TOPINCS: A SOFTWARE FOR RAPID DEVELOPMENT OF WEB DATABASES

Robert Cerny
Independent Software Consultant, Robert-Fuchs-Gasse 22, A-1140 Wien, Austria

Keywords: Tools, Information storing, Information sharing, Semantic technology, Rapid application development, Topic maps.

Abstract: Topincs is a software for rapid development of web databases on top of the familiar LAMP stack. It overcomes the slow development of rigid RDBMS based applications by using the semantic technology Topic Maps. A Topincs store is developed by modeling a Topics Maps schema. This schema is used for form generation and powers a virtual domain specific OOP interface which is used in services to transform, aggregate, and modify the data. Topincs takes over well known usage patterns from the traditional wiki, but replaces markup authoring with a form based approach for data entry. In addition any technological jargon is hidden from the user. This combination minimizes the entry barrier, so that a new user who is familiar with the domain of the web database can be pointed to the start page from where he can follow his nose.

1 INTRODUCTION

The wiki technology has reached its peak with the omnipresent Wikipedia. But many smaller communities also use wikis as a tool to store and share *their* information. Enterprises are beginning to realize the power this technology holds and are learning how to use it for their benefit (Stocker and Tochtermann, 2009). The obvious success of the technology overshadows the fact that wikis have brought three powerful usage patterns to a broader audience:

- *One Topic per Page and One Page per Topic*. All statements on the page should be about one topic. But also everything the community knows about this topic should be on its page in the wiki.
- *An enclosed Referral Space*. The majority of links in a wiki link to pages within the wiki. This takes into consideration that the group of wiki users, no matter how it is assembled, form a community. A wiki offers a single place to put things and where to start looking.
- *Editing in the Browser*. The pages of a wiki can be easily edited in the web browser. This feature was anticipated in the web architecture from the very beginning. Older web browsers like the Netscape Navigator had a menu item *Put* for saving new versions of the web page. But only the wiki made editing in the browser practical for many people.

These three patterns form a strong alliance and allow a generic viewing and editing approach to any sort of information. But there are also drawbacks in the wiki technology from the point of view of an enterprise:

- A markup technology has to be learned. Syntactic structuring of text is an everyday task for programmers and computer scientists which leads to the false believe that it is easy and everyone can do it.
- It takes time and effort to communicate simple facts by means of natural language in writing. This, in the nineteenth century, led Charles Babbage to come up with a text with blank spaces to fill in, the *form* (Babbage, 1835).
- It is not possible to computationally access and create value from the information in a wiki.

2 OVERVIEW OF TOPINCS

Topincs is best described as a wiki where authors use forms to edit pages. In this sense it is similar to a semantic wiki, but does not require any technological skill to enter information. A storage unit of Topincs is called a *web database* or *store*.

A Topincs web database allows a group of users to share information about one domain. Not only hu-

Cerny R..
TOPINCS: A SOFTWARE FOR RAPID DEVELOPMENT OF WEB DATABASES.
DOI: 10.5220/0003657001870194
In *Proceedings of the International Conference on Knowledge Management and Information Sharing* (KMIS-2011), pages 187-194
ISBN: 978-989-8425-81-2
Copyright © 2011 SCITEPRESS (Science and Technology Publications, Lda.)

mans can act as data provider and consumer, but also computer programs. It also offers a generic virtual domain specific OOP interface to program services that support the activities of the user group. While domain expertise has to be available when setting up a Topincs web database, the exact boundaries of the domain do not have to be known to start modeling the schema. In the center of schema development is the *topic type*. Examples for topic types are *Person*, *Tournament*, and *Scientific Paper*. The design and development of a schema with ten topic types and various relationship types takes approximately half a day. Then users can be pointed to the start page of the web database and start recording information. Extending the schema with new attributes or relationships can be done in little time while the database is in use.

This paper uses screenshots to illustrate the user interface. Unfortunately not all details and aspects can be captured this way. It is recommended that the reader tries Topincs himself to gain a better understanding of the text. This is possible in the Topincs trial store about *Movies* at http://www.topincs.com/trial/movies/. Use *Guest* for user name and password.

