.2 The importance of temporally consistent data

One main aspect of providing non-contradictory information on web pages is to update information when an event occurred which has been previously announced on the WWW. Examples of such events are conferences, meetings or courses given at certain days or time periods. In most cases, the information provider describes a forthcoming event. An event is often characterized as a fact occurring at an instant (Jensen et al., 1993). The problems associated with „interval events" and with recurrent events will be neglected in this paper.

Several reactions may be appropriate after the occurrence of an event which has been announced on the web. If the only goal of the WWW presentation was to attract an audience for the event and there will be no further references to this event, the information resp. page may be simply canceled. This may result in

dangling links from other pages. When a certain conference is finished, it is not necessary e.g. to provide the information about lodging facilities any longer. However, somebody may plan a trip to the *conference place* independently from the conference and may be pleased to find some neutral information about appropriate facilities even if this information may not be up-to-date.

Often information about bygone events is intentionally presented on the web. Proceedings of a past conference may be available and the web-surfer should be informed about the authors and the papers that were presented and included in the proceedings. It may also be interesting to compare how the calls for papers for a certain (e.g. annual) conference changed over time.

The reader will usually regard an information provider as lazy or too busy or unobservant if a sentence written in future tense refers to a time point which is already in the past. The degree of temporal integrity achieved may be defined as the number of pages without temporal inconsistencies divided by the total number of pages provided by an organization. The web-surfer may assume that the degree of temporal integrity correlates positively with the seriousness of the information provided or even with the seriousness of the information provider as a business partner. Thus, inadequate maintenance of web pages may have consequences for the competitive position of an enterprise (Kahn, 1996).

If someone would base his portfolio decisions on the stock market data of 1995-09-22 which is delivered on a CD-ROM in 1996 (Knolmayer/Leuenberger 1997), this could obviously result in inappropriate decisions. Daily comments on stock markets, as provided e.g. by the teletext service of the TV station European Business Network, do not always clearly define the day which is referenced. The WWW presentation is more precise, stating e.g. „Page last updated August 22, 5:46:20 PM CET" at
http://www.ebn.co.uk/Markets/MarketReports/Zurich/
However, in the terminology of temporal database theory, this is the registration time which may differ from the valid time (Jensen et al., 1993), e.g. if system problems did not allow a well-timed update.

Timeworn information may result in a huff if the user is confronted e.g. with a time-constrained sales promotion which is not any longer valid. He may start a bargaining process to obtain the attractive offer although it is formally gone. Outdated information may be especially annoying if it is part of a logical temporal sequence and the more recent information is not accessible. One example are university departments showing information about courses given in past terms but not about courses to be given in the next term (Schmitz, 1997).

In markets with high technological progress the data books are often outmoded. Thus, the user may wonder whether the printed catalog or the WWW data have to be applied. To avoid these problems, the webmaster may put on dummy changes to produce a recent time-stamp, thus confirming that the data are maintained and very likely not outdated. Also a fictitious end of valid-time may be placed on the page, providing the user with confidence about the validity of the data. The next

maintenance of the page has to occur before the end of valid-time and may simply increment it to a later time-point.

On the web, the user is usually suffering from information overload. This overload is partially due to the fact that outdated and thus irrelevant information is still on the web. Search machines that estimate the relevance of a document could also consider the day of its last update and assign non-maintained pages a low presentation priority.

To avoid the negligence connected with temporal inconsistencies it seems worthwhile to develop tools for identification of those parts of WWW pages that contain temporal information. Thus we are interested in building a temporal agent which supports the webmaster in assuring temporal integrity.

# 3 TEMPORAL INTEGRITY

## 3.1 Integrity in database systems

The transaction concept of DBMS is very stringent to secure the integrity of data. On the other hand, deferred updates have been widely accepted e.g. in decision-making applications and recent work considers the relaxation of the requirements for DBMS transactions in distributed systems (Alonso/Hagen/Schek/Trech, 1997). The price to be paid for distributed systems are somehow relaxed, probably application-dependent integrity requirements.

Integrity aspects of structured temporal data are discussed e.g. in (Gertz/Lipeck, 1995; Myrach/Knolmayer/Barnert, 1996; Knolmayer/Myrach, 1997). In the DBMS-context, temporal key integrity holds if the associated time-stamps of equal-valued (time invariant) keys do not overlap. Guaranteeing temporal referential integrity means that a foreign key value must not only match the value of the referred snapshot primary key but also the time-stamp associated to the foreign key must be contained in the associated time-stamp of that primary key.

## 3.2 Integrity in un- or semi-structured documents

Compared to the rather easily defined integrity constraints for structured data, the goal of presenting non-contradicting information in un- or semi-structured data is far more demanding. Checking the integrity of unstructured texts requires the formalization of statements given in natural language. Although there is much work on temporal reasoning (Montanari/Pernici, 1993), the potential of using these results for maintaining WWW pages seems to be rather small.

Temporal information on the WWW can be regarded as robust and we assume that there is an interval in which the web-surfers are not concerned about a missing update of temporal data. Thus, we base our integrity checking on the following time points:

Day of **i**nserting a web page or of last update of a web page: *i*

This information may be provided by the webmaster or by the WWW server.

