University of Alberta

OF WIKIS AND BLOGS: UNDERSTANDING AND SUPPORTING ON-LINE COMMUNITIES OF PRACTICE

by

Cleo Espiritu ©

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science

Edmonton, Alberta
Fall 2006

Your file Votre référence
ISBN: 978-0-494-22259-1
Our file Notre référence
ISBN: 978-0-494-22259-1

Abstract

Maintaining communities of practice (CoP) has been eased by web-based tools, which allow members to participate and contribute to the community anytime, anywhere. Tools such as wikis and blogs enable people to publish content with ease so they can easily share knowledge and ideas. Giving users the freedom to publish will increase the amount of content available; however, presenting a large amount of content using hypertext media can overwhelm readers, as most hypertext media lack an intuitive navigational structure that effectively guides users through the content. In this thesis we developed a family of components that analyze wikis and blogs, infer semantic structures from their content and interactively visualize these structures to improve the users' access and understanding of their contents. Our experiments with several systems built with these components indicate that, indeed, such semantic support and visualizations help users make better use of wikis and blogs and more easily perceive the knowledge communicated by their content.

Acknowledgement

I would like to thank: Dr. Eleni Stroulia, for her continual support and guidance ever since I was a high school intern for the WISEST summer program of 1999; Tapanee Tirapat, for her work, insights and ideas for our projects; Michelle Annett, for her development of AnnokiBlooms; Megan Chartrand and Janice Annett, our two WISEST interns, for their contributions to ENWiC and AnnokiBlooms, respectively; and to all my supervisors and colleagues from projects and jobs outside of this thesis, for giving me valuable working experiences. Finally, I would like to thank my family, for always being there.

The work of this thesis was funded by a NSERC CGS-M, an iCore Graduate Student Fellowship, and a Walter H Johns Graduate Fellowship.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

1.1 Motivation and Background

The World-Wide-Web has become an indispensable tool for collaboration and information dissemination. People now have the ability to access anything, anytime, from any computer connected to the Internet. A phenomenon of particular interest is the formation of online communities of people who use web-based systems to communicate with family, friends and people with similar interests, across geographical and time-zone boundaries.

The unprecedented accessibility and convenience of accessing the web has given rise to the seamless integration of the web in a multitude of daily activities, in a way that enhances their quality through the interaction and collaboration with other people with relevant experience and knowledge. People use the web to plan their travels by comparing prices, getting reviews of hotels and resorts and getting advice on what to do in their destination. They read on-line reviews about movies, books and products before making a purchase. They get informed about politics and recent events. Moreover, they become lifelong learners; web resources are regularly used to enhance traditional classroom delivery and distance-learning programs, relying in a substantial way on the web, have recently mushroomed for all education phases, from primary school to University.

The Web-based systems that saw the most dramatic increase in usage in recent years are wikis and blogs. The most popular and comprehensive wiki online today is Wikipedia [67], which acts as an open encyclopedia for any subject. As of August 2006, it contains over 1.3 million English articles. As well, there are many individual wikis on the web that focus on specific topic(s), such as WikiTravel [68], LinuxWiki [51], and Javapedia [50]. Such vast information repositories are invaluable for learners, and they are regularly utilized by students in their learning process. Wikis can also be used as a group project management tool; they are especially useful in the intranets of companies and institutions, where members can use them as means of communicating knowledge, ideas and plans to other members. The most appealing benefit of using a wiki is that it provides a simple interface for content publishing, which makes it easy for anyone to learn to use. Furthermore, since everyone who has access to the wiki can contribute and edit content, users are collaboratively enhancing the content on the wiki continuously.

Blogs are also simple publishing platforms that allow users to express their comments and knowledge. Blogs tend to be more personal and thus their content is usually more opinionated. The domain where blogs can really shine in is market research. Blogs can be seen as a vast collection of consumer opinions, and analyzing this collection should give companies an idea of what types of product and feature are currently popular with the customers.

The ease of publishing made possible with wikis and blogs does come with a drawback – readers may end up feeling 'overloaded' with too much information. While having a large amount of data is good for research purposes, navigating and reading through so much data may overwhelm a casual user. To make matters worse, most conventional wikis and blogs do not provide much support for intuitive navigation. Users usually need to read through long lists of text links to locate the desired articles or entries, and this can be a daunting task that degrades user satisfaction. It is also easy for users to feel lost after clicking through multiple links. Users may also likely miss relationships between related articles/entries, as conventional interfaces do not make these associations apparent to the users, and all users really see are discrete, single pages. For education purposes, relationships can be valuable to a student's understanding of the domain, and thus a good interface should highlight these relationships so they would not be as easily missed by users.

Another problem with systems that allow free publishing is that it is hard to guarantee that the content on these systems is credible. Since anyone can contribute content, erroneous or inappropriate information may be submitted, and readers may be unable to distinguish it from correct information. Wikis are especially vulnerable to this problem, especially when they reside in the public domain. Wikipedia [67] has been criticized many times for this problem; one of the most notable and published error is the false biography on journalist John Seigenthaler [30]. This lack of trust to Wikipedia is also evidenced by the fact that most academic institutions disapprove students citing it as a reference source for reports and assignments [13][27]. Blogs are susceptible to the same problem, but readers tend to treat most blog content as personal opinions instead of facts and thus the "trustworthiness" concern is not equally strong. Readers also seem to eventually learn to trust information only from reputable blogs of credible bloggers.

1.2 The Research Problem

The overall objective of this thesis is to *enhance the usability of wikis and blogs* by enabling users to more easily review their content and access information of their interest. The work of this thesis focused on designing and implementing components that can extend the basic wiki/blog functionalities, so that semantic information can be naturally weaved in the published content, overlaying it with a more intuitive structure. Furthermore, the users' access of this content – especially in learning scenarios – is further supported through an intuitive, interactive, visual interface that encourages the users' reflection on the content presented.

1.3 Methodology

The two specific hypotheses that this thesis examines are:

1) The use of a semantic structure to represent hypertext content can assist users in effectively identifying information of interest; and
2) Visualizing a semantic structure helps users understand the content in the domain better, as supported by researches on the benefits of visual learning [1][28][10][8].