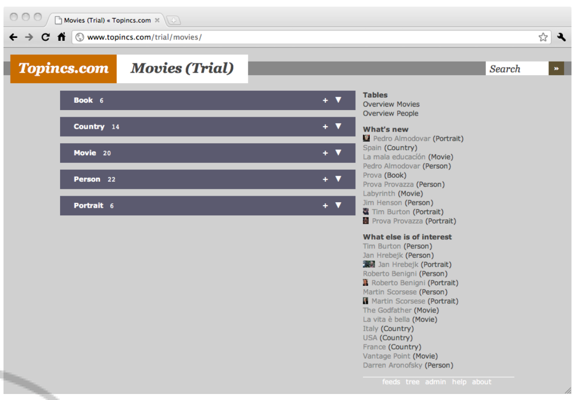
2.1 User Interface Components

From the start page users can create new topics or reach existing ones which they can edit in a form. On every page of the web database there is a link back to the start page. We will describe the main components of the user interface namely *start page*, *topic page*, *form page* and *tables* in greater detail.

2.1.1 Start Page

The start page of a Topincs web database is the main entry point for the user. It allows direct access to all topic pages. It is segmented into an index grouped by type and a list of recently created and edited topics. This segmentation mirrors the partition of the human memory into long-term memory and working memory. Frequently the user does not need to consult the index since he sees what he is looking for in the list of recent items.

The direct access to any topic from the start page implies that every recorded statement is only one click away and can be accessed from any perspective. This allows the user to reach the information from his current thought, which helps to avoid the classification problem (Lansdale, 1988). Suppose you have a Topincs web database holding family tree data and you want to know who the mother of Joe is. You can access this fact by looking up Joe in the index under *Man*. Accessing the topic page of Joe will show you


Movies (Trial) « Topincs.com
www.topincs.com/trial/movies/
Topincs.com
Movies (Trial)
Search
Book 6
Country 14
Movie 20
Person 22
Portrait 6
feeds tree admin help about

Tables
Overview Movies
Overview People

What's new
Pedro Almodovar (Portrait)
Spain (Country)
La mala educación (Movie)
Pedro Almodovar (Person)
Prova (Book)
Prova Provazza (Person)
Labyrinth (Movie)
Jim Henson (Person)
Tim Burton (Portrait)
Prova Provazza (Portrait)

What else is of interest
Tim Burton (Person)
Jan Hrebejk (Person)
Jan Hrebejk (Portrait)
Roberto Benigni (Person)
Roberto Benigni (Portrait)
Martin Scorsese (Person)
Martin Scorsese (Portrait)
The Godfather (Movie)
La vita è bella (Movie)
Italy (Country)
USA (Country)
France (Country)
Vantage Point (Movie)
Darren Aronofsky (Person)

Figure 1: The start page of the trial web database about movies.

who his mother is. Let us assume it is Mary. This can also be accessed from the other angle. If you ask the question "Who are the children of Mary?", you look up Mary in the index under *Woman* and by accessing her topic page you will find a list of her children, Joe among them.

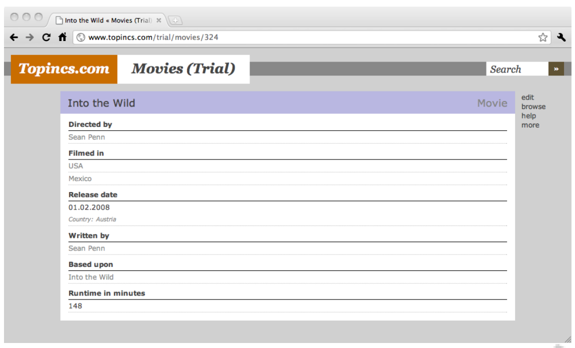
It is crucial to the success of the system that all users have a shared understanding of the domain. A user must feel familiar with the terms he sees on the start page as if it was an extension of his memory. Designing a Topincs web database should always be done in close collaboration with the users to gain an understanding of their view of the domain.

2.1.2 Topic Page

A topic page displays all statements that are recorded about one topic like a fact sheet. It does not matter whether the user has edited this page or referred to this topic in another page. There is a layout separation of text and data. Some things cannot be formalized, e.g. a quote from a text, a margin note, or the description of a software bug. Topincs allows to enter plain text or wiki markup. If data and text are available, text is displayed to the left and data are displayed to the right. In the data section associations to other topics of the web database are rendered as links and allow associative browsing.

