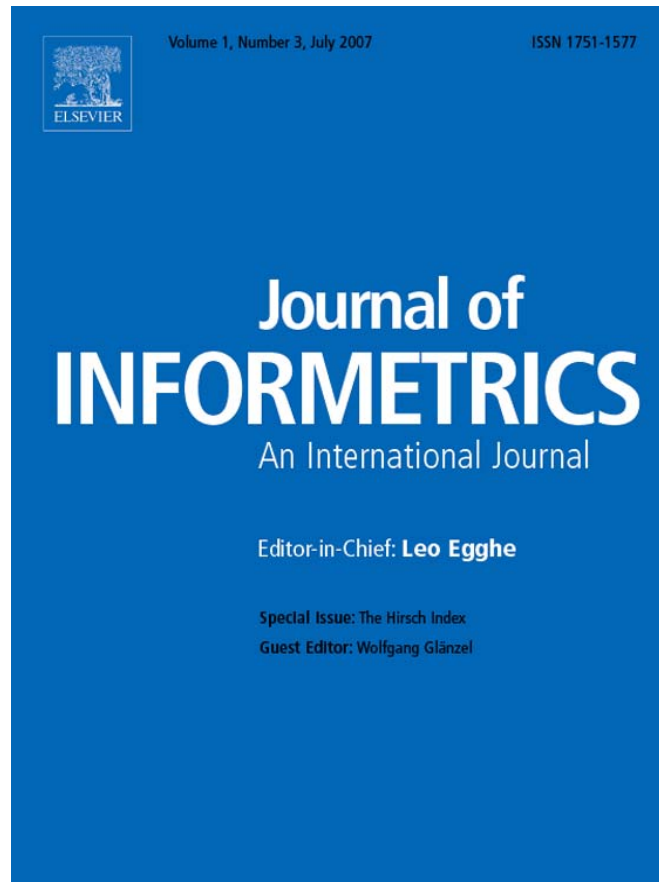


Provided for non-commercial research and education use.  
Not for reproduction, distribution or commercial use.



This article was published in an Elsevier journal. The attached copy is furnished to the author for non-commercial research and educational use, including for instruction at the author's institution, sharing with colleagues and providing to institution administration.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

<http://www.elsevier.com/copyright>



# Taxonomy visualization in support of the semi-automatic validation and optimization of organizational schemas

Katy Börner<sup>a,\*</sup>, Elisha Hardy<sup>a</sup>, Bruce Herr<sup>a</sup>, Todd Holloway<sup>b</sup>, W. Bradford Paley<sup>c</sup>

<sup>a</sup> *Indiana University, SLIS, 10th Street & Jordan Avenue, Wells Library, Bloomington, IN 47405, USA*

<sup>b</sup> *Indiana University, Computer Science Department, Bloomington, IN 47405, USA*

<sup>c</sup> *170 Claremont Avenue, Suite 6, New York, NY 10027, USA*

Received 27 July 2006; received in revised form 6 March 2007; accepted 6 March 2007

---

## Abstract

Never before in history has mankind produced and had access to so much data, information, knowledge, and expertise as today. To organize, access, and manage these valuable assets effectively, we use taxonomies, classification hierarchies, ontologies, controlled vocabularies, and other approaches. We create directory structures for our files. We use organizational hierarchies to structure our work environment. However, the design and continuous update of these organizational schemas with potentially thousands of class nodes organizing millions of entities is challenging for any human being.

The taxonomy visualization and validation (TV) tool introduced in this paper supports the semi-automatic validation and optimization of organizational schemas such as file directories, classification hierarchies, taxonomies, or other structures imposed on a data set for organization, access, and naming. By showing the “goodness of fit” for a schema and the potentially millions of entities it organizes, the TV tool eases the identification and reclassification of misclassified information entities, the identification of classes that grow too large, the evaluation of the size and homogeneity of existing classes, the examination of the “well-formedness” of an organizational schema, and more. As a demonstration, the TV tool is applied to display and examine the United States Patent and Trademark Office patent classification, which organizes more than three million patents into about 160,000 distinct patent classes. The paper concludes with a discussion and an outlook to future work.

© 2007 Elsevier Ltd. All rights reserved.

*Keywords:* Patents; Taxonomy; Ontology; Classification hierarchy; Visualization

---

## 1. Why and how the TV came into existence and related work—a foreword by Katy Börner

This foreword motivates the need for the TV, gives a timeline of the events that lead to its implementation, introduces major developers and their contributions, and discussed related work.

In October 2004, I attended three meetings with substantial discussions about the update and optimization of existing classification hierarchies and taxonomies. The first meeting was at the National Science Foundation (NSF) where my talk on Knowledge Domain Visualizations (Börner, Chen, & Boyack, 2003; Shiffrin & Börner, 2004) inspired a discussion about the goodness of fit among NSF research divisions and NSF research proposals and awards. The next day, I attended a panel meeting on the *Field of Science and Engineering Taxonomies*. The meeting was convened by

---

\* Corresponding author. Tel.: +1 812 855 3256.

E-mail address: [katy@indiana.edu](mailto:katy@indiana.edu) (K. Börner).

the National Science Foundation and organized by SRI International. It brought together a very interdisciplinary group of scholars and practitioners to brainstorm an evaluation and optimization of taxonomies in the field of science and engineering (S&E). These taxonomies are used to report research and development (R&D) results to Congress and also to decide about and communicate R&D budgets and spending. Several taxonomies had been last updated in the mid 1990s. Since then, many new research results had been published and new funding had been allocated. A manual update of the taxonomy seemed impossible due to the amount of data that needed to be incorporated. The third meeting was by the Association of Computing Machinery (ACM) Board and took place in New York City. Among others, an update of the ACM computing classification system was discussed. This hierarchy had been first published in 1964, been replaced by an entirely new system in 1982, and new versions of the 1982 system were published in 1983, 1987, 1991, and 1998, see <http://www.acm.org/class>. Since 1998, many more documents had been added to the ACM library. Yet, the manual update of the hierarchy seemed to be too daunting of a task. Interestingly, the 1998 version of the classification system is still in use today.