Our research methodology for investigating our hypotheses is outlined in the following steps.

1) We first reviewed current hypertext technologies to select the types of hypertext documents that contain useful information, but are in need of a better organization structure.
2) We then reviewed current semantic web technologies to select the type of semantic structure that can effectively represent the hypertext content from the domain selected in (1).

3) Next, we developed tool(s) to semi-automatically infer a semantic structure from the hypertext content to obtain a representation of the domain.

4) We subsequently created a visual interface that presents the semantic structure in a manner that promotes visual learning.

5) We empirically demonstrated that the visual interface does help users understand and/or locate information better.

6) We then conducted statistical analysis to determine if the data conveyed in the visualization indeed reflects significant trends in the domain.

7) Finally, we improved the hypertext technology selected in (1) to add semantics that can further enhance the quality of the semantic structure and visualization.

We selected wikis and blogs for (1), and topic maps for (2). We developed two systems, ENWiC (EduNuggets Wiki Crawler) and eNulog (EduNuggets Blogger), which mine wikis and blogs respectively and extract semantic structures in the content. We implemented a topic map visualization tool named TOMU, which draws topic maps as graphs, to help create the interface required in (4). We carried out a usability experiment with ENWiC for (5) to observe if the visualization interface did help students to learn faster. We performed a statistical analysis using data obtained from eNulog to investigate if the trends and patterns that can be observed from the visualization interface in eNulog do reflect what actually happened in the domain. We developed a third system named Annoki (Annotation Wik), which aimed to improve the semantics and content of a wiki to further enhance the quality of the domain representation and visualization. We revisited step (5) with Annoki and evaluated its usability specifically for storytelling.

1.4 Contributions

The work described in this thesis has resulted in two different types of contributions.

1) *System Building:* We developed a set of components, which we configured in three different systems, each one addressing a particular objective in the context of a specific environment. ENWiC and Annoki aim to improve the users' learning experience with a wiki: ENWiC does not require any changes to the basic wiki technology, while Annoki is meant to enhance the underlying wiki technology. ENulog is designed to analyze blog content and is motivated by a market-research scenario.
2) *Empirical Case Studies:* Our second contribution is the empirical case studies we have conducted in order to refine and evaluate our hypotheses that semantic structures and visualizations help users grasp what is present in the domain. Statistical analyses are used to further support our observations.

1.5 Thesis Outline

Chapter 2 summarizes the background information for the work of this thesis, and the related research in similar areas. It first describes current online communities of practices and topic maps, and then introduces web content visualizations, its motivation, and its current applications. Chapter 3 describes the components and systems we developed to investigate the hypotheses stated in Section 1.2. The three systems are: (1) ENWiC, which utilizes visual diagrams to help learners locate and understand information on a wiki faster; (2) eNulog, which mines blog information to enable market researchers to examine the blog 'buzz' for a product; and (3) Annoki, which is designed to enhance the semantics and quality of wiki content. Chapter 4 covers the empirical studies we performed with our systems, which include: the ENWiC usability experiment that evaluated the system's ability to improve user learning; the eNulog movie case study which examined the correlation between blog buzz and movie success; and the AnnokiBlooms study where the functionalities of the Annoki system were extended to help young students plan and write stories. Chapter 5 summarizes the work and contribution of this thesis, and outlines some possible future work.

Chapter 2: Background and Related Work

2.1 On-line communities of practice

In general, the term ‘community of practice’ (CoP) refers to a group of people with some common interest, and the group members exchange ideas and knowledge to learn and/or form new ideas [23]. Communities of practice are valuable because when all members participate and contribute to the community for a common goal, they all help improve the efficiency and effectiveness of the community. Lesser & Storck [18] suggested that CoPs contribute to the development of social capital, where members learn to trust each other, achieve a sense of belonging to the social network in the community, and gain a clear understanding of the common goal. This development of social capital is valuable to business operations, as it enables members to learn faster, be more innovative, and better understand what they have to do, which results in a decrease of errors that has to be corrected as the project progresses.

A community should bring a sense of ‘togetherness’ to all members and technology has made it easier for a community to obtain that sense of togetherness, as members are no longer restricted by locations or schedules [38]. Members with an Internet connection can go online anytime and participate in the sharing of ideas. Different online tools are available for different purposes, e.g. chat rooms and video conferencing are used for synchronous interactions, while emails and discussion forums are used for asynchronous interactions [38].

Though online tools can be convenient to a CoP, they can also put stress on the members, as they may feel overloaded with too much information [38]. Since online tools have made it very easy for members to share content with the community, this may result in members passing too many messages and documents around the community, making it difficult for each individual member to digest all the information available. To prevent such problems from arising, there should be mechanisms in place to control or organize the flow of information to members.

2.1.1 Wikis

A software technology that is increasingly being used for setting up an online CoP is a wiki. Wikis allow users without any markup language knowledge to publish content on web pages. The first wiki, named the WikiWikiWeb, was developed by Ward Cunningham for the Portland Pattern Repository [55]. It was designed to enable people from the software-development community to talk about people, projects and patterns [69].

Various wiki clones were developed since then for other types of communities, and functionalities are implemented to suit different communities' needs. TWiki [65] is a structured wiki that is designed to be a data management tool. Users can use TWikiForms to structure and organize content, and templates are available to help users create new pages with the desired structure. Along with features such as revision control, access control and file upload, TWiki's advantage lies in its Plugin API and library, which enable users to develop custom plugins that can be invoked in a wiki page. TWiki has been used in companies such as Motorola and Texas Instrument for project management and collaboration [65].

MediaWiki [53] was designed for Wikipedia [67] to handle a large amount of pages and content. Since Wikipedia is targeted for people around the globe, a main emphasis in the design of MediaWiki is internationalization. Thus, the MediaWiki interface has the capability to adapt to many different languages. MediaWiki also allows users to create image galleries with ease, and it is currently being extended to provide similar support for other media galleries for Wikimedia Commons [66].

TikiWiki [58] is a content management system where users can create and share many different types of documents, such as newsletters, charts, image galleries, polls and quizzes. TikiWiki also offers communication tools such as chat rooms, messaging and web mail, and navigational tools such as categories and templates. TikiWiki has been used in a number of educational institutions with reported success [59].