Day of the **e**vent: *e*

This information is hopefully found by the scanning mechanism.

**R**obustness interval in days: *r*

The interval length has to be defined by the webmaster, based on the type of information presented, on the frequency of page maintenance, and on the „netiquette".

Real **t**ime: *t*

The present day in the time zone in which the webmaster acts.

Applying these definitions, the following temporal business rules can be defined:

*If (((i-e)<0) AND ((t-e)>0)) AND page not intentionally frozen*
*Then Mark Node for Maintenance*

*If (((i-e)<0) AND ((t-e-r)>0)) AND page not intentionally frozen*
*Then Trigger Delete (parts of) the WWW page OR*
*Alert Webmaster forUrgent Maintenance*

In formulating these rules we assume that pages with ((i-e)<0) are written in future tense. A standard for signifying that a certain page has been frozen intentionally, e.g. for documentation purposes, would be helpful.

## 3.3 Examples of temporal inconsistencies

Examples in which the webmasters did not care about temporal integrity are very common. Searching for

"will take place" NEAR (1990 or 1991 or 1992 or 1993 or 1994 or 1995 or 1996)

results in approx. 6000 hits by Advanced AltaVista (as of 1997-05-13). Searching for

Deadline NEAR (1990 or 1991 or 1992 or 1993 or 1994 or 1995 or 1996)

results in approx. 20 000 hits. We quote an example from the server of Temple University, Philadelphia (http://joda.cis.temple.edu/~friedman/ACM50thTop.html):

"The ACM 50th Anniversary Celebration consists of a series of events and special programs beginning in Philadelphia, PA, in February, 1996 (the 50th anniversary of the ENIAC) and culminating with the 50th Anniversary of ACM in San Jose, CA, March, 1997. The February, 1996, "Launch Week" events will take place in conjunction with ACM Computing Week '96 and will focus on the historical perspective of what has happened to date in computing; the 1997 segment at ACM Computing Week '97 will focus on the future of computing and its impact on the world community." (as of 1997-05-13)

Making reference to the organizing institution, we give some examples of outdated information about IFIP events. The server of IFIP WG 11.8 at
http://www.ifip.tu-graz.ac.at/TC11/WGS/wg-11.8.html mentions
"A regular WG 11.8 meeting will take place some time in February-March 1996 in Europe." (as of 1997-05-13) and IFIP WG 11.3 announces at
http://www.itd.nrl.navy.mil/ITD/5540/ieee/cipher/cfps/cfp-wg11395.html :

"CALL FOR PAPERS. The Ninth Annual IFIP WG 11.3 Working Conference on Database Security.
Rensselaerville, New York, U.S.A. August 13 - 16, 1995.
Deadline for Submission: March 20, 1995" (as of 1997-05-13).

# 4 SCANNING SINGLE WEB PAGES FOR TEMPORAL DATA

## 4.1 Search strings

A first step in checking temporal integrity is to scan a certain WWW page which has to be defined as input for the scanning program. No links to other WWW pages are considered in this approach.

First we assume that there exists a time stamp *i* for the last change of a WWW page. The time stamp may be provided by the webmaster and be visible on the WWW page. Many web servers automatically transmit the day of the last update of the HTML-file to the client systems. Some servers (e.g. http://www.unibe.ch) do not provide this service; this behavior may depend on the server software or its configuration. The AltaVista search engine gives the date of the last update of the WWW page in

d mmm yy (e.g. 4 Dec 97)

notation (as explained below). Although it is not clear from the documents available, we assume that AltaVista provides the date of adding a WWW page to its index if the web server does not deliver the day of the last update. In the following we assume that the server provides the date of the last update. In this case the maintenance support tool may read the http-field „Last Modified".

The representation of temporal data differs in distinct cultures, even if the same (e.g., Gregorian) calendar is used as reference; date representations differ on WWW pages formulated e.g. in USA or Europe. Additionally, time-related expressions used in different languages may be of importance. A formalized date representation consists of basic elements connected by some separators (which also include a blank). In the context regarded we are interested in the user interface; thus, special internal date representations, e.g. Lilian dates or packed time data (IBM, 1997) are not relevant. In Tables 1, 2, and 3 we use the notation in which Microsoft explains the date formats available in the English version of Excel 8.0:

| „To display | Use this format code |
|---|---|
| Months as 1-12 | m |
| Months as 01-12 | mm |
| Months as Jan-Dec | mmm |
| Months as January-December | mmmm |
| Months as the first letter of the month | mmmmm |
| Days as 1-31 | d |
| Days as 01-31 | dd |
| Days as Sun-Sat | ddd |
| Days as Sunday-Saturday | dddd |
| Years as 00-99 | yy |
| Years as 1900-9999 | yyyy" (Microsoft, 1997). |

In Table 1 we assemble the most important date representations that may be derived from these basic elements. We ignore the ambiguous mmmmm and ddd and dddd representations, remaining with 4 basic elements for representing the month, 2 elements for depicting the day and 2 elements for representing the year. Thus, the 4*2*2=16 basic combinations shown in Table 1 have to be considered scanning English pages; the basic elements mmm and mmmm are language-dependent. We show different representations for the day 1997-07-04 (in ISO 8601 notation) which is well suited to show the differences between d and dd resp. m and mm. We do not present all possible combinations between separators and the „th" appendix and omit the representation of years as 'yy. Some pages abbreviate the names of the months with more than 3 characters. On
http://moon.inf.uji.es/Ian/HTMLdocs/PCTOOLS/pc_editors.htm (as of 1997-08-23) 5, 4, and 3-digit characterizations of months are intermixed (e.g., April, Sept, Dec).