Taken together, there seemed to exist a need to evaluate the goodness of fit of an organizational schema (e.g., NSF directorate structure/S&E taxonomy/ACM classification system) and the data it organizes (e.g., NSF proposals and awards/research results and spending/documents in the ACM library). Based on the evaluation result, a librarian (or those in charge of updating the organizational schema) could then make informed decisions about where new data items should go, what classes need renaming, what new classes are required, and what major re-organizations of the schema make sense, among other things. The great need for ease of use and stability of the discussed organizational schemas means a complete or radical reorganization was expected to be counter-productive for their utility.

October 2004 was also a time when major software companies and search engine providers started to tell their customers that they could 'live in flatland'. They claimed that directory structures and meaningful file names were not needed any more, and that information could simply be found by entering a few search terms. I argue here that search engines are great for fact finding. However, they provide no 'up' button, no global view, and no structure to organize and make sense of knowledge, actions, and insights. The use of search engines can be compared to navigating the physical world by teleporting from one place (search result) to the next without ever climbing a tower or mountain and without ever seeing a map. As long as your tour guide (search engine) is knowledgeable, this works, but if you lose your guide, then you are lost. While there are many tasks supported by search engines, many tasks require a map of data, information, knowledge, and expertise for their completion. Identifying how knowledge interrelates and groups or existing trends and patterns in data are among tasks not well supported by search engines.

To address the need for semi-automatic validation and optimization of organizational schemas, I started to design the interface and basic architecture of a system—today called the taxonomy visualizer and validator (TV). Later, I met W. Bradford Paley, an interaction designer and artist from New York City who specializes in the design of readable, clear, and engaging representations of complex data. He carefully listened to my description of the needs and looked at my first sketches. Then, within a few hours, we jointly conceptualized the main parts of the TV interface. Many parts can be seen in Figs. 5 and 6. It was an unusual experience to merge my engineering brain with the brain of a professional designer and artist. The result was worth the struggle, and Brad Paley and my lab have since collaborated on several other projects.

Back in Bloomington, I wrote a detailed specification of the TV functionality. The static visualization functionality has been implemented and is detailed and shown in this paper. The dynamic visualization functionality and the semi-automatic validation and optimization functionality are under development and sketched in the future work section.

It took about 17 months to fully specify the TV, to implement and test first prototypes, to learn how to deal with millions of data objects, and determine how to render visualizations into files that can be printed in large formats at high resolution. Bruce Herr, programmer at the Cyberinfrastructure for Network Science Center at Indiana University, did the majority of the programming with input by Brad Paley and Shashikant Penumarthy. Todd Holloway, a computer science Ph.D. student at Indiana University, worked on the database backend and the data preparation. Elisha Hardy, undergraduate student and designer, Brad Paley, and myself worked on the layout and design.

Today, the image in Fig. 5 is part of the *Places & Spaces: Mapping Science* exhibit currently on display at the SIBL branch of the New York Public Library (NYPL). The image was also added to the map archive of the NYPL. It was because of this exhibit that the TV received my lab's high priority attention. It is my hope that this paper will create (financial) interest in the TV's dynamic visualization and semi-automatic validation and optimization functionality. The fully functional TV might become an invaluable tool for improving many of the organizational schemas we are using today.

There are many areas of research that aim to advance our understanding of concept formation and utilization in perception and categorization (Goldstone & Kersten, 2003) or improve ontology design, optimization, and usage for knowledge-sharing activities among men and machines (Berners-Lee, Hendler, & Lassila, 2001; Gruber, 1995). Work by Joseph D. Novak on concept maps (Novak, 1998) adds a visual means to communicate knowledge structures for sharing and consensus finding. Today, concept map-like displays are implemented as visual interfaces to diverse digital libraries (Börner & Chen, 2002; Doms & Schroeder, 2005; Shum, Motta, & Domingue, 2000). A long-term objective of this work is the design and utilization of libraries of reusable knowledge components and knowledge-based services that can be invoked over networks. To our knowledge, there exists no system that supports the visual examination of the “goodness of fit” of an organizational schema and the potentially millions of entities it organizes.

The subsequent sections are organized as follows: Section 2 introduces the TV functionality and the terminology used throughout the paper. Section 3 sketches a system architecture that supports the specified functionality. Section 4 details the TV interface. Section 5 demonstrates the TV using United States Patent and Trademark data. Sections 6 and 7 conclude the paper with a discussion of results and an outlook to future work.

Note that Sections 2–4 and 6 explain the full functionality of the TV while Section 5 shows the static interface part of the TV.

## 2. TV functionality and used terminology

This section details the functionality of the TV on the basis of the wish lists collected during the three meetings mentioned in Section 1. In order to define the TV functionality in detail we will use the following terminology:

- ‘Entity type’ refers to the type of an entity, e.g., paper, author, patent, grant, email, image.
- ‘Entity’ refers to a specific instantiation of an entity type, e.g., a specific paper or author.
- ‘Links’ refer to connections among entities.
- ‘Link type’ refers to the type of a link, e.g., identical, similar-based-on-x, paper-citation, co-author. A data set can have multiple link types.
- ‘Organizational schema’ refers to a tree structure imposed on a set of entities for organization and structure. Examples are classification or file hierarchies, taxonomies, and ontologies.
- ‘Organizational label’ refers to a textual means to label a set of entities. Examples are class or category names, taxonomy or MeSH terms.
- ‘Organizational node’ refers to a node in the organizational schema. Examples are classes or categories.
- ‘Size’ refers to the number of information entities in one organizational node.
- ‘Similarity’ indicates how much a set of entities have in common. The similarity of an entity set is typically computed by means of a similarity measure. It can also be specified a-priori.