There have been several projects focused on enhancing the wiki infrastructure for e-Learning purposes. In [24], O'Neil proposed the idea of converting presentation slides into wiki pages, where students can annotate the slides' content with their own comments. The underlying motivation was to help students to review and recollect what they have learned in the lecture presentations, and help instructors to monitor the students' progress and development on the wiki pages. O'Neil implemented the slides2Wiki system to automate the conversion process; instructors can write presentation slides in LaTeX or an XML-based file format (e.g. OpenOffice Impress files, KDE KPresenter files, and Apple Keynote files), and the slides2Wiki system can parse the file to produce a wiki markup for the corresponding wiki pages.

Wang and Turner [37] extended the functionalities of a conventional wiki to accommodate the needs for online course delivery. The main feature they implemented on their wiki is access control to allow users to restrict read/write permissions on individual pages. Access control allows instructors to make certain pages not visible or editable by students; for example, course outlines should not be modifiable by students, and assignment solutions should only be visible to students after all assignments have been submitted. The access control also allows students to protect pages they created, if they do not wish other students to access and/or modify them. The wiki also has a page locking mechanism to permit only one user to edit one page at a time, in order to avoid problems with concurrent edits by multiple students.

WikiDev [20] is a project management environment that integrates a set of tools, such as CVS, newsgroups and code analysis tools, with a wiki-based user interface. Students' familiarity with the web-based wiki interface can help them learn and effectively utilize the integrated tools for project management, and instructors are able to monitor students and provide feedback on the wiki.

Currently, there is ongoing research interest on how to improve a wiki's structure and semantics. This has led to the emergence of semantic wikis, which integrate semantic web technologies into wikis. The Platypus wiki [4] and the SHAWN wiki [2] utilize RDF (Resource Description Framework) metadata to specify ontological structures to the wiki pages. The Platypus wiki [4] provide users with a RDF metadata editor to insert and edit metadata to a page. The wiki also provides users with a simple way of establishing links by the use of a link table; the link table allows users to map a keyword to a URL link, and whenever that keyword appears on the content of a wiki page, it is automatically made into a link to the corresponding URL in the link table. The SHAWN wiki [2] allows users to type in property-value pairs directly in the page editor in the form of *property: value*, and the wiki automatically identifies them as metadata for the page. Both wikis use the metadata to provide better navigation support for the users.

Semantic MediaWiki [56], an extension of the MediaWiki [53], allows users to enter additional semantics, such as attributes and relations, to a wiki page. The semantics help enhance the search function, which allows users to search by metadata. The wiki can export pages in RDF format, and import vocabularies and ontology.

Oren [25] developed SEMPERWIKI where users can annotate pages with natural language description or RDF metadata. SEMPERWIKI is designed for 'knowledge workers' who need to publish and share their knowledge with ease. The descriptions and metadata helps users to easily locate existing knowledge so that users can utilize it to form new knowledge and solutions.

IkeWiki [29] utilizes semantics web standards, i.e., RDF and OWL (Web Ontology Language), to help knowledge workers formalize their knowledge. Its main features include the ability to support different types of knowledge formalization, which ranges from informal text to formal ontology, and it is also capable of handling OWL-RDFS reasoning. The wiki is designed to support users at any skills level; users with no semantic web technology experience can add content to the wiki, and a 'knowledge engineer' can formalize the knowledge added to the wiki.

2.1.2 Blogs

Another popular tool for communities of practice is the blog. Users can communicate with each other by participating in the same blogs, or link to each other's blog entries in their individual blogs. Users may blog about anything from personal events, to entertainment news. Blog entries are generally opinionated, and due to the diverse social, educational and cultural backgrounds of different bloggers, we can get many different perspectives on the same topic.

The collection of opinions originating from a large, diverse set of people can be very useful in market research. Businesses often need to evaluate the public's opinions on their products, as well as the competitors' products, to determine current trends and coherent customer groups. The benefit of using blogs is that blog data is 'free' data that is publicly accessible, which eliminates the need to manually collect data by distributing surveys and questionnaires. With the increased usage of blogs from all around the world, along with the frequent updates blogs receive from their participants, blogs are becoming increasingly valuable as a data source for research [39].

Currently, there are blog search engines available online for users to locate blog entries of interest, such as Blogdigger [41], Bloglines [42], and Technorati [57]. Using RSS and Atom web feeds from blogs, these engines provide up-to-date indices of blog entries made in the 'blogosphere'. Blogdigger [41] provides a search engine interface similar to Google [48], and search results can be syndicated as RSS/Atom feeds if users would like to subscribe to certain keywords to receive updates.

Bloglines [42] provide a variety of blog-related services to registered users; users can subscribe to their favorite blogs/websites, share their subscriptions with friends, search for blogs using keywords, and create their own blogs. As of August 2006, Bloglines have 2,600,000,000+ articles indexed for their search engine.

Technorati [57] tracks 52,000,000+ blogs as of August 2006 for their search engine. They also maintain lists of popular blogs, search keywords and tags. They identify top movies, news and books by determining how many links originating from blog entries point to a movie, a news article, or a book, respectively.

Nielson BuzzMetrics [54] provides a set of blog-mining services that are mainly targeted for market research and commercial usage. Their services provide companies with analyses of consumer generated media (CGM), such as blogs, forums and newsgroup, to determine the current 'buzz' surrounding specific brands and/or companies.

BlogPulse [43] is a set of blog-analysis tools provided by Nielson BuzzMetrics. It includes a variety of tools. A trend search enables users to graph the amount of blog posts related to a term of their choice over a specified amount of time. In the featured trends section, BlogPulse identifies the popular terms from the blogosphere and performs the trend searches on these terms. Finally, a conversation tracker enables users to enter a root URL to a blog entry and see what subsequent entries from other blogs refer back to that root entry.