Topic pages can be browsed in chronological order which resembles the core idea of Lifestreams of going back and forth through the information one encounters like in a diary (Freeman and Gelernter, 1996).

The topic page can be viewed in different formats, most of them are targeted to machines. But it can also be retrieved as plain text which is handy for copying information into emails or other text documents. Alternatively the user can use the URI of the page.


Into the Wild « Movies (Trial)
www.topincs.com/trial/movies/324
Topincs.com Movies (Trial) Search

Into the Wild — Movie

edit
browse
help
more

Directed by
Sean Penn

Filmed in
USA
Mexico

Release date
01.02.2008
Country: Austria

Written by
Sean Penn

Based upon
Into the Wild

Runtime in minutes
148

Figure 2: A topic page about a movie.

2.1.3 Form Page

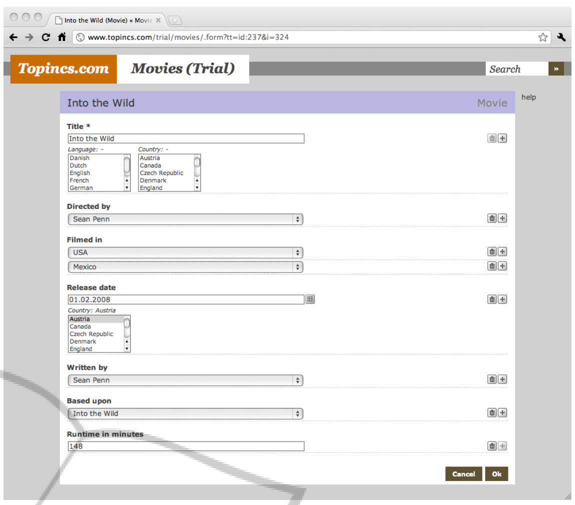
The form for editing a topic page is auto-generated from the schema. It is possible to influence the order of the fields and this is respected in the topic page as well. All form fields allow the specification of a minimal and maximal cardinality and explanatory information. There are specialized editors for date, time, boolean, wiki markup, and various number types.

An association between two topics is created by using a drop down box with some special features:

- It allows only to connect to topics that make sense given the schema, e.g. only a person can be the author of a paper.
- Since there can be thousands of possible options, only the ten most likely are presented directly. These are new ones and topics that were recently used in a similar way. All options can be accessed by entering a separate dialog via the last option *More* which also allows to search by name.
- It is possible to create new topics in the select box, if a topic is not present. For example, to create a new person the user chooses *New: Person* and enters a nested form. The parent form is on hold until the child form is completed or cancelled. This procedure can be repeated to any depth.
- The advanced features described here can be deactivated so that just an ordinary drop down box is presented which simply contains all options.

The form page as well as the other main components can be used solely with the keyboard in order to speed up data entry. The mouse is still better suited for exploring a web database through associative browsing.

Computation and organization make it necessary to guarantee a certain data quality. Validation rules can be specified for a form. These range from mandatory fields, over unique values to matching of a regular expression. Customizable messages make it easy


Into the Wild (Movie) « Movie
www.topincs.com/trial/movies/.form?tt=id:237&i=324
Topincs.com Movies (Trial) Search

Into the Wild — Movie

help

Title *
Into the Wild
Language: - Danish, Dutch, English, French, German
Country: - Austria, Canada, Czech Republic, Denmark, England

Directed by
Sean Penn

Filmed in
USA
Mexico

Release date
01.02.2008
Country: Austria
Austria, Canada, Czech Republic, Denmark, England

Written by
Sean Penn

Based upon
Into the Wild

Runtime in minutes
148

Cancel Ok

Figure 3: Editing a movie.

for the user to correct errors. A form with validation errors cannot be submitted.

2.1.4 Table

The topic page and form page handle many statements about *one* subject. Many times it is more convenient to display and edit statements about *many* subjects simultaneously as the broad adoption of spreadsheets shows. Topincs allows quick creation of tables by specifying *extent* and *columns*. Once a table is declared it can be reached from the start page. A table cell in Topincs can hold more than one statement unlike in spreadsheet applications where there is only room for one. The current implementation is fast enough to display thousands of rows and still shows good performance when sorting. It is possible to edit cells, but not to add new rows.