Obviously, there exists an interesting interplay between the structure of the organizational schema and the set of entities it organizes: The organizational schema strongly depends on the set of entities it organizes and the organization of the entities depends on the structure of the organizational schema. Yet, it is beneficial to distinguish functionality that is mostly related to the optimization of the organizational schema and functionality mostly related to the best possible organization of entities. Subsequently, we list the properties of an ideal organizational schema.

### I. Ideally, the organizational schema

- I.1 Is a balanced tree, in which the main branches have approximately the same depth and approximately the same number of subtrees or leaf nodes, to support efficient search.
- I.2 Is evenly used, i.e., there is an equal number of entities in each organizational node.
- I.3 Organizes the entities so similarity is high within organizational nodes and low between organizational nodes.

These ideals are only obtainable to a certain degree. In most cases, the organizational schema and the entity set are not static. Often, a growing stream of new entities needs to be sorted into existing organizational nodes and the organizational schema needs to be continuously modified to best fit old and new entities. Secondly, the organizational schema needs to be changed gradually as it is the only means for people to make sense of a potentially very large set of entities. Replacing an existing organizational schema by a completely new one is just as problematic in information

spaces as in supermarkets. It is possible that the perfect classification of one new entity would require a complete reorganization of the schema to fulfill all other criteria. However, re-organizing an existing schema whenever a new product or entity comes in, possibly several times per day, would considerably lower the value of the schema. The troubles caused by re-organization need to be weighed against the troubles of an imperfect schema.

Given the need for continuous, gradual optimization of organizational hierarchies, the TV needs to support the examination of time-based variables such as

- T.1 Growth of the organizational schema including what nodes are new, which have been renamed, etc.
- T.2 Growth of the size of organizational nodes over time.
- T.3 Changes in the similarity of entities that are classified into the same organizational node over time.

There also is a need to see a major part of the organizational schema and all the entities it organizes at once. For example, librarians wanted to see all ACM classes that contain papers published in journal x, all people attending conference y, all graduate students in computer science. They also wanted to know if entities in related organizational nodes are interlinked more often, e.g., papers cite each other more often or authors co-author more often.

Last but not least, there was a need for the

#### A.1 Automatic reorganization of subtrees in the organizational schema.

For example, a user might identify an organizational node with diverse information entities and request a re-organization of this node and its child nodes using a certain similarity measure and clustering algorithm, e.g., she may like to request that all papers that highly cite each other or papers that share many words are grouped together.

### 3. TV interface

The TV interface needs to support the functionality identified in Section 2. It needs to be easy to learn, communicate information effectively, and be aesthetically pleasing. It should optimally split work among human users, who are powerful visual processors and can judge the quality of and name entity groupings, and computers, which are able to analyze and visualize very large amounts of data.

#### 3.1. Major interface parts

Given the requirement specification in Section 2, the TV interface needs to provide a means to examine and evaluate:

- The current organizational schema (to check I.1).
- Changes in the organizational schema (to check T.1).
- Organizational nodes and the entities they contain (to check I.2 and partially I.3).
- The similarity of entities belonging to one organizational node (to check I.3).
- Entity links (to check I.2).
- Number of new items sorted by time (to check T.2).
- Entity attributes, e.g., similarity, sorted by time (to check T.3).
- Entity search (for new entities to check T2 and T3 and for entities with certain properties to check I.3) and to retrieve more details on demand.

Subsequently, we describe the visual rendering of these interface parts.

#### 3.2. Organizational schema

Conceptually, the organizational schema is the main reference system that organizes all entities. Given its function as a frame of reference, it is visually rendered as a base map. All other information is laid out using this base map. Nodes in an organizational schema commonly have an organizational label one or more words long. People need to be able to read these words to understand and navigate this abstract information space. Hence, a layout needs to be

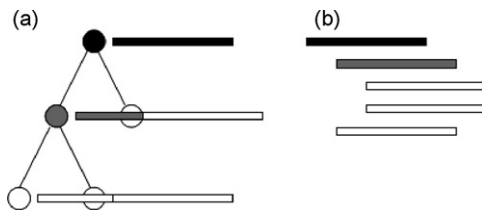


Fig. 1. (a) Tree structure and (b) indented list representation of an organizational schema.

found that supports the ordered display of as many words as possible. An organizational schema could be rendered as a tree (cf. Fig. 1a) or as an indented list (cf. Fig. 1b). The latter is analogous to a table of contents or a file directory structure, where node depth in the hierarchy is indicated by indentation level. In Fig. 1, circles represent organizational nodes, rectangles organizational labels. Black filled circles and rectangles indicate the root node. Gray and white nodes denote intermediate and leaf nodes, respectively. Both representations quickly reveal if a schema is well-formed (I.1). However, the labels in the tree representation occlude each other—particularly if many nodes share the same level. Node labels are easy to read in the indented list representation. Hence, it is beneficial to use the indented list representation for schemas with many nodes.

Color coding can be applied to visualize changes in the organizational schema (T.1). Let us assume the black node in Fig. 1 came into existence first, then the gray node was added, then the white nodes. In this case, the color coding also reflects the age of the nodes. Different colors can be employed to differentiate node renaming from node insertion and deletion.

### 3.3. Organizational nodes and entity attributes

The size of an organizational node, i.e., the number of entities it contains, can be visualized using a bar graph in which each bar represents exactly one entity, see Fig. 2. Entities can be counted non-recursively (cf. Fig. 2b) or recursively (cf. Fig. 2c).

The height and color of bars can be used to depict attribute values of entities. For example, the distance of an entity from the mean of the other entities in an organizational node can be expressed by the height of the bar—in an analogy of nails that stick out and simply do not fit. Color can be used to highlight entities that match search queries, such as all entities published in 2004 or in the last month or that contain a certain word in the title or have a certain author.