Not all blogs are syndicated through web feeds, which are missed by most blog search engines. Nanno et al. [22] devised a system that utilizes a crawler to mine and monitor

Japanese blogs. Japanese users have used HTML pages as their 'diaries' before the emerging of blogging tools, and some continue to use these pages instead of switching to blogging software. To obtain data from these diary pages, the system is designed to analyze HTML data to determine if a page is structured like a blog. The system can also find more blogs by examining links in blog entries that has been discovered, or by taking links from known blog link lists. Along with identifying blogs, the system can also identify terms that appear frequently in blog entries at certain timeframe to determine what the popular topics are at the time.

Kurashima et al. [14] utilized blog content to provide travel advices to users. They implemented a system that discovers association rules from travel blog entries to find patterns in the tourists' experiences in certain destinations. A user can specify a location and a time period, and the system returns the types of activities that are popular in that location during that timeframe based on the patterns it found in the blog entries.

In [12], Herring et al. used social-network analysis techniques to show that the blogosphere is partially connected. While the results did not suggest that the entire blogosphere to be interconnected, Herring et al. discovered that there are many one-way links to popular 'A-list' blogs, making them the central points of the social network. They also found some parts of their sample of blogs to be more interconnected with two way links, and upon examining these blogs, they found that these blogs are related to a common subject, such as Catholicism and home schooling. This suggests that bloggers with same interests are more likely to link each other. We can see this as bloggers gathering into a community of practice where they share knowledge and ideas about a common subject of interest through blogs.

2.2 Topic maps

Inspired by semantic-networks formalisms [33], topic map is a graph-based knowledge representation mechanism for modeling domain knowledge [62]. The topic-map notation is an ISO standard with a corresponding XML-based syntax. It represents the domain using topics, associations and occurrences. A topic can represent any type of entity, such as a concept, an idea, an object or an event. An association is a relationship between two topics. These generalizations allow topic maps to be flexible enough to accurately represent any subject domain.

The benefit of using a topic map is that it gives the domain a semantic structure. It is also very useful for indexing large sets of data and documents [11]. One can build a topic map containing all topics covered in the data set, where each topic is associated with every related resource in the data set as its occurrences, and topics are connected to each other through associations. Users can then identify what types of information are available in the domain by examining the topic map without actually reading the information in detail [11]. A topic map can also enrich search results; when users query for a topic of interest, they can see all related topics through the associations, and all related resources through the occurrences.

Topic maps are straightforward to use programmatically, as they can be represented using the XML topic map (XTM) format [62], which is a standard that can be used by other external applications. Topic maps are simple to create, which enables users who are not familiar with semantic web standards to work with them. The simplicity of topic maps also makes it easier for software developers to design and implement an automated process to create the topic map. [Figure 1] shows a portion of a topic map in XML format. A topic is identified by its ID and its name. The parent of the topic can be specified with the instanceOf tags. A topic can have variant names, which are alternative forms of the topic's name. An association is defined by a type, and the two member topics in the association.

```
<topicMap xmlns="http://www.topicmaps.org/xtm/1.0/"
xmlns:xlink="http://www.w3.org/1999/xlink">

<topic id="t1-clownfish">
<baseName>
<baseNameString>
Clownfish
</baseNameString>
<variant>
<variantName>
<resourceData id="name">Anemonefish</resourceData>
</variantName>
</variant>
</baseName>
</topic>

<topic id="t2-marvin">
<instanceOf>
<topicRef xlink:href="#t1-clownfish">
</instanceOf>
<baseName>
<baseNameString>
Marvin
</baseNameString>
</baseName>
</topic>

<topic id="t3-anemone">
<baseName>
<baseNameString>
Anemone
</baseNameString>
</baseName>
</topic>

<association id="1">

<instanceOf>
<topicRef xlink:href="#t0-livesIn"/>
</instanceOf>

<member>
<roleSpec>
<topicRef xlink:href="#t1-clownfish"/>
</roleSpec>
</member>

<member>
<roleSpec>
<topicRef xlink:href="#t3-anemone"/>
</roleSpec>
</member>

<topic id="t0-livesIn">
<baseName>
<baseNameString>
Lives in
</baseNameString>
</baseName>
</topic>
```

```
</association>

</topicMap>
```

Figure 1: XML markup for four topics: "Clownfish", "Marvin", "Anemone" and "Lives in", and an association. Clownfish has a variant name of Anemonefish. Marvin is a type of clownfish, as specified in the instanceOf tag. The association relates clownfish and anemone together.

In [26], Power illustrated how topic maps are useful for context information system, and described a system that enables context information providers to import and export each other's topic maps to share data. Lavik et al. [16] implemented the BrainBank Learning tool, which allows users to create and edit topic maps online. They investigated its usage for e-Learning and found that students were interested in building their own knowledge management system, and such documentation of their knowledge has helped them in their learning process. Librelotto et al. [19] implemented the TM-Builder, which aimed to give ontology in the form of a topic map to information resources written in XML document. They introduced the language XSTM (XML specification for Topic Map), which can be used to describe how the information resources should be translated to a topic map. Based on an XSTM specification, a topic map is generated for the information resources using XSL processing.

In this thesis, we use topic maps to represent the content present in wikis and blogs. While there are semantic wikis that also aim to organize and structure the wiki better for users, the advantage of using topic maps to represent the wiki semantics is that it gives a model to the domain that can be easily pictured by the users. Most semantic wikis rely on textual metadata, and topic maps can provide an entirely different experience for the users to help them see the structure of the domain in a different manner. To represent blogs as topic maps for market research, we treat blog entries and products as topics, and establish associations between each entry and the product(s) it covers. The market researchers can then examine the topic map and identify the blog activities surrounding a product.

2.3 Web-content visualization

Most web content today is tabulated in text indices and link lists. Search engine results are displayed as a list of links to relevant pages. On a wiki, users would navigate by looking at links in the body text of a page or in a special index page, such as a Category page. Similarly, blog entry archives are displayed as a list of links in chronological order. Users may get overwhelmed when navigating such lists when there are many pages linking many resources.

To help users navigate through large sets of links to resources, this thesis focuses on improving the way these links are presented to help users better recognize what resources are available to them. This is especially important in e-Learning applications, where students need to read and comprehend the resources given to them on their own. If the students have difficulty navigating through the resources, it may drastically slow down their learning progress, and cause feelings of frustration and dissatisfaction, which can decrease their motivation to learn. Students may also find it difficult to keep track of where they are, what they have gone through, and what materials they should turn to next after going through a large number of links. A possible solution to this problem is to visualize a resource link list into a meaningful diagram.