2.2 Topic Maps

At the core of Topincs lies the semantic technology *Topic Maps*. It uses the Topic Maps Data Model (ISO/IEC 13250-2, 2006) and the Topic Maps Constraint Language to achieve its flexibility (ISO/IEC FCD 19756, 2009). A familiarity with these complex technologies is essential to a successful long-term application of Topincs. Yet, schema modeling can be learned easily in a hands-on fashion when instructed by a tutor since there is immediate feedback, and no programming skills are necessary.

2.2.1 Topic Maps Data Model

The Topic Maps Data Model (TMDM) represents knowledge by connecting *topics* through *associa-*

tions. The connected topics play a certain *role* in an association, e.g. in our previous example, Joe is the *Child* and Mary is the *Mother*. A topic is an instance of one or more topic types. *Mary* is a *Woman*. A type hierarchy can be formed around topic types. *Woman* is a kind of *Person*.

Some statements only have one subject, e.g. *Joe's first name is "Joe"* or *Mary is born on the 13th of May, 1980*. These constructs are called *names* and *occurrences*.

2.2.2 Topic Maps Constraint Language

The Topic Maps Constraint Language (TMCL) is a mechanism to decide whether a given collection of statements, a *topic map*, conforms to a certain form, the *schema*. A TMCL schema is composed of *constraints* which specify certain aspects of topic types and statement types. Here are some examples for constraints expressed in natural language:

- A *Person* must have a *first name*.
- A *Motherhood* relationship has exactly one *Mother* and one *Child*.
- A *Woman* can play the role *mother* 0 to n times in a *Motherhood* relationship.
- A *Person* can play the role *child* 0 or 1 time in a *Motherhood* relationship.

Modeling in Topincs is the creation of topic types, statement types and constraints. A developer does this with the same generic data entry approach that is used by the user to edit domain data. The only difference is that in this case the TMCL meta schema is interpreted for form generation.

2.3 Architecture

Topincs is realized on top of Apache, MySQL and PHP. It is recommended to use Linux as a server operating system, but Mac OS X or Windows will suffice as well. Any recent version of the major web browsers can be used to access and edit information as well as mobile devices like the iPhone or the iPad.

The MySQL database scheme holds tables for the TMDM constructs and some auxiliary tables for performance reasons. For every topic map item stored in the database an id is created and the time and user of the modification or creation is stored so that it is possible to reconstruct how information came about.

The client uses JavaScript to make editing comfortable and to realize the difficult task of conveniently editing a graph structure in arbitrary depth. For communication between client and server a nearly RESTful web interface (Cerny, 2007) is used to create information items and allow concurrent editing of one topic page.

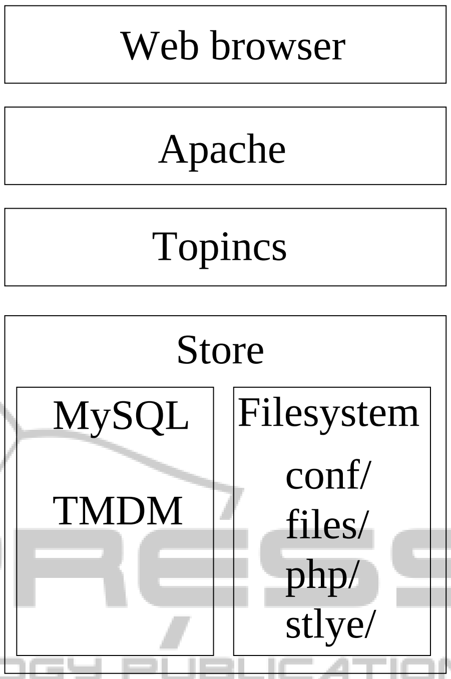


Figure 4: Architecture of Topincs.

2.4 Programming

The programming interface to a Topincs web database is virtual, it must neither be declared nor generated. It is given, once the developer assigns *serialization names* to the statement types of the schema. It can be used in domain classes which perform some computation, and in services which support users in their activities.