The bar graphs can be sorted by time, helping indicate if entity similarity increases or decreases over time (T.3). They can be sorted by similarity or any other attribute value to gain a quick overview of the attribute distribution.

If a large number of entities needs to be encoded, the width of a bar can be varied to represent multiple information entities. For example, bars that represent 10 information entities might be twice as wide as bars that represent one information entity. Bars that represent 100 information entities might be three times as wide as bars that represent one information entity, etc. Entities that match a certain search query can be color coded as well. Examples are given in Fig. 3.

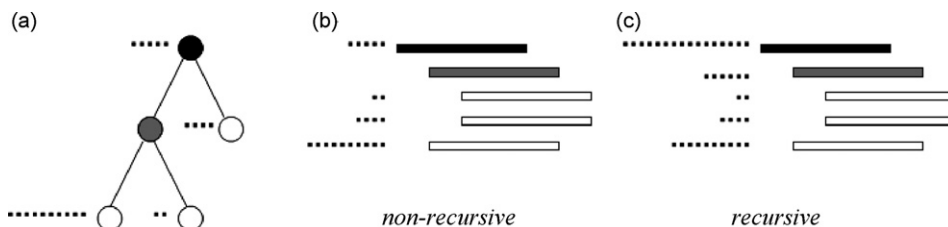


Fig. 2. (a) Tree structure and (b and c) indented list representation of an organizational schema. See Fig. 1 for shape and color coding. Dots to the left of organizational nodes denote the number of entities they contain. (b) Lists the entities in each node exclusively. (c) Recursively counts the number of entities under a certain node, i.e., the root node contains all entities.

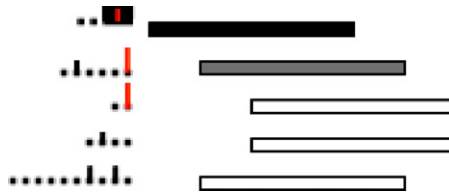


Fig. 3. Bar graph height and color (red) coding as well as examples of bar graph aggregation. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of the article.)

### 3.4. Line overlays

Line overlays can be used to indicate citation, co-author, class-inheritance or any other linkages among entities. Lines can interconnect the bars that represent certain entities or interconnect the organizational nodes that contain related entities. Text occlusion by links needs to be minimized. Link direction can be indicated by color coding, e.g., by drawing the beginning of a link in a darker and the end in a lighter color. Slightly random or attribute based (e.g., time based, see Fig. 5) color assignments also help to distinguish different links.

### 3.5. Interaction design

The display of an organizational schema with 100,000 nodes using 6pt type font and 1pt line spacing, i.e., 7pt or 4 mm space per line, results in a list of 400,000 mm or 400 m length—too long to make sense of or manage. Hence, interactive manipulation becomes extremely important. In particular, it appears to be desirable to facilitate the subsequent activities:

- Parts of the organizational schema can be collapsed and expanded as needed.
- Alternative organizational labels can be selected.
- Bar graphs can be sorted according to different entity attributes.
- Search queries can be run and matching entities highlighted.
- Detailed information on selected entities can be retrieved.

### 3.6. Animation design

To address the needs T1–T3 identified in Section 2, the TV needs to support an animation of the:

- Evolving organizational schema, by renaming organizational labels and adding, moving, and deleting organizational nodes.
- The growth of entities at each organizational node, including the growth of bar graphs and their properties as well as the re-organization of entities.
- Line overlays, such as evolving citation linkages or co-authorship relations.

The animation needs to be controllable in speed and direction (forward and backward) to examine specific changes in detail.

## 4. General system architecture

The taxonomy visualization and validation tool currently runs as a stand-alone tool using a precompiled, static dataset. In the near future it will become available as a Web service and be able to process streaming data, see Section 7.

The general system architecture is shown in Fig. 4. It consists of four major components: An engine responsible for maintaining communication between the other TV components, a PostgreSQL database, a visualization component, and the user interface. All four parts are explained below.

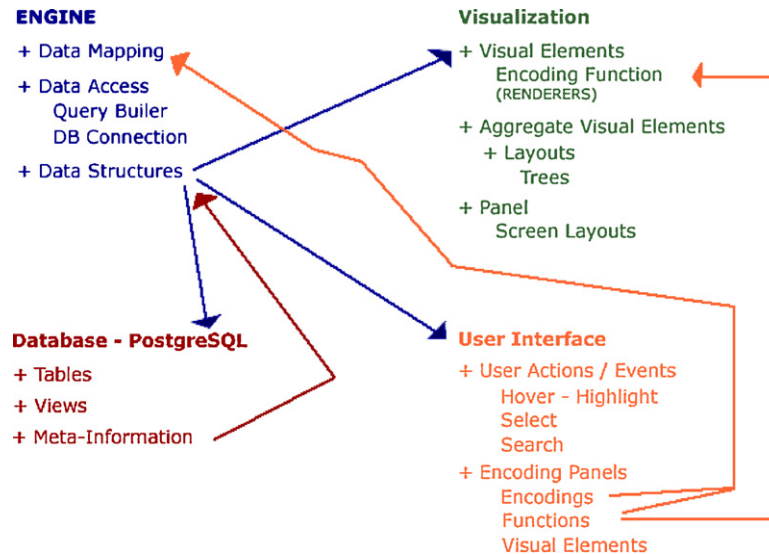


Fig. 4. Components of the TV system architecture and their major interactions.

#### 4.1. Engine

The engine is the heart of the TV architecture, because it organizes and maintains the communication between all components. The engine is responsible for establishing database connections, handling SQL queries, resolving data type issues related to data mapping, and listening to events from other components and visual elements. It also maintains an internal data structure that monitors the current TV status and changes based on user actions. The engine also performs memory management, controlling the internal representation of the data, its temporary database persistence, and the 'intelligent' creation of indexes for speed-up. Memory management is a large part of the TV by virtue of the size of the datasets, complexity of 'goodness of fit' measures, and number of elements in the visualization.