2.3.1 Visual Learning and Bloom's Taxonomy

In the Theory of Multiple Intelligence, Gardner [10] divided human intelligence into multiple categories. He stated that in order to help all students to learn, instructors must cater to all categories of learning style. The different types of intelligence Gardner identified in [10] are:

1) **Verbal/linguistic** - the ability to understand and convey with words;
2) **Logical/mathematical** - the ability to work with numbers and logic;
3) **Visual/spatial** – the ability to understand and express with pictures;
4) **Body/kinesthetic** – the ability to use physical movements;
5) **Auditory/musical** – the ability to hear and use sounds and rhythms;
6) **Interpersonal**– the ability to communicate with others; and

7) **Intrapersonal** – the ability to discover things within one's self.

Traditional classroom teaching focuses on verbal intelligence and mathematical intelligence. Students are usually required to take text notes, read textbooks, and take written exams. They also need to work with numbers and perform calculations and reason about formulas. It is important to recognize that not all students can comprehend textual and numerical information, and that educators should try to convey the same information through various different means, such as diagrams and graphs. While visual, kinesthetic and auditory mediums are not completely absent from a typical classroom, they should be utilized more frequently to engage students with different learning styles.

The term "graphic organizers" (GOs) refers to pictures that depict knowledge and/or ideas. In education, graphic organizers give students a different perspective on the content presented, and they encourage students to think and brainstorm in a more creative manner [15]. There are different types of graphic organizers designed for various purposes, such as: concept maps for describing knowledge and ideas; double cell diagrams for comparing two entities; hierarchy diagram for classifying entities; and linear string for listing sequence of events [60].

Bloom's taxonomy [3] described the six different levels of cognitive understanding that learners can achieve. The six levels include:

1) **Knowledge**, where the learner can remember facts and information;
2) **Comprehension**, where the learner gained enough understanding to describe, organize, and translate facts;
3) **Application**, where the learner can apply knowledge to a new problem;
4) **Analysis**, where the learner can recognize and organize parts that form the domain;
5) **Synthesis**, where the learner can generalize facts enough to combine them and create new plans and solutions to problems; and
6) **Evaluation**, where the learner knows the domain well enough to criticize and judge ideas and theories.

We can use different types of GOs to target the different levels of cognitive understanding. Donelan [8] suggested the following GOs for each cognitive level:

1) At the knowledge level, spider maps allow users to list entities in the domain, while linear strings allow users to describe sequences.
2) At the comprehension level, hierarchy diagrams allow users to organize entities.
3) At the application level, flow charts allow users to form a course of action to solve a problem.
4) At the analysis level, fishbone maps allow users to identify cause-and-effects, while concept maps allow user to link concepts together.
5) At the synthesis level, idea maps allow users to brainstorm and combine ideas.
6) At the evaluation level, double cell diagrams allow users to compare similarity and difference between two entities, while comparison matrix allows users to compare different properties of multiple entities.

Visualizing web resources as GOs and diagrams can be very useful for e-Learning because of a visualization's ability to summarize a large set of information into small figures that can help students digest the information faster and easier. The potential to reduce learning time using visualizations is important in e-Learning because the success of e-Learning depends on whether we can capture the students' attention enough to motivate them to carry out self-learning. If we present an overview of a large set of information in a diagram, the students can immediately gain an idea of what the content is as a whole, which can engage them to explore the content further in detail. A visual diagram also enables students to recognize how smaller sets of information relate to each other in the 'big picture', which helps them navigate the information in a meaningful order. This is also important in e-Learning as students can easily lose their sense of direction when they are learning by themselves, especially when there are so many resources available. Visual diagrams can become their 'road maps' to steer them in the right directions to the appropriate resources.

2.3.2 Applications of web-content visualizations

Smith [32] implemented a set of visualization tools for Usernet. Usernet can potentially have many messages, and a conventional Usernet client does not structure these messages in a manner that conveys the ontology of the messages. Using various types of data, such as thread history and user participation history, Smith created a set of visualizations that helps users easily see the activities that are occurring in the group.

In [6], Dennis and Jarrett created visualizations of syndicated web content. Their system provided three types of network visualizations: the item-content network, which visualizes words that appeared in the web feeds as nodes, and words that appear in the same article are linked together by edges; the source-content network, which visualizes the web feeds as nodes, and an edge is made between two nodes when the system finds text that appears concurrently in the corresponding web feeds; and the subscription affiliation network, which relates web feeds globally based on blogrolls.

CmapTools [5] and VisSearch [17] utilize visual diagrams to enhance web search. CmapTools [5] uses concept maps, a type of GO, to help users visually organize knowledge. The map contains concepts and propositions; users can click on a concept node, and the system generates search queries based on relating concepts and authority concepts. Search results are ranked based on distance matrices taken from the map. VisSearch [17] allows users to create a search graph. Each node represents a query, and users can establish associations between related queries. Users can combine query nodes to form new queries. When users find a relevant link, they can associate it to the appropriate query node on the graph as a bookmark. VisSearch can give recommendations of potentially useful links based on search history; a VisSearch server records the search result history of each user and analyzes the data to look for commonalities, so that when another user makes similar search queries, it can provide some recommendations based on what seemed interesting to other users.

Visualization of topic maps has been applied to some web domains as well. Touchgraph [64] is an open source tool that takes an XTM file as input and displays the content as a graph. To demonstrate the tool, Touchgraph.com released a wiki browser for the Meatball Wiki [52] and the Emacs Wiki [47]. Since the browser uses the link database of the wiki to create the topic map, its use is limited to wikis that have a link database. Touchgraph also released browsers for some popular sites: a LiveJournal browser, which shows users as nodes and 'friendships' between users as edges; an Amazon browser, which shows items as nodes, and an edge between two items indicate users who brought one of the items also brought the other; and a Google browser, which show sites as nodes, and the browser fetches related links between sites to establish associations between nodes.