2.4.1 Tobjects and Domain Classes

An object representation of a topic is called *tobject* and has methods for data access and manipulation. For example a tobject representing a *Person* might have methods like `get_date_of_birth`, `set_date_of_birth`, `get_children`, and `set_mother`. A domain class is a PHP class which is put over a tobject to extend its API with methods that perform some computation. The following example is taken from the issue tracking use case which will be presented later and computes the duration of a work session for which the users record start and end time:

```
class WorkSession extends Tobject {

  function compute_work_time_in_sec() {
    return $this->get_end()->format("U")
      - $this->get_start()->format("U");
  }
}
```

Any number of domain classes can be put over a tobject in mix-in style allowing multiple inheritance. This is convenient since a topic can be instance of more than one type or might inherit features from abstract types.

Since program code subscribes to the schema used for a domain, it is important to be prepared for change and make it as easy as possible. Existing code does not have to be altered at all since an adapter domain class can satisfy the old interface. Importing the adapter is the sole change that needs to be performed in code artifacts that use the *old* schema.

2.4.2 Services

Services help the users in their activities by transforming, aggregating, or manipulating the data in the web database. A service has a number of arguments which can be bound to a topic (e.g. a *Person*, a *Book*) or a value (a date, a number). Services are accessible from the start page where all parameters are queried. But they can also be invoked from topic pages where the fitting argument is bound to the topic displayed.

The domain class in the previous example is used in a service which computes the total time for an issue by iterating over all assigned work sessions and building the sum. This service has one argument which accepts an *issue*, thus it will be offered at topic pages which display a bug, a change, or an enhancement.

2.5 Archiving Files

Topincs allows to integrate files into the schema. Upon creation of such a topic, the user has to upload a file. This file will be stored in the server file system. Miniatures of various sizes are created, if possible, and integrated in the user interface.

This feature opens up many possibilities for annotation. Pictures can be annotated with the people they depict and the occasion they were taken at. Emails and documents can be assigned to issues. Slides can be assigned to the presentation they were made for.

For some, it might suffice to identify the file with the work it contains, for others, it might be necessary to draw a clear line between the two, e.g. one picture can be contained in many different files with varying formats or resolutions. Topincs considers two files the same if their byte sequences are identical.

2.6 Scalability

A Topincs web database exposes the domain directly to the user like the Naked Object technology does as well (Pawson and Matthews, 2001). It is not recommended to store millions of topics and statements in *one* database since this would reduce performance and also make it impossible for humans to handle the data directly. Scalability tests have shown that a store with 20000 topics and double as many associations still shows satisfactory computational performance.

Topic Maps allows the distribution of topics and statements over various sources. Topincs supports this, so that e.g. information about one person can be distributed in an organization over various databases. The topic page in one web database will contain links to other web databases when the identity of the subjects has been declared. This also applies to the statement level so that fragments of one database can be mirrored or transcluded in another database (Cerny, 2008a).

2.7 License

Topincs is distributed under a proprietary license which restricts usage in the Internet to non-commercial applications. In a private network there are no usage restrictions.

3 FROM PERSONAL KNOWLEDGE BASE TO WEB DATABASE

The original motivation to start developing Topincs was to create a *Personal Knowledge Base* which Davies defines as “an electronic tool through which an individual can express, capture and later retrieve the personal knowledge he or she has acquired” (Davies, 2011). Vannevar Bush mentioned this concept already 1945 in his influential article “As We May Think” and referred to it as the *Memex*.

A first prototype of Topincs was presented at an open space session of the *International Workshop on Topic Maps Research and Applications* in 2005 in Leipzig. This version did not use Topic Maps, but rather a proprietary data model. The central focus lay on the recording of *encounters* with *topics* in *resources* in order to create a *knowledge log* (Sigel, 2006).

The next step was to modify the software to use the Topic Maps Data Model followed by the search for a user interface that anyone can use. The Topincs

Wiki (Cerny, 2008b) turned out to be a dead end. It was too cumbersome to fetch form fields from a list and required too much explanation altogether because it broke with well established usage patterns. Up to here, Topincs did not use an explicit schema. It was based on *ontological reflection* (Cerny, 2008b). In order to extend the system an administrator had to create a prototypical statement.

The demand for an easy user interface ultimately lead to forms as they represent the most common approach for data entry. This also meant giving up the idea of prototyping and ontological reflection. An explicit schema is necessary to generate forms, because it is hard to infer cardinalities reliably by reflection.