#### 4.2. PostgreSQL database

Input data comprises an organizational schema and a set of information entities with assigned organizational labels. The computation of the fit of entities into an organizational node and the automatic restructuring of organizational schemas require a means to identify the similarity of information entities.

A PostgreSQL database is used with a generic database schema to store these data types in multiple tables. Parent-child information from the organizational schema is stored in one of the tables with child node entries being unique. Another table stores labels and levels information related to each child node. A third table is used to store labels of the organizational hierarchy nodes. Two more tables capture information on entities associated with organizational nodes and different variables of entities. These tables are useful when it comes to querying and classifying entities.

#### 4.3. Visualization

The visual interface was implemented using Java Swing. The panel provides the drawing area upon which the visual elements are rendered. The layout component of the visualization determines the location of layout for the visual component. Another component of visualization called the 'Renderer' is responsible for visual encoding of the visual component based on the underlying data.

The taxonomy visualization can also be rendered into a postscript file supporting truly global views in high resolution and on large sheets of paper. Printing into postscript takes as input the hierarchy and entity data to be displayed as well as configuration information, such as color, size, and font selections. It then re-computes the layout and renders the result into a file.



#### 4.4. User interface

The user interface component handles all user-based queries. It captures user actions, sends them to the engine and visualization components and displays the result of the interactions via changes in the visual display.

All data preprocessing, analysis and visualization algorithms are implemented as plug-ins. This eases the combination, utilization, and comparison of algorithms and continuous improvement of TV functionality.

### 5. Visualizing the united states patent and trademark hierarchy

The TV was applied to visualize the United States Patent and Trademark Office (USPTO) patent classification, which organizes about 3.2 Million patents into about 160,000 distinct patent classes. Interestingly, the USPTO classification system is changed on a continuous basis, see <http://www.uspto.gov/go/classification/> for the current version. The XML-encoded data dumps downloadable from <ftp://ftp.uspto.gov/pub/patdata/> as well as original (old) green book data files available for the same from the USPTO may or may not have the correct current classifications. To ensure that the classifications assigned to patents match the currently used patent classification, the current patent classification hierarchy as well as the current classifications of all 3 Mio patents were downloaded from the USPTO site.

Our original plan was to print the complete hierarchy-all 160,000 classes organized in an organizational schema that is up to 15 levels deep. However, a quick calculation let us realize that this would require much more space than we had available-even if a very small font was used and partial over plotting of category label names was employed. We therefore decided to plot only the first three levels of the hierarchy using a very small font.

Specifically, 7pt is used for level 1, 3.5pt indented by 1.5pt for level 2, and 1pt indented by 3pt for level 3. It still took 25 columns to render those 51,391 categories. The result is the fabric-like pattern shown in the middle of Fig. 5. The area can be seen as a 1.5 dimensional reference system that captures the main structure of this complex information space.

The reference system was used to depict the impact (Fig. 5, left) and prior art (Fig. 5, right) of two patents. The patent on Gore-Tex<sup>®</sup> – the lightweight, durable synthetic fiber used as a tissue filler in cosmetic implants, waterproof clothing, and many other products – was selected to show the impact a patent might have. The Gold Nanoshell patent was exemplarily selected to show the prior art of a patent. Gold Nanoshells are a new type of optically tunable nanoparticles. Their ability to “tune” to a desired wavelength is critical to in vivo therapeutic applications such as thermal tumor destruction, wound closure, tissue repair, or disease diagnose. The cover pages of both patents and their position in the 25 column classification hierarchy are shown. Line overlays represent citation linkages. Red lines denote 182 citations to the Gore-Tex<sup>®</sup> patent. They are sorted in time with dark red indicating older and bright red younger citations. Blue lines represent the 16 prior art references of the Gold Nanoshell patent to the classes of the cited patents.

Fig. 6 shows a zoomed in version of the Gore-Tex<sup>®</sup> patent. The class of the patent is highlighted in brown. The bar that represents the patent is circled and linked to the cover page of the patent via a brown line. The bar graphs next to each class indicate how many patents are in the class. Each bar represents one patent. To determine how similar a patent is to all other patents in its class, we examine the number of references it shares using data downloaded from <http://patft.uspto.gov/netahtml/PTO/search-bool.html>.

To avoid a potentially very large number of pair-wise comparisons due to computing the similarity of each patent to all other patents, we compute a patent prototype for each class  $c$  and determine the similarity of each patent to this prototype. The prototype of a class is defined to be a feature vector  $PT_c$  of length equal to the number of unique references occurring across all patents in the class. Each feature corresponds to the prior probability of a reference occurring in that class, i.e., the number of patents that have this reference divided by the total number of patents in this class. A patent is represented as a feature vector  $P$  of the same length as the prototype of its class. It is given binary features indicating the presence or absence of a reference.

For each patent feature vector, the similarity between it and its class prototype were computed using the cosine similarity measure which equals the dot product of the two vectors divided by the product of the magnitudes of the two vectors, i.e.,

$$\text{sim}(P, PT_c) = \frac{P \bullet PT_c}{|P||PT_c|}. \quad (1)$$

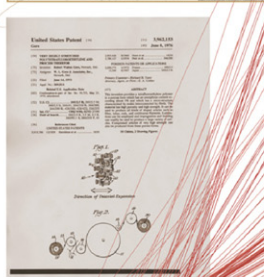
The resulting similarity value was then used to indicate the ‘goodness of fit’ of a patent and its class.