Each of the resulting combinations may be formatted e.g. using different separators and distinct sequences of the basic elements. Predominant sequences in Europe are shown in Table 2. Unfortunately, the ISO 8601 standard for time representation published in 1988 (Table 3) is not widely accepted. Our prototype *INT²IME* (cf. chapter 6) identifies all the entries shown in the Tables 1, 2, and 3.

| | | | | | | |
|---|---|---|---|---|---|---|
| mdyy | 7-4-97 | 7/4/97 | 7 4 97 | | | |
| mdyyyy | 7-4-1997 | 7/4/1997 | 7 4 1997 | | | |
| mddyy | 7-04-97 | 7/04/97 | 7 04 97 | | | |
| mddyyyy | 7-04-1997 | 7/04/1997 | 7 04 1997 | | | |
| mmdyy | 07-4-97 | 07/4/97 | 07 4 97 | | | |
| mmdyyyy | 07-4-1997 | 07/4/1997 | 07 4 1997 | | | |
| mmddyy | 07-04-97 | 07/04/97 | 07 04 97 | | | |
| mmddyyyy | 07-04-1997 | 07/04/1997 | 07 04 1997 | | | |
| mmmdyy | Jul-4-97 | Jul/4/97 | Jul 4 97 | Jul 4, 97 | Jul. 4 97 | Jul 4th 97 |
| mmmdyyyy | Jul-4-1997 | Jul/4/1997 | Jul 4 1997 | Jul 4, 1997 | Jul. 4 1997 | Jul 4th 1997 |
| mmmddyy | Jul-04-97 | Jul/04/97 | Jul 04 97 | Jul 04, 97 | Jul. 04 97 | Jul 04th, 97 |
| mmmddyyyy | Jul-04-1997 | Jul/04/1997 | Jul 04 1997 | Jul 04, 1997 | Jul. 04 1997 | Jul 04th, 1997 |
| mmmmdyy | July-4-97 | July/4/97 | July 4 97 | July 4, 97 | | July 4th 97 |
| mmmmdyyyy | July-4-1997 | July/4/1997 | July 4 1997 | July 4, 1997 | | July 4th 1997 |
| mmmmddyy | July-04-97 | July/04/97 | July 04 97 | July 04, 97 | | July 04th, 97 |
| mmmmddyyyy | July-04-1997 | July/04/1997 | July 04 1997 | July 04, 1997 | | July 04th, 1997 |

**Table 1** Month - Day - Year (US-Notation).

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| dmyy | 4-7-97 | 4/7/97 | 4 7 97 | 4.7.97 | | | |
| dmyyyy | 4-7-1997 | 4/7/1997 | 4 7 1997 | 4.7.1997 | | | |
| dmmyy | 4-07-97 | 4/07/97 | 4 07 97 | 4.07.97 | | | |
| dmmyyyy | 4-07-1997 | 4/07/1997 | 4 07 1997 | 4.07.1997 | | | |
| dmmmyy | 4-Jul-97 | 4/Jul/97 | 4 Jul 97 | 4. Jul 97 | 4 Jul. 97 | 4. Jul. 97 | 4th Jul 97 |
| dmmmyyyy | 4-Jul-1997 | 4/Jul/1997 | 4 Jul 1997 | 4. Jul 1997 | 4 Jul. 1997 | 4. Jul. 1997 | 4th Jul. 1997 |
| dmmmmyy | 4-July-97 | 4/July/97 | 4 July 97 | 4. July 97 | | 4. July, 97 | 4th July 97 |
| dmmmmyyyy | 4-July-1997 | 4/July/1997 | 4 July 1997 | 4. July 1997 | | 4. July, 1997 | 4th July 1997 |
| ddmyy | 04-7-97 | 04/7/97 | 04 7 97 | 04.7.97 | | | |
| ddmyyyy | 04-7-1997 | 04/7/1997 | 04 7 1997 | 04.7.1997 | | | |
| ddmmyy | 04-07-97 | 04/07/97 | 04 07 97 | 04.07.97 | | | |
| ddmmyyyy | 04-07-1997 | 04/07/1997 | 04 07 1997 | 04.07.1997 | | | |
| ddmmmyy | 04-Jul-97 | 04/Jul/97 | 04 Jul 97 | 04. Jul 97 | 04 Jul. 97 | 04. Jul. 97 | 04th Jul. 97 |
| ddmmmyyyy | 04-Jul-1997 | 04/Jul/1997 | 04 Jul 1997 | 04. Jul 1997 | 04 Jul. 1997 | 04. Jul. 1997 | 04th Jul. 1997 |
| ddmmmmyy | 04-July-97 | 04/July/97 | 04 July 97 | 04. July 97 | | 04 July, 97 | 04th July 97 |
| ddmmmmyyyy | 04-July-1997 | 04/July/1997 | 04 July 1997 | 04. July 1997 | | 04 July, 1997 | 04th July 1997 |