Already in the early stages it became evident that the design of the system allowed more than one person to access and edit a store. Particularly helpful was the influence that REST (Fielding, 2000) had on the architecture:

- Every data item is addressable through an URI and can be modified independently. A topic page which is composed of many data items can consequently be edited simultaneously by more than one person. Currently conflicts are resolved by letting the last one win, but this can be adjusted to require human attention.
- All communication between client and server is stateless. This allows the server to fully interpret each request individually without considering any context that might have been built up in any previous communication. For this reason the schema can be altered while the database is in use.

4 USE CASES

In the following the versatility of Topincs will be illustrated with three very different real world use cases. *Issue tracking* poses a solution for which standard software products exist, but the installation time of a foreign product would have exceeded the development time of a Topincs solution. *Managing readings and margin notes* exemplifies how a Topincs solution can reach into the real world. *Organizing a sports team* describes an application where a group of authors work in collaboration to relieve a coordinator from organizational work and confirms that no training is necessary to use Topincs.

4.1 Issue Tracking

The issues web database holds information relevant to the development of Topincs. While we were exposed to many issue tracking systems in a professional environment and could have used any of the free ones on the market, the realization of this solution with Topincs was intended to be a proof of concept. But it most certainly offers the added benefit that it can be extended ad hoc to fulfill new requirements.

The overall effort that has gone into developing the system was approximately one day since the solution was setup up in 2007. The following tasks are performed to our full satisfaction on a regular basis:

- Three types of issues are recorded: bugs, changes, and enhancements. They have many things in common but also show subtle differences, e.g. a bug is always discovered in a product version whereas an enhancement is not.
- When work on a new product version starts, the release is planned by assigning issues to this version or delaying them to a later version.
- Work sessions are recorded on a daily basis to get an overview of how much time goes into which issue and, overall, into a product version.
- When committing changes to the version control system, the URI of the issue is used as the commit message.
- A plain text version of the release notes is generated on demand by a separated process when a product version is finished. The generated file is included in the distribution file.
- The topic page of the product version is used as online release notes and its URI is sent out in announcements.

4.2 Managing Readings and Margin Notes

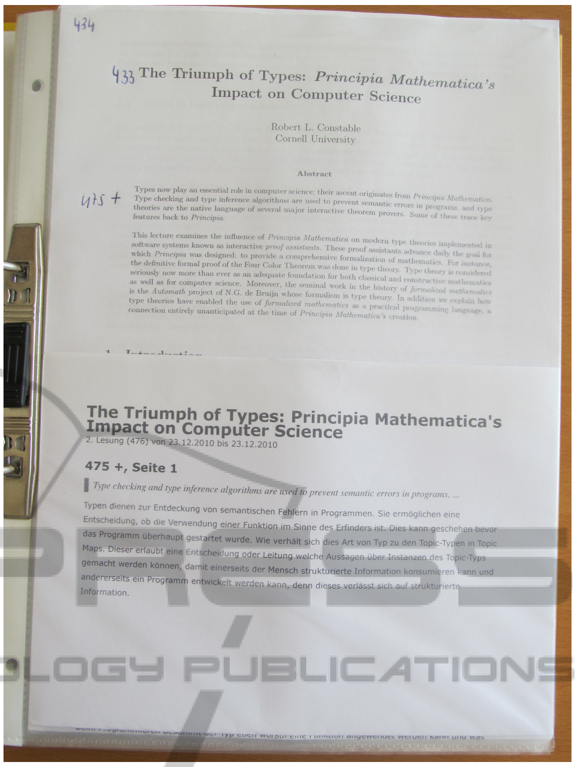
Reading of texts is still the main activity in acquiring knowledge, and, quite likely will always be. We are using a Topincs web database to manage all information regarding our readings. There is a clear distinction between the abstract work, e.g. novel or scientific paper and the item which is archived. Archiving happens digitally by uploading a PDF to the web database or physically by filing a document into a binder in strict chronological order by filing date. The document is annotated with the ids the database uses for the document, the abstract work, and the margin notes. The margin notes for one reading are assembled by a service into a document. A printout is filed together with the document. Figure 5 shows the physically archived document.