## Impact

The United States Patent and Trademark Office does scientists and industry a great service by granting patents to protect inventions. Inventions are categorized in a taxonomy that groups patents by industry or use, proximate function, effect or product, and structure. At the time of this writing there are 163,521 categories in a hierarchy that can get as deep as 15 levels. We display the first three levels (13,529 categories) at right in what might be considered a virtual map of inventions.

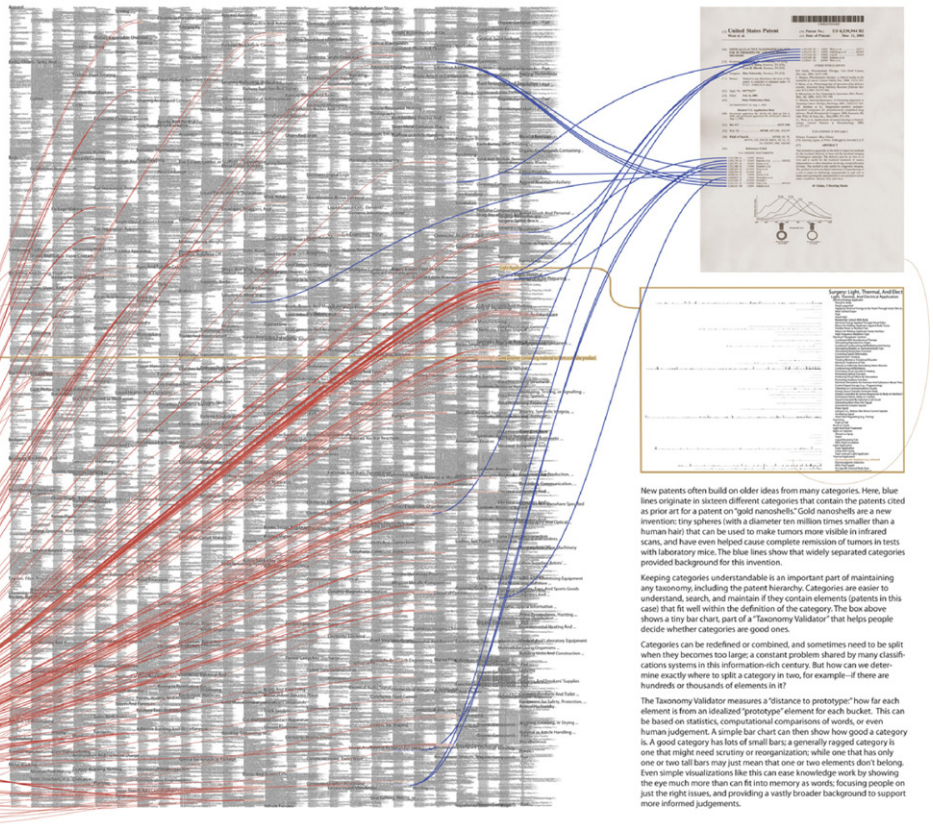
Patent applications are required to be unique and non-obvious, partially by revealing any previous patents that might be similar in nature or provide a foundation for the current invention. In this way we can trace the impact of a single patent, seeing how many patents and categories it affects.

The patent on GoreTex—a lightweight, durable synthetic fiber—is an example of one that has had significant impact. The box below enlarges the section of the hierarchy where it is filed, and the red lines (arranged to start along a time line from 1981 to 2006) point to the 130 categories that contain 182 patents, from waterproof clothing to surgical cosmetic implants, that mention GoreTex as prior art.



## US Patent Hierarchy

## Prior Art



New patents often build on older ideas from many categories. Here, blue lines originate in sixteen different categories that contain the patents cited as prior art for a patent on "gold nanoshells." Gold nanoshells are a new invention: tiny spheres (with a diameter ten million times smaller than a human hair) that can be used to make tumors more visible in infrared scans, and have even helped cause complete remission of tumors in tests with laboratory mice. The blue lines show that widely separated categories provided background for this invention.

Keeping categories understandable is an important part of maintaining any taxonomy, including the patent hierarchy. Categories are easier to understand, search, and maintain if they contain elements (patents in this case) that fit well within the definition of the category. The box above shows a tiny bar chart, part of a "Taxonomy Validator" that helps people decide whether categories are good ones.

Categories can be redefined or combined, and sometimes need to be split when they become too large, a constant problem shared by many classifications systems in this information-rich century. But how can we determine exactly where to split a category (in two, for example—if there are hundreds or thousands of elements in it)?

The Taxonomy Validator measures a "distance to prototype" how far each element is from an idealized "prototype" element for each bucket. This can be based on statistics, computational comparisons of words, or even human judgement. A simple bar chart can then show how good a category is. A good category has lots of small bars; a generally ragged category is one that might need scrutiny or reorganization; while one that has only one or two tall bars may just mean that one or two elements don't belong. Even simple visualizations like this can ease knowledge work by showing the eye much more than can fit into memory as words; focusing people on just the right issues, and providing a vastly broader background to support more informed judgements.

Fig. 5. Taxonomy visualization of patent data.

We wanted outliers to be easily visible—to stick out of the crowd. Hence, similarity was converted into distance and patents that did not fit well in a class, i.e., had a large distance to their prototype, were represented by a high bar. Patents with a small distance got a low bar. Final bar heights were linearly scaled to fit between zero and 4 mm.

As can be seen in Fig. 6, most patents fit very well into their class (according to the selected similarity measure). Very few patents do not fit well and should be examined in more detail. The 25 column rendering of the hierarchy can also be used as a reference system over which, e.g., citation patterns can be overlaid.

To create the 24 in. × 30 in., high resolution version of the image shown in Fig. 5, several layers were generated and composed. The TV tool was used to generate the taxonomy substrate, two zoomed substrate sections, and the line overlays. These layers were outputted as postscript and merged in Adobe Photoshop® along with informational graphics like text, some minor connecting lines, and the two patent front page images. Note that the rendering code of the TV tool was customized for the data, the placement of the patents, and the dimensions of the canvas in order to get a cohesive final product.