**Table 2** Day - Month - Year (European notation).

| | | | | | | |
|---|---|---|---|---|---|---|
| yymd | 97-7-4 | 97/7/4 | 97 7 4 | | | |
| yymdd | 97-7-04 | 97/7/04 | 97 7 04 | | | |
| yymmd | 97-07-4 | 97/07/4 | 97 07 4 | | | |
| yymmdd | 97-07-04 | 97/07/04 | 97 07 04 | | | |
| yymmmd | 97-Jul-4 | 97/Jul/4 | 97 Jul 4 | 97, Jul 4 | 97 Jul. 4 | 97, Jul 4th |
| yymmmdd | 97-Jul-04 | 97/Jul/04 | 97 Jul 04 | 97, Jul 04 | 97 Jul. 04 | 97 Jul. 04th |
| yymmmmd | 97-July-4 | 97/July/4 | 97 July 4 | 97, July 4 | | 97, July 4th |
| yymmmmdd | 97-July-04 | 97/July/04 | 97 July 04 | 97, July 04 | | 97 July 04th |
| yyyymd | 1997-7-4 | 1997/7/4 | 1997 7 4 | | | |
| yyyymdd | 1997-7-04 | 1997/7/04 | 1997 7 04 | | | |
| yyyymmd | 1997-07-4 | 1997/07/4 | 1997 07 4 | | | |
| yyyymmdd | 1997-07-04 | 1997/07/04 | 1997 07 04 | | | |
| yyyymmmd | 1997-Jul-4 | 1997/Jul/4 | 1997 Jul 4 | 1997, Jul 4 | 1997 Jul. 4 | 1997, Jul 4th |
| yyyymmmdd | 1997-Jul-04 | 1997/Jul/04 | 1997 Jul 04 | 1997, Jul 04 | 1997 Jul. 04 | 1997 Jul. 04th |
| yyyymmmmd | 1997-July-4 | 1997/July/4 | 1997 July 4 | 1997, July 4 | | 1997, July 4th |
| yyyymmmmdd | 1997-July-04 | 1997/July/04 | 1997 July 04 | 1997, July 04 | | 1997 July 04th |

**Table 3** Year - Month - Day (Sequence according to ISO 8601).

Strings like „Monday" ... „Sunday" or „Mon" ... „Sun" often indicate temporal data. However, e.g. „DEC" or „Sun" do not necessarily express temporal information. Thus, the homonym problem exists also in connection with temporal data. Its scope is visualized in Table 4 resp. Figure 2 where we present some search results for the number of documents that may make references to days. Work days are more often referenced at the WWW than the weekend days; this is shown by the smaller number of hits obtained for Saturday. Thus, one may assume that the larger number of counts for Sun results, e.g., from weather reports, description of lifestyles, or the hardware vendor. This example shows that the search mechanisms provided on the WWW influence the appropriateness of firm names. The associations connected with Sun are obvious but also acronyms like SAP are used in many contexts outside the enterprise management systems market (Knolmayer, 1996).

| | | | |
|---|---|---|---|
| Mon | 383 333 | Mon OR Monday | 880 006 |
| Tue | 304 314 | Tue OR Tuesday | 696 256 |
| Wed | 359 444 | Wed OR Wednesday | 743 065 |
| Thu | 305 793 | Thu OR Thursday | 726 759 |
| Fri | 364 095 | Fri OR Friday | 937 266 |
| Sat | 266 604 | Sat OR Saturday | 692 135 |
| Sun | 643 588 | Sun OR Sunday | 1 036 419 |

**Table 4** Search results (as of 1997-05-13).

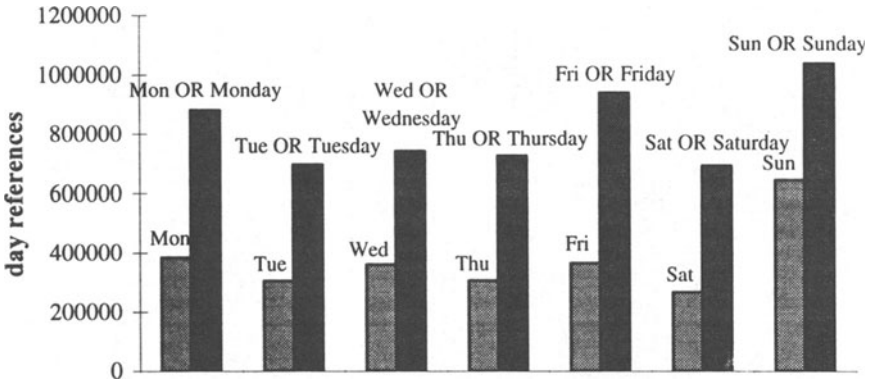


**Figure 2** Number of candidate day references (as of 1997-05-13).

A more powerful tool for maintaining integrity of temporal data would allow the webmaster to define those temporal representations which she expects to exist on the scanned sites. Temporal data like

Q1 96 (e.g., at http://www.intel.com/intel/finance/earnings/q1-96.htm)

H1 '96 (e.g., at http://research.refco.com/Docs/Samples/Q4.htm)

are easily identified e.g. by members of the financial service industries but may not be understood outside this subculture and by scanning tools. One may also come across entries like

95W05 (e.g., at http://www.ft.uni-erlangen.de/~mskuhn/iso-time.html) or

Thu Mar 6 6:05:37 1997

(e.g., at http://www.lottolink.com/UNSTATES/MO/HTML/FANTSKIP.HTML).