TM4L (Topic Map 4 Learning) [7] is a topic map editor and viewer designed for e-Learning. The editor provides pre-defined topic types and associations, which are common in an e-Learning environment. This enables instructors to easily add and classify content. Instructors can also define their own topic types and associations for their own needs. Students can access the topic map via the viewer, which visualizes the topic map as a graph or a tree.

EduNuggets [34] is an intelligent education content management system. It represents subject domains as topic maps, and instructors can link online source materials to the topics in the maps. The system provides two clients: an administrative client for instructors and a visualization client for students. URL pointers to relevant web documents are stored in a repository. Using the administrator program, instructors can start a new domain and add new topics into it. They can also establish associations between topics, and add document links to the repository. Each document link is treated as a 'nugget', and each nugget is associated with a topic in the domain that is relevant to it. Instructors can also provide EduNuggets with a resource directory containing links or copies of documents; a crawler goes through the directory and processes the documents to create the appropriate topics and nuggets. Documents can be simple text/HTML pages, audio/video files, and SMIL files. Associations are then created by determining if two topics have nuggets that refer to the same document. Students can access topics and nuggets via the visualization client, which draws the domain as a graph. Information retrieval algorithms such as latent semantics indexing and Naïve Bayers classifiers are also implemented in the system to allow students to make queries to find desired topics.

There are limitations to the EduNuggets framework, mainly with the automated topic map generation process. Since EduNuggets has to handle different types of media originating from different sources, it has to analyze files with vastly different structures and layouts. Due to the diversity present in the source materials, the system may not understand the content of the files enough to create appropriate topics and associations for them. As a result, the quality of the topic map suffers, which makes the domain visualization less effective for users.

The work of this thesis is based on the EduNugget philosophy of using topic maps for content management; however, instead of having the system handle content from any external source, this thesis focuses on specific source domains that contain similar types of documents. Since we are familiar with the structures and types of files we need to handle, we are able to instruct the crawler on how to analyze the files to get suitable topics and associations.

Since learning usually requires collaboration between groups of people (e.g. teachers/students sharing notes or working in a group project), and e-Learning can be delivered through online collaborative tools, we focus on designing our framework specifically for online communities of practices such as wikis and blogs.

All wikis share a similar structure and organization. Though not enforced, most users usually follow these organization guidelines when writing wiki content:

- A page deals with only one topic/subject.
- The title of the page is the topic.
- Intra-wiki links can be made simply by writing certain words on the body text with a special syntax, where the syntax depends on the wiki software.
- Intra-wiki links are used to link related topics (i.e. pages).
- Links to external sources related to the page's topic are also listed on the page.

These properties enable us to create topic maps that better represent the domain covered in the wiki. Acquiring a good semantic representation of the domain helps improve the quality of the visualization interface.

The benefits of linking wikis and topic maps go both ways. Wikis provide the source materials needed for users to further explore a topic from the topic map in detail, while topic maps help structure wikis into more comprehensible 'road maps'. If there are many pages in the wiki, the number of intra-wiki links that the users face can be overwhelming. Topic map visualization provides users with an overview of the pages available in the wiki to reduce the cognitive load on the users.

Blogs follow a similar organization and layout. Blog entries are listed in chronological order, and users usually give an entry a title that reflects the subject of the entry. Users also frequently write links to related sites in their entries. Based on these assumptions, we designed a crawler that retrieves blog entries to make them into topics for the topic map, which is then visualized to give an overview of the bloggers' activities.

Chapter 3: Components and System Implementations

In this chapter, we discuss the components and systems we developed to empirically evaluate our hypotheses. We describe the two components we used for producing topic maps of hypertext resource domains: the Hypertext Document Retrieval (HDR) component and the Live Topic Map Feed (LTM) component. HDR is used for the ENWiC (EduNuggets Wiki Crawler) and eNulog (EduNuggets Blogger) systems for retrieving information from wikis and blogs, respectively. LTM is used in Annoki (Annotation Wiki) system, which structures the wiki database as a topic map to produce a live feed. Next, we introduce our Visualization Interface (VI) component, which takes the topic maps produced by HDR or LTM to create various visualizations of the topic maps. VI is used to produce the client interface for all three of systems above. Finally, we describe the three systems we created in detail, and how the constituent components interact in the context of these systems. These systems are referred to as one system family since they originate from the initial EduNuggets [34] design.

3.1 Component overview and systems summary

[Figure 2] summarizes our components and systems. There are three core entities common between all systems:

1) *The resource domain*, which includes the content developed by a community of practice, as well as the software they run on (e.g. a wiki software or a blog tool);
2) *The semantic representation* of the domain, which is the topic map generated based on content from the resource domain; and
3) *The client interface*, which is essentially the interactive visualization of the topic map.

In the systems composed of these components, users are presented with parallel interfaces; they can add/edit content using the interface of the "native" domain software, i.e., wikis and blogs, and they can visually navigate the content from the client interface of the component. HDR and LTM deal with the construction of the semantic representation from the resource domain, while VI deals with the visualization of the semantic representation for the interface.