## 6. Discussion

This paper motivated and explained the taxonomy visualization and validation tool. The tool helps combine the expertise of human specialists with automatic data analysis and visual rendering. It requires the existence of an organizational schema and a set of information entities that are classified into this schema. A similarity measure is needed to compute the fit of entities into an organizational node. Some of the TV analysis, dis-

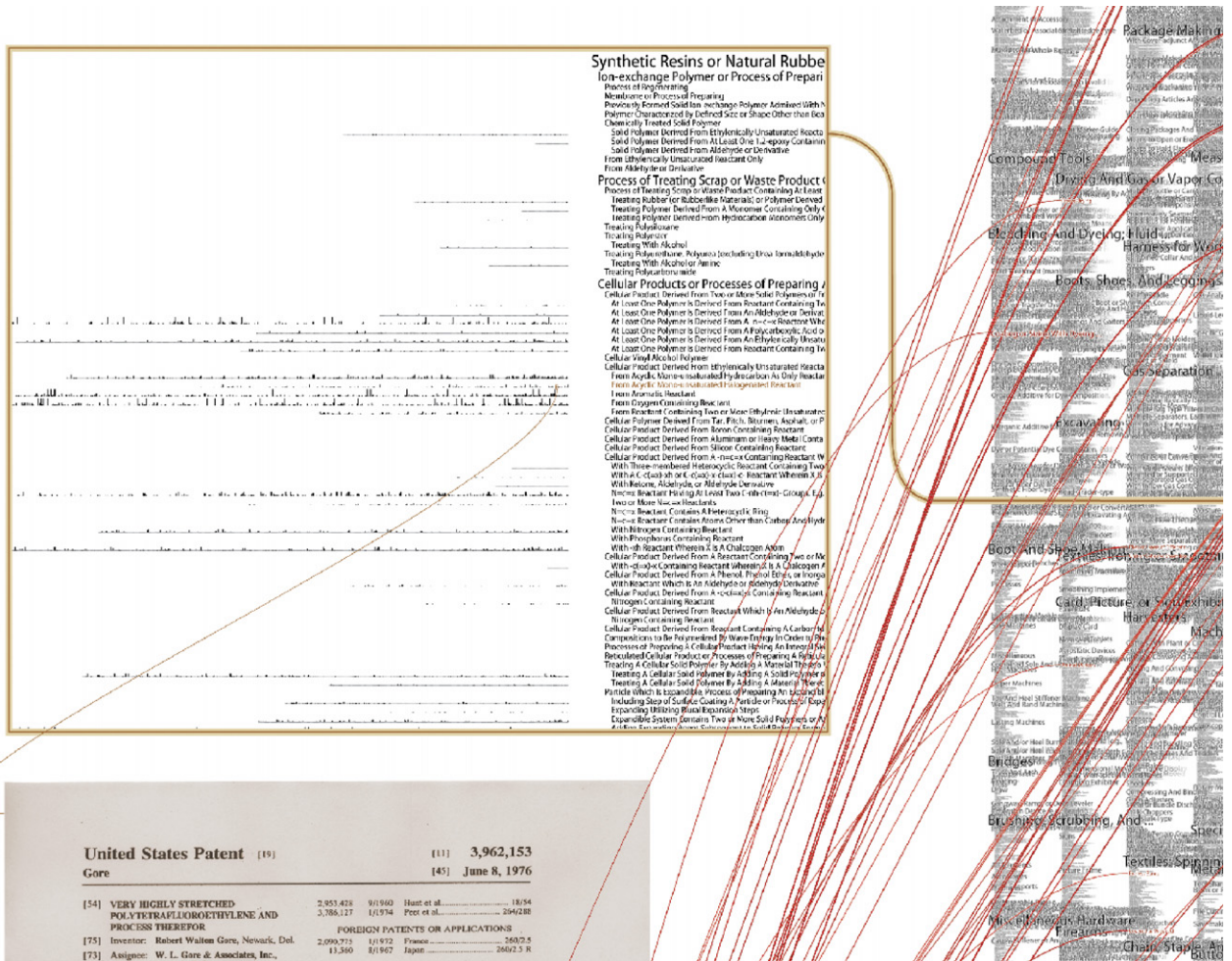


Fig. 6. Zoom into the taxonomy visualization given in Fig. 5. Shown is a close-up of the patent classification environment of the patent on Gore-Tex®.

play, and interaction techniques are newly developed; others were combined in an unusual way. The TV tool is unique in:

- Its usage of bar graphs to display properties of organizational nodes, e.g., size, and entities, e.g., similarity, age.
- Its usage of a static (yet interactively navigable) ‘substrate map’ of the organizational schema and dynamically changing ‘bar graphs’ and ‘line overlays’.
- Its usage of organizational nodes to apply a divide and conquer strategy during the analysis and visualization of potentially very large-scale data sets.

6.1. Applicability

The TV might be applicable to help examine, validate, and optimize organizational schemas as diverse as: the ACM computing classification system, S&E taxonomies, patent classification hierarchy, MeSH Controlled Vocabulary Thesaurus, Google’s categories at <http://www.google.com/dirhp>, file directories, the Yellow page directory of businesses, and many others.

## 6.2. Customizability

Each new data set and user group will require a customized user interface that matches existing conceptualizations and information needs. The visual appearance of the TV interface will have to be customized to the specific datasets and user tasks. Ideally, the TV interface will reflect the business practices librarians and other decision makers have worked with for years and spent decades mastering.

The use of the CIShell software framework discussed in Section 4 and the ‘interface configuration’ detailed in Section 7 support easy and fast customizability.

## 6.3. Scalability

The TV has been used to render 160,000 organizational nodes at once. The number of nodes and entities that can be rendered is only limited by the amount of memory available. Computing ‘goodness of fit’ for a large dataset is very computation and memory intensive, but can be done offline in advance and in a parallel fashion.