The difficulty in recognizing the last formulation is that the time is put *between* the day and year information. (All examples as of 1997-05-13).

More semantic reasoning is needed if verbal descriptions of dates should also be recognized, e.g., Christmas Eve, Whitsunday, Halloween or Valentines Day. If necessary, the tool could access a calendar routine to determine the verbally referenced date.

Another fuzzy string that might be relevant is „new". If a certain time interval since the creation or the last update of a WWW page has passed and the page contains the string or icon „new", the webmaster should consider whether the accompanying message can still be characterized in this way. There are several icons in use at the WWW to stress recent information. These icons are usually named with the file-extension .gif; thus, one may search for an icon named e.g. new.gif. Searching for the accompanying bit maps would be very cumbersome. Therefore, webmasters should support the maintenance of their pages by using the standard names of the icons which can be easily checked during the scanning procedure.

Not only the determination of the search expression but also the output of the scanning process may be organized differently. It is easy to output the string which has been detected as a candidate expression for temporal information. The most user *un*-friendly output is to count the number of characters scanned and to give the number of the character where the temporal data starts. A bit more user-friendly would be to output the information at which line of the WWW page temporal data have been detected. However, this is difficult to determine the line because it would need complex analyses of the HTML code. Another difficulty is that browsers automatically wrap lines depending on the width of the window. Thus, the same document may be wrapped differently.

A more sophisticated procedure would print the contexts in which the search strings have been found. One could print $u$ characters or $y$ words ahead of and $v$ characters or $z$ words behind the identified search string. From this extract the person who is maintaining the page could hopefully check whether the tense of the formulation is in accordance with the relationship between the present date and the date of the mentioned event.

A further step to user friendliness would be to print the whole sentence in which one of the search strings has been detected. In this case, a set of punctuation marks must be defined which should be interpreted as the beginning and the end of the sentence. One has to decide whether e.g. the semicolon should be handled as punctuation mark or not. Furthermore, dots are also used in abbreviations and additional problems result from the fact that the dot may also be used as separator in many temporal expressions without finishing the sentence.

The most user friendly implementation would be obtained if the scanner could determine by temporal reasoning whether a certain sentence is written in future or past tense and print the sentence only if there seems to be a contradiction between the tense used in the natural language expression and the difference between „now" and the temporal data mentioned.

### 4.2 Using the results of the search

The results of the scanning procedure can be

- used for judging the temporal integrity of the WWW page at the present moment
- applied for determining a prospective maintenance schedule to keep or recover the temporal integrity of the web pages
- stored in a WWW repository to facilitate access to data which are relevant in checking the temporal integrity of the pages.

## 5 SCANNING SEVERAL WWW PAGES FOR TEMPORAL DATA

### 5.1 Scanning the own web server

Temporal integrity should not only be regarded from the viewpoint of a single page. The possibility of temporal inconsistencies is higher if a larger number of web pages has to be maintained. Therefore, the webmaster should be supported by a procedure which scans *all* pages for which he is responsible.

One way to realize this goal is to use a web repository in which meta-data about web pages is registrated (preferably) automatically or manually. The derivation of structured meta-data from WWW pages is discussed in (Pitkow/Recker, 1995; Shklar/Shah/Basu, 1995; Shklar/Sheth/Kashyap/Shah, 1995). In the Multimedia Oriented Repository Environment MORE (Eichmann, 1996; University of Houston, 1996; MountainNet, 1997) a unique directory is created at change requests; all documents generated during the process are stored in it. The directory name consists of the originators login name and a date/time (mmddyyhhmm) stamp (i.e. "mjackson_1204971130"). Thus, some temporal data is already stored in this repository. However, all temporal information contained in a web page could be

extracted at its creation or update time and stored in the repository; checking for temporal integrity would become more efficient because accessing the repository is less time-consuming than scanning web pages. However, the performance of the *INT²IME* tool is very satisfying and the need for improving its performance seems to be small if it is used on the own server.

The second approach is to implement a web robot which follows the links of the pages and checks whether a certain page has already been visited or not. A web robot is a program that automatically traverses the web's hypertext structure by retrieving a document, and recursively retrieving all documents that are referenced. In usually starts from the root document or a list of URLs, especially of documents with many links elsewhere. Given those starting points a robot can select URLs to visit, parse and use as a source for new URLs (Koster, 1995). In our concept of supporting temporal integrity, the robot would choose only links with identical URL base. Thus, we will restrict the search to the set of WWW pages for which a certain webmaster is responsible.