The overall time that has gone into setting up the system was approximately half a day including its design. There are a couple of significant advantages over the solutions we previously used:

- The main access point into the system is the start page of the Topincs web database, all information regarding our readings is reachable from there. Access to a physical document also starts there by looking up its id.
- There is no doubt about the filing of physical documents: by id in ascending order and thus strictly chronological by filing date. This simplifies the task of filing, because it avoids the classification problem and the piling strategy that tries to compensate it (Lansdale, 1988; Malone, 1983).
- The simplification of filing goes hand in hand with simplification of finding a document. Human memory is relieved of the unnecessary tasks of having to remember in which binder a document is.
- The system can be easily explained to novices where as classification systems are usually hard to learn and debatable since they allow room for different interpretations.
- Margin notes in digital form allow full text search which would not be possible with scribblings on the document, but the filing of the margin notes with the document maintains the important physical connection between the two of them.
- The system gives a clear structure to all relevant tasks, e.g. reading a paper, taking notes, and filing a document, and therefore eases their implementation.
- The web database has a table which lists all readings in reverse chronological order.
- If the digital information of the solution in Topincs is lost, all information is still present in the binders. This also addresses issues with the longevity of digital information (Rothenberg, 1995).
- The margin notes in the web database hold the exact quote. This makes it possible to create annotations for a digital version of the document if it should become necessary.

4.3 Organizing a Sports Team

Even in a small leisure time sport team of 15 people a lot of information is produced. Many things need to be known by all members on a frequent basis, e.g. "When and where will that meeting be again?", "Will enough people come to practise? Will it take place?", "When is that tournament again? Who will


434

433 The Triumph of Types: *Principia Mathematica's* Impact on Computer Science

Robert L. Constable
Cornell University

Abstract

475 + Types now play an essential role in computer science; their ascent originates from *Principia Mathematica*. Type checking and type inference algorithms are used to prevent semantic errors in programs, and type theories are the native language of several major interactive theorem provers. Some of these trace key features back to *Principia*.

This lecture examines the influence of *Principia Mathematica* on modern type theories implemented in software systems known as interactive *proof assistants*. These proof assistants advance daily the goal for which *Principia* was designed: to provide a comprehensive formalization of mathematics. For instance, the definitive formal proof of the Four Color Theorem was done in type theory. Type theory is considered seriously now more than ever as an adequate foundation for both classical and constructive mathematics as well as for computer science. Moreover, the seminal work in the history of *formalized mathematics* is the *Automath* project of N.G. de Bruijn whose formalism is type theory. In addition we explain how type theories have enabled the use of *formalized mathematics* as a practical programming language, a connection entirely unanticipated at the time of *Principia Mathematica's* creation.

The Triumph of Types: Principia Mathematica's Impact on Computer Science
2. Lesung (476) von 23.12.2010 bis 23.12.2010

475 +, Seite 1

Type checking and type inference algorithms are used to prevent semantic errors in programs, ...

Typen dienen zur Entdeckung von semantischen Fehlern in Programmen. Sie ermöglichen eine Entscheidung, ob die Verwendung einer Funktion im Sinne des Erfinders ist. Dies kann geschehen bevor das Programm überhaupt gestartet wurde. Wie verhält sich dies Art von Typ zu den Topic-Typen in Topic Maps. Dieser erlaubt eine Entscheidung oder Leitung welche Aussagen über Instanzen des Topic-Typs gemacht werden können, damit einerseits der Mensch strukturierte Information konsumieren kann und andererseits ein Programm entwickelt werden kann, denn dieses verlässt sich auf strukturierte Information.

Figure 5: A document is filed together with a print out of its margin notes in a binder. The ids that the web database assigns to the document, the abstract work and the margin notes are written on the document.

be going?". In this solution a shared login to a Topincs web database was used to offer information retrieval and recording services to the members of the team. They had no training on how to use Topincs but were generally able to perform their tasks by following their nose. One problem caused by an outdated web browser surfaced within a six months' period. The design and development of the schema took approximately half a day. The following organizational tasks were performed:

- Collection of information about the players: name, email address, and phone number.
- Collection of information on a dress order: shirt number, clothing size, and quantities.
- Collection of information on participation of meetings, practises, and tournaments.
- Distribution of information on event dates and locations.

While previous solutions relied on a coordinator to integrate all information collected via email or a wiki, this solution reduced the work load by shifting the burden invisibly to the team members who created more structured data by filling out forms. The access to information for team members was improved. They had all information they needed to know in one place. The start page used familiar terms and provided a gateway to all relevant information.