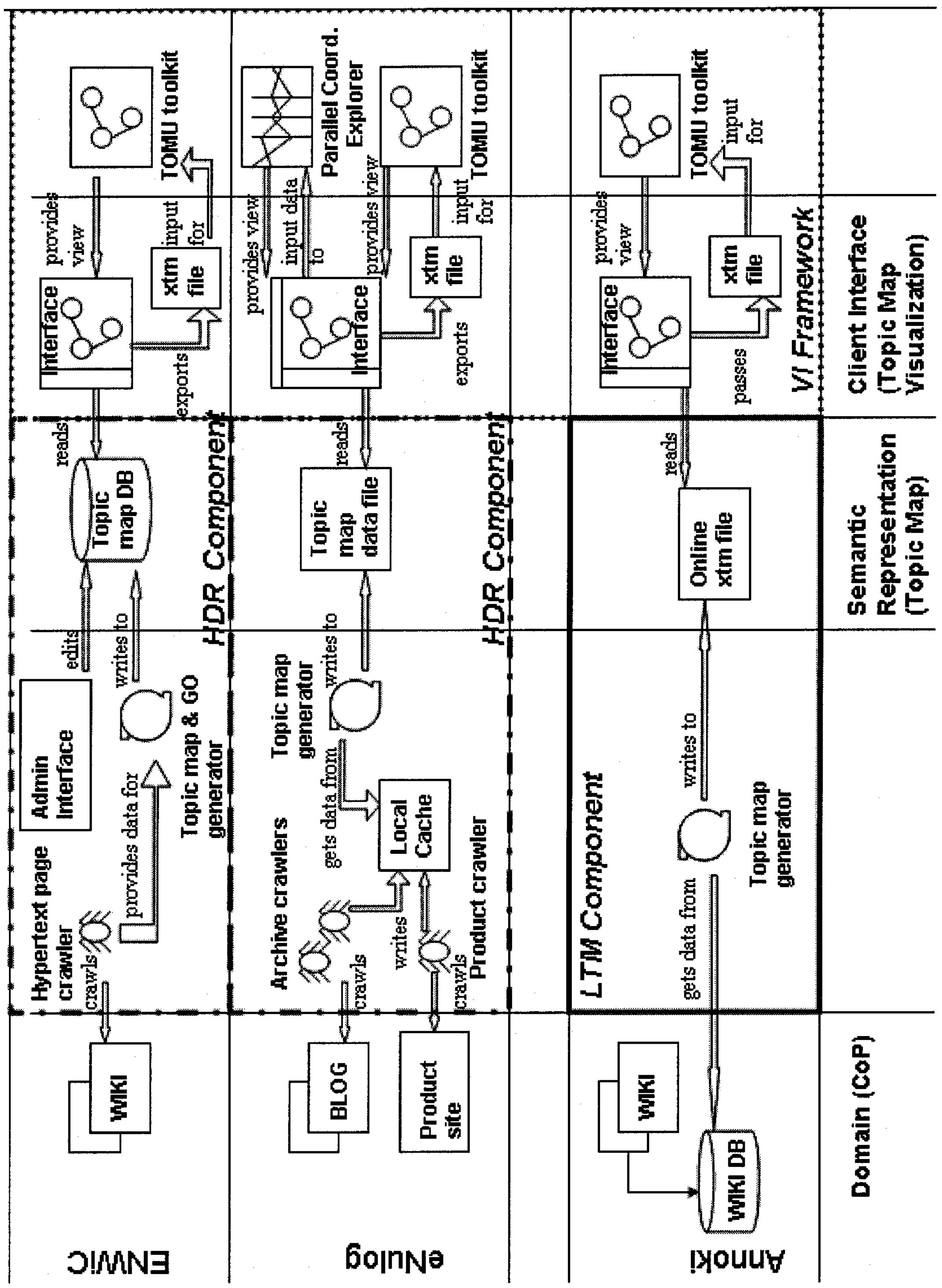


Figure 2: Systems summary. Each row corresponds to a system, and each named column represents a core entity in the systems. The dashed rectangles highlight where the HDR is utilized; the bolded rectangle highlights where LTM is utilized, and the dotted rectangle indicates where VI is utilized

3.1.1 The Hypertext Document Retrieval (HDR) Component

The HDR component is relevant in the scenario where the hypertext documents in the domain are accessible by URIs, and they have to be retrieved and indexed to generate a topic map that conveys the content available in the domain, and their relationships with each other. Hypertext page crawlers are used for information retrieval, and the retrieved data is processed to generate a topic map. ENWiC and eNulog are built using this component.

3.1.1.1 Document Retrieval

The goal of the document retrieval process is to collect hypertext documents from the domain. When users provide a starting URL, a hypertext page crawler retrieves the hypertext markup of the document, and extract links from <a> tags in the markup to discover new documents. The crawler can also collect segments of the document based on tags; for example, the crawler can be configured to collect words encapsulated inside <title> tags to get the title of the document. All tag analyses are based on regular expression matching; the crawler is configured with a set of tags to examine by adapting the regular expressions used to match the hypertext markup in the code.

The benefit of using a crawler to create the topic map is that it offers implementation independence. Regardless of what software is used for the resources in the domain (e.g. any of the wiki clones), the crawler can process it and create a good topic map representation just as long as the pages are displayed in HTML. The crawler is also not limited to sites for communities of practice only; it can also handle any well-structure websites where topics are individually organized into pages.

3.1.1.2 Topic Map Generation

After the crawler collects data from the resource domain, they are processed to produce a topic map that helps represent the domain semantics. The procedure varies depending on what resources are in the domain, and how they have to be structured based on the purpose of the envisioned application.

Since this thesis deals mainly with wikis or blogs as the resource domains, a simple topic map transformation is to make each hypertext page as a topic, and each page link as an association between the pages. This simple transformation is effective for wikis especially because each wiki page usually deals with one topic only, and authors usually set up intra-wiki links to point to relating topics. Based on domain knowledge, other domain-specific topic/association transformations can be implemented as well to enhance the topic map for a specific application. The topic map generation process does not have to be completely automated; we can provide users with an interface to manually apply some topic/association transformations.

3.1.2 The Live Topic Map Feed (LTM) Component

When the implementation of the domain software, i.e., wiki, is open and all its data is accessible, the topic map can be extracted directly based on the data contained in the wiki database (as opposed to data collected by crawling the hypertext interface exposed to the wiki users). As the structure of the database table is known, the transformation of particular table tuples into topics and associations is straightforward. While this approach is less general, one can apply this method only to wikis developed using the particular wiki software, having control over the domain software enables:

1) direct access to the database by making a web feed available, which eliminates the need to crawl the domain and maintain a local database for the discovered information; and
2) the injection of additional semantics and functionalities to the domain software, which improves the quality of the topic map, and at the same time improves the interface of the domain software. This helps improve users' experience on both ends of the system, instead of just on one end with the client interface.

The LTM component consists of structuring the content in the domain database to form a topic map, which is then used as a web feed to broadcast up-to-date content from the domain. Annoki is built with this component, where we modified the implementation of the wiki domain for better topic map generation.

3.1.3 The Visualization Interface (VI) Component

The process of producing a visualization of the topic map is common between all our systems. This process involve taking the topic map from the topic map generation step, which may be stored in various mediums depending on the system implementation (e.g. a database or a local file), to create an input file for a visualization system.