## 5.2 Scanning external servers

If a service provider is in charge for the maintenance of his customer's web pages, he may run the tool at his own server without delivering it to the customer. Outsourcing of at least some webmaster activities has become a standard offering, e.g. by The Internet Outsourcing Group (1996): „TIOcom will perform all necessary WEB page updates, changes and additions, on a bi-weekly (or more often, if needed) basis. These could be due to normal system maintenance, additions you request or changes that you request due to normal business change (new product prices, new products, spec changes, limited time promotions, new phone and so on). We could also arrange for you to do all these changes and general WEB Site development, if need so." Other sources stress than maintenance of WWW pages should be done internally: „There are also lots of companies ... that will prepare web pages fairly inexpensively. You still need your webmaster and your websters to take responsibility for the content, but you can outsource the actual authoring. You will probably decide fairly soon that you need to have at least some of the authoring inside, however. Sites need to be kept up-to-date, and it's a lot easier if someone that's available all the time inside the company can do it" (Digital Mix, 1996).

Scanning an external WWW site may also be applied by a distributor of a maintenance tool who wants to demonstrate its potential for supporting temporal integrity on the live data found on the web pages of a potentially interested party.

One problem in scanning foreign sites is the ambiguity of date representations in different cultures; e.g., 07-04-1997 could mean the 4th of July to an American reader whereas an European might interpret it as the 7th of April. If the same page also contains a date like 13-12-1997 one can conclude that the webmaster uses the

dd-mm-yyyy and not the mm-dd-yyyy notation. Thus, scanning of foreign pages needs special attention to the date notation employed.

## 6 THE *INT²IME* PROTOTYPE AS JAVA APPLET

The prototype developed is named *INT²IME* because its goal is to keep the *INT*egrity of *TIME*-oriented data. Several options exist with respect to the implementation of the procedure described. In Table 5 we compare the pros and cons between possible implementations of our scanning approach in Java, C++ and Perl. General comparisons of these languages can be found e.g. in (Martin, 1996; Laroche, 1997; Swiss SunSite, 1997). On the basis of our comparison, Java was chosen as implementation language for the *INT²IME* prototype. In Figure 3 we show the application of the prototype to a page with many different time representations. We output the position of the first character of the date string and do not distinguish between presentations like feb and Feb and FEB, presenting the string in lower case characters. The prototype allows to scroll the output window.

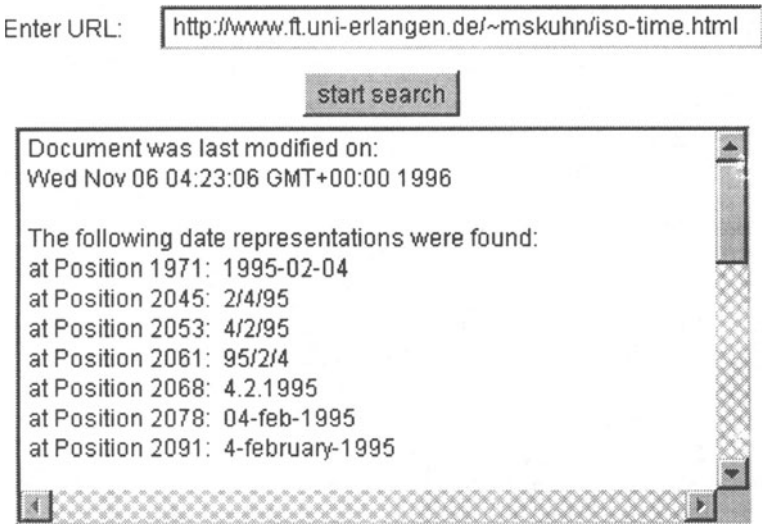


**Figure 3** Output of applying the *INT²IME* applet to http://www.ft.uni-erlangen.de/~mskuhn/iso-time.html (as of 1997-05-13).

| Java | C++ | Perl |
|---|---|---|
| + platform independent | – portability problems possible | + platform independent |
| – comparatively poor perfomance | + very high performance | – code must be compiled in each execution; may result in poor performance |
| + high robustness (no pointers, garbage collection, exception handling) | – pointer arithmetic and explicit memory management are error-prone | – danger of introducing type errors |
| + ease of programming and maintenance | – difficult and programmer-unfriendly | + ease of programming |
| + standard API for GUI, network, I/O | – API for GUI, network not in language specification | – primitive network functions; no API for GUI |
| + easy to embed into WWW (as applet) | – embedding into WWW cumbersome (e.g., via CGI) | – embedding into WWW cumbersome (e.g., via CGI) |
| – limited expressive power | + powerful and very expressive | + very powerful search options (regular expressions) |

**Table 5** Comparison of programming languages with respect to implementing a WWW maintenance tool.

In Fig. 4 we give an example in which $u=v=35$ characters encompassing the successful search string should be presented. The underlying HTML-code is

„<P>
The next meeting of the
<A HREF=/zobis/jahr2000/chig2000.html>CHIG2000</A>
will be held on <B>October 6, 1997</B> in Bern.
<P>
<HR> ...“

For ease of readability we filter the HTML-tags in presenting the output to the webmaster. The exact presentation of $u$ resp. $v$ characters needs to filter the HTML-text before counting the number of characters shown. Otherwise, one could accept $u$ resp. $v$ as fuzzy numbers and filter after selecting a certain substring. In any case, one will not continue the presentation if e.g. a dot is followed by a <P> because this signifies the existence of a new paragraph.