5 OUTLOOK

A query language can help Topincs in many ways, but in particular helps to get more *common sense* into the application, so that data entry becomes even easier. For example it would be possible in the issue web database described earlier to restrict the selection of issues for a product version during release planning to *unresolved* issues. The validation of a *work session* in the same store could help the user recognize that he entered a *end time* that lies before the *start time*. The standardization of a query language for Topic Maps is in process.

6 CONCLUSIONS

Topincs is a tool that offers quick help in many situations where there is need to store and share information so that it is accessible for humans and computers alike. It combines the simplicity of a wiki with the power of an application. Its generic approach to viewing and recording information relieves the user of having to learn new user interfaces for different applications and subsequently lowers training costs in organizations. Development costs for solutions are significantly reduced since error prone and slow programming tasks are avoided. There is no need for expensive manual user interface tests since the generic source code runs in many web databases and is constantly in use.

The absence of hierarchies and the presence of the domain vocabulary of the user make it easy to find things. It acts as memory extender as envisioned by Vannevar Bush (Bush, 1945). Such power and agility does not come at no cost. Developers with their current educational background need additional training in the semantic technology Topic Maps and must overcome many hard learned but ultimately inappropriate paradigms that do not match the dynamic world that the 21st century is bringing upon us.

ACKNOWLEDGEMENTS

I want to thank Salvatore Vassallo, Markus Ueberall, Patrick Durusau, and Kambiz Darabi for their valueable feedback to draft versions of this paper.

REFERENCES

Babbage, C. (1835). *On the economy of machinery and manufactures*. Charles Knight, London.

Bush, V. (1945). As We May Think. *Atlantic Monthly*, 176(1):641–649.

Cerny, R. (2007). Topincs – A RESTful Web Service Interface for Topic Maps. In Maicher, L., Sigel, A., and Garshol, L., editors, *Leveraging the Semantics of Topic Maps*, volume 4438 of *Lecture Notes in Computer Science*, pages 175–183. Springer Berlin / Heidelberg.

Cerny, R. (2008a). Connecting Topincs – Using transclusion to connect proxy spaces. In Maicher, L. and Garshol, L., editors, *Subject-centric Computing*, volume XII of *Leipziger Beiträge zur Informatik*, pages 275–284.

Cerny, R. (2008b). Topincs Wiki – A Topic Maps Powered Wiki. In Maicher, L. and Garshol, L., editors, *Scaling Topic Maps*, volume 4999 of *Lecture Notes in Computer Science*, pages 57–65. Springer Berlin / Heidelberg.

Davies, S. (2011). Still building the memex. *Commun. ACM*, 54:80–88.

Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, Irvine.

Freeman, E. and Gelernter, D. (1996). Lifestreams: a storage model for personal data. *SIGMOD Rec.*, 25:80–86.

ISO/IEC 13250-2 (2006). *Information Technology – Document Description and Processing Languages – Topic Maps – Data Model*. ISO, Geneva, Switzerland.

ISO/IEC FCD 19756 (2009). *Information Technology – Document Description and Processing Languages – Topic Maps – Constraint Language, 2009-10-19*. ISO, Geneva, Switzerland.

Lansdale, M. (1988). The psychology of personal information management. *Applied Ergonomics*, 19(1):55 – 66.

Malone, T. W. (1983). How do people organize their desks?: Implications for the design of office information systems. *ACM Trans. Inf. Syst.*, 1:99–112.

Pawson, R. and Matthews, R. (2001). Naked objects: a technique for designing more expressive systems. *SIGPLAN Not.*, 36:61–67.

Rothenberg, J. (1995). Ensuring the Longevity of Digital Documents. *Scientific American*, 272:42–+.

Sigel, A. (2006). Report on the open space sessions. In Maicher, L. and Park, J., editors, *Charting the Topic Maps Research and Applications Landscape*, volume 3873 of *Lecture Notes in Computer Science*, pages 271–280. Springer Berlin / Heidelberg.

Stocker, A. and Tochtermann, K. (2009). Exploring the Value of Enterprise Wikis. In Liu, K., editor, *Proceedings of the International Conference on Knowledge Management and Information Sharing*, pages 5–12.