Note that the automatic reclassification is applied to organizational nodes (excluding the root node) only. This corresponds to a ‘divide and conquer’ strategy for the examination of the homogeneity of entities in a node and the re-organization of parts of the organizational schema.

## 6.4. Open questions

As the TV is applied to help organize diverse datasets new questions arise. Among them are: What information should be encoded in what way? For example, the age of an entity can be encoded via the color of bar graphs or can be actively queried for via search. The identity of two data files stored in different directories can be visually depicted by coloring their bars identically or by inter-linking their bars. Also, what is the ‘optimal’ data density? How many nodes and bar graphs should be shown to support efficient work? What similarity measures are best to compute the goodness of fit? How does one display and interact with potentially very large hierarchies on a monitor screen with a very limited number of pixels?

## 7. Future work

This section discusses planned work that aims to extend the current TV implementation so that it supports the functionality detailed in Section 2.

### 7.1. Web services

In many cases, clients might like to run the TV as a web service. This way, they login to an online site, select the organizational schema they would like to work with, and start the validation and optimization process. The CIShell discussed in (Herr, Huang, Penumarthy, & Börner, 2007) is a plug-and-play architecture that supports the plug and play of different datasets and algorithms. Using CIShell as the TV core supports its deployment as a Web service, stand alone tool, or peer-to-peer application.

### 7.2. Handling streaming data

Most datasets evolve dynamically over time. The easier and faster new data entities can be incorporated and the organizational schema can be adapted to different user needs the more valuable the TV becomes.

### 7.3. Interface configuration

To ease the adaptation of the interface appearance and functionality and to serve different datasets to different user groups there needs to be a way to quickly configure the general layout of an organizational schema (e.g., what subset of the hierarchy is shown, in how many columns and with what type font, font size, indenting and on what background), the layout and encoding of bar graphs (e.g., sorted by time, in a certain color, and with or without (non)recursive

aggregation), lines (e.g., what do the lines represents and in what color and thickness are they drawn), and interactivity elements (e.g., search field, means to zoom, pan, and request details).

#### 7.4. Semi-automatic optimization of the organizational hierarchy

A user should be able to select any part of the organizational schema that has an organizational label and request an automatic reorganization. They will need to specify a similarity measure and clustering algorithm, e.g., entities that share words are assumed to be similar, apply  $k$ -means clustering with a given  $k$ . Each of the resulting  $k$  cluster nodes will contain entities that share many words. Users can then assign organizational labels to those organizational nodes. Users might like to test and compare different similarity measures and clustering approaches to find a combination that best matches their intuition of a good data organization.

#### 7.5. User management

To restrict access rights and to keep a record of who made what changes and when, user access and control management similar to concurrent version control (cvs) is needed. All user interaction is stored in a log file as a personal and corporate record. The user can also leave comments about major restructuring, interesting observations, etc. that are saved into the log file. Based on these user logs, the evolution of the organizational schema can be recorded and visualized over time. All actions of a specific user, user group or all users can be analyzed and replayed.

### Acknowledgements

We would like to thank Josh Bonner and Alaa Elie Abi Haidar for programming initial TV prototypes, Shashikant Penumarthy for his expert advice regarding the specification and implementation of the current system, and Eric Giannella for his guidance in the selection of patent examples. We appreciate the insightful comments by Kevin W. Boyack, Russell Duhon, and the anonymous reviewers on an earlier version of this paper.

This research is supported by the National Science Foundation under IIS-0513650, IIS-0534909, and a CAREER Grant IIS-0238261 as well as by a James S. McDonnell Foundation grant in the area Studying Complex Systems. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

### References

- Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The semantic web. *Scientific American*, 284(5), 34–43.
- Börner, K., & Chen, C. (Eds.). (2002). *Visual Interfaces to Digital Libraries*. Springer Verlag, < <http://link.springer-ny.com/link/service/series/0558/tocs/t2539.htm>, >
- Börner, K., Chen, C., & Boyack, K. (2003). Visualizing knowledge domains. In B. Cronin (Ed.), *Annual Review of Information Science & Technology* (pp. 179–255). Medford, NJ: Information Today, Inc./American Society for Information Science and Technology.
- Doms, A., & Schroeder, M. (2005). GoPubMed: exploring pubmed with the gene ontology. *Nucleic Acids Research*, 33(Web Server Issue), W783–W786.
- Goldstone, R. L., & Kersten, A. (2003). Concepts and categories. In A. F. Healy & R. W. Proctor (Eds.), *Comprehensive Handbook of Psychology* (pp. 591–621). New York: Wiley.
- Gruber, T. (1995). Toward principles for the design of ontologies used for knowledge sharing. *International Journal Human-Computer Studies*, 43(5–6), 907–928.
- Herr, B. W., Huang, W., Penumarthy, S., & Börner, K. (2007). Designing highly flexible and usable cyberinfrastructures for convergence. In W. S. Bainbridge & M. C. Roco (Eds.), *Progress in Convergence—Technologies for Human Wellbeing* (pp. 161–179). Boston, MA: Annals of the New York Academy of Sciences.
- Novak, J. D. (1998). *Learning, Creating, and Using Knowledge: Concept Maps as Facilitative Tools in Schools and Cooperations*. Lawrence Erlbaum Associates.
- Shiffrin, R. M., & Börner, K. (Eds.). (2004). Mapping knowledge domains. *Proceedings of the National Academy of Sciences*, 101 (Suppl. 1), <<http://ella.slis.indiana.edu/%7EKaty/paper/04pnas-intro.pdf>>.
- Shum, S. B., Motta, E., & Domingue, J. (2000). ScholOnto: an ontology-based digital library server for research documents and discourse. *International Journal on Digital Libraries*, 3(3), 237–248.