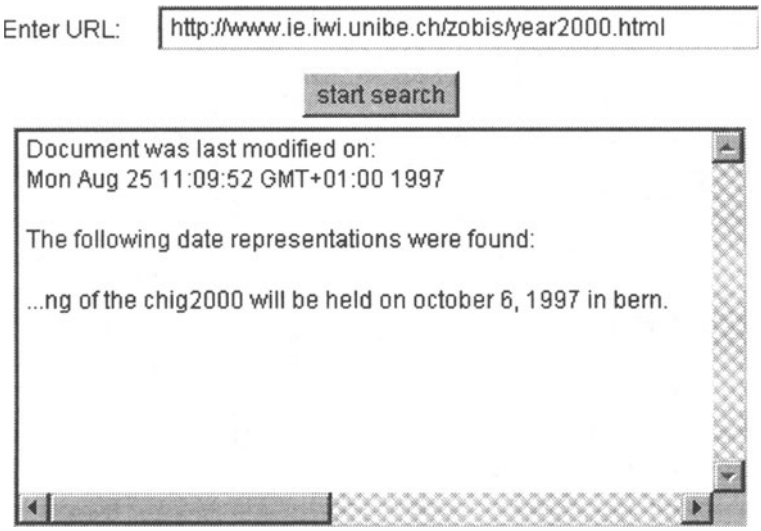


**Figure 4** Output of applying the *INT²IME* applet to http://www.ie.iwi.unibe.ch/zobis/year2000.html (as of 1997-08-28).

## 7 SUMMARY

In this paper we emphasize the existence of a vast amount of temporal data on the WWW and the necessity of supporting webmasters in maintaining these data. We discuss the options for scanning WWW pages for temporal data and describe a Java applet which has been developed for this purpose.

## 8 REFERENCES

Aimar, A. et al. (1995) WebLinker, a tool for managing WWW cross-references. *Computer Networks and ISDN Systems*, **28**, 99-107.

Alonso, G., Hagen, C., Schek, H.-J. and Trech, H. (1997) Towards a Platform for Distributed Application Development, in *Advances in Workflow Management Systems and Interoperability* (ed. A. Dogac et al.), NATO Advanced Study Institute, Istanbul.

Bair, J. (1996) It's About Time! Supporting Temporal Data in a Warehouse. *Info DB*, **10**, 1, 1-7. (cf. http://www.leep.com/what.html)

Bichler, M. and Nusser, S. (1996) Modular Design of Complex Web-Applications with W3DT, in *Proceedings of the 5th Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*. IEEE Computer Society Press, Los Alamitos, 328-333.
http://dec9.wu-wien.ac.at/w3dt/wetice/wetice.html

Digital Equipment Corporation (1997) AltaVista
http://www.altavista.telia.com/cgi-bin/telia?country=de&lang=de

Digital Mix (1997) Industrial Marketing on the Internet
http://www.digitalmx.com/industrial/book/planning.html

Eichmann, D. (1996) MORE 2.0 Overview
http://rbse.jsc.nasa.gov:81/DEMO/

Fielding, R.T. (1994) Maintaining Distributed Hypertext Infostructures: Welcome to MOMspider's Web. Paper presented at the *First International World-Wide Web Conference (WWW94)*, Geneva.
http://pigeon.elsevier.nl/cgi-bin/WWW94link/03/overview

Fisher, A.S. (1988) CASE: Using Software Development Tools. Wiley, New York et al.

Gertz, M. and Lipeck, U. (1995) „Temporal" Integrity Constraints in Temporal Databases, in *Recent Advances in Temporal Databases* (ed. J. Clifford and A. Tuzhilin), Springer, Berlin et al., 72-92.

Graham, I. (1996) PC HTML EDITORS
http://moon.inf.uji.es/Ian/HTMLdocs/PCTOOLS/pc_editors.html

Holloway, S. and Bidgood, T. (1991) CASE Handbook for Information Managers. Avebury, Aldershot.

IBM (1997) The Year 2000 and 2-Digit Dates. Guide Document Number: GC28-1251-06. 6th ed., Poughkeepsie 1997-04-11
http://ppdbooks.pok.ibm.com:80/cgi-bin/bookmgr/bookmgr.cmd/books/y2kpaper/

Isakowitz, T., Stohr, E.A. and Balasubraman, P. (1995) RMM: A Methodology for Structured Hypermedia Design. *Communications of the ACM*, **38**, 8, 34-44.

ISO 8601 (1988) Data elements and interchange formats - Information interchange - Representation of dates and times. Reference number ISO 8601: 1988 (E). First ed., Geneva.

Jensen, C.S. et al. (Eds.) (1993) A Consensus Glossary of Temporal Database Concepts. Technical Report R 93-2035, Aalborg University November 1993. Also as digest in: *SIGMOD Record* **23**, 1, 52-64.

Kahn, R.L. (1996) Building Your Home on the World Wide Web; Researching, Designing, and Maintaining a WWW Home Page. http://www.arsc.sunyit.edu/~com400/spec2.html.

Knolmayer, G. (1996) SAP-Produkte und deren Umfeld auf dem Internet. *Wirtschaftsinformatik* **38**, 87-93. http://www.ie.iwi.unibe.ch/sap/wisurf01.html.

Knolmayer, G. and Leuenberger, B. (1997) CD-ROMs - Eine brauchbare Alternative zum World-Wide-Web? *Wirtschaftsinformatik* **39**, 5, to appear.

Knolmayer, G. and Myrach, T. (1997) Die Berücksichtigung fehlerhafter Daten durch historisierende Datenhaltung, in *Rechnungslegung und Prüfung* (ed. T. Fischer and R. Hömberg), IDW 1997, Düsseldorf, to appear.

Koster, M. (1995) The Web Robots FAQ. http://info.webcrawler.com/mak/projects/robots/faq.html.

Laroche, E. (1997) Differences between Java and C++. http://www.access.ch/lr/java/javacppdiffs.html

Martin, R.C. (1996) Java and C++, A critical comparison. http://www.innergy.com/idm/v1n5/idm0996a.html.

Microsoft (1997) Online-Help for Excel, English Version 8.0, Redmond.

Millman, H. (1997) www.toolboxes. *Computerworld* **31**, 22, 79.

MountainNet (1997) What is MOREplus. http://rbse.mountain.net/MOREplus/.

Montanari, A. and Pernici, B. (1993) Temporal Reasoning, in *Temporal Databases* (ed. A.U. Tansel et al.), Benjamin/Cummings, Redwood City et al., 534-562.

Myrach, T., Knolmayer, G.F. and Barnert, R. (1996) On Ensuring Keys and Referential Integrity in the Temporal Database Language TSQL2, in *Databases and Information Systems* (ed. H.-M. Haav and B. Thalheim), Proceedings of the Second International Baltic Workshop, Volume 1: Research Track, Tallinn, 171-181.

Pitkow, J.E. and Recker, M.M. (1995) A Simple Yet Robust Caching Algorithm Based on Dynamic Access Patterns. http://www.vuw.ac.nz/~mimi/www/www-caching/caching.html.

Rubin, H.A. (1996) Millennium Metrics: Truth & Consequences. *Proc. 10th International Conference on Software Maintenance & Software Management*, Year 2000 Solutions, Bethesda.

Schmitz, U. (1997) Bei der Suche nach Studieninformationen müssen die Surfer jede Uni abklappern. *Computer Zeitung* **28**, 34, 8.

Shklar, L., Shah, K. and Basu, C. (1995) Putting Legacy Data on the Web: A Repository Definition Language. *Computer Networks and ISDN Systems*, **27**, 939-951.

Shklar, L., Sheth, A., Kashyap, V. and Shah, K. (1995) InfoHarness: Use of Automatically Generated Metadata for Search and Retrieval of Heterogeneous Information, in *Proceedings CAiSE'95* (ed. J. Iivari, K. Lyytinen and M. Rossi), Springer, Berlin et al., 217-230.

Snodgrass, R.T. (Ed.) (1995) The TSQL2 Temporal Query Language. Kluwer, Boston et al.

Snodgrass, R.T., Böhlen, M.H., Jensen, C.S. and Steiner, A. (1996a) Adding Valid Time to SQL/Temporal. Change Proposal, ANSI X3H2-96-501r2.

Snodgrass, R.T., Böhlen, M.H., Jensen, C.S. and Steiner, A. (1996b) Adding Transaction Time to SQL/Temporal. Change Proposal, ANSI X3H2-96-502r2. ftp://ftp.cs.arizona.edu/tsql/tsql2/sql3/

Swiss SunSite (Ed.) (1997) Perl versus ... Here are some comparative language discussions. ftp://sunsite.cnlab-switch.ch/mirror/CPAN/doc/FMTEYEWTK/versus/index.html.

The Internet Outsourcing Group (1996) WEB Services. http://www.tio.com/web.html

University of Houston at Clear Lake (1996) The RBSE Program. http://rbse.jsc.nasa.gov:81/MORE/.

Wreggit, D.J. (1995) Software Agents Using Java. Research Report CS694c, Science Applications International Corporation. http://www.saic.alaska.net/wreggit/djw_paper.html.

## 9 BIOGRAPHIES

Gerhard F. Knolmayer was born in Vienna in 1948. He received a master and a Ph.D. degree from the Vienna University of Economics and Business Administration. In 1979 he became Professor of Business Administration at the University of Kiel. Since 1988 he is Professor at the Institute of Information Systems of the University of Bern. He currently serves as Vice-Dean of the Law and Economics Faculty, as chairman of the SIG „Business Information Systems" of the German Computer Society, its subgroup „Time-oriented Business Information Systems" and of the Swiss Interest Group in Solving the Year 2000 Problem. His research interests include enterprise management systems, business rules and workflow management, and temporal IS.

Thomas Buchberger was born 1973 in Bern, Switzerland. He is a graduate student at the Institute of Computer Science and Applied Mathematics of the University of Bern. His research interests include Computer Graphics and Software Engineering.