Two visualization systems are used in VI: TOMU (Topic Map User Interface), and the Parallel Coordinate Explorer [31]. TOMU takes topic maps written in XTM files as input, and render them as graphs. Since XTM is a standard that can be used in various open source tools, such as TM4J [63], the XTM files produced by our systems for this process can also used in other applications, which increases the component's flexibility. The Parallel Coordinate Explorer [31] enables users to view information across different dimensions. It takes text files as input, where each line is a tuple. It was not originally designed for topic maps, and thus the topic map has to be mapped into the required input format in order to use this system.

3.1.3.1 The TOMU visualization system

TOMU takes topic maps in XTM files and visualizes it in a graph. Its interface design is based on Touchgraph [64], and it is written in Java. The main goal in the development of TOMU is to provide a system that can be easily customized by both programmers and non-programmers. Programmers can configure the system by extending and adding to the default classes and packages, while non-programmers can, to a lesser extent, configure the interface using configuration files. This allows for administrators such as teachers to have some control over the look of the interface without having to rely on programmers to make the changes. [Figure 3] describes the architecture of TOMU.

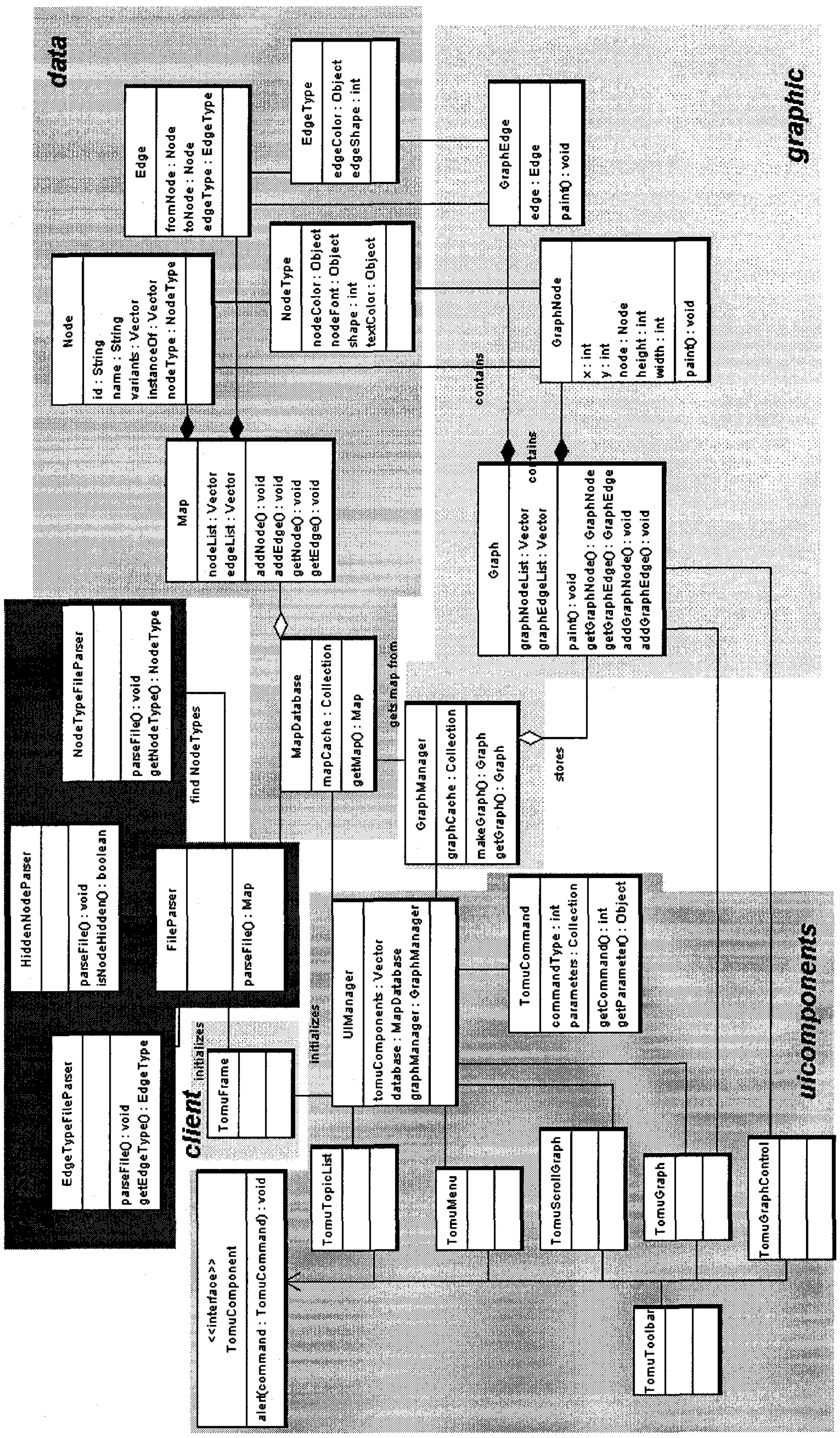


Figure 3: UML class diagram of TOMU.

The TOMU classes are organized into five packages:

1) **Data** – Contains classes that define the data structures for topic maps. *Map*, *Node* and *Edge* are similar to the *TopicMap*, *Topic* and *Association* classes in TM4J [63] respectively, with methods and parameters tailored more specific to TOMU. Other classes in this package include:
    - *NodeType, EdgeType* – Represent a type of node/edge, respectively. While the instanceOf tag in the XTM specifications provide a standardized way of classifying topics and associations, the presence of these two classes allow for additional classifications that are more domain-specific. Moreover, domain-specific attributes can be implemented in these classes by the developer. For example, when TOMU visualizes the topic map, users may want a particular set of nodes/edges to be colored the same way. In the XTM format, there is no direct way to add a Color attribute to a topic. Also, since the Color attribute has no meaning outside of the context of TOMU, adding it to the XTM file itself may not be beneficial for others who want to export the file. Thus, we implemented a Color parameter in *NodeType* and *EdgeType*, which is visible only in our application, and not in the XTM file itself.
    - *MapDatabase* – Stores Map objects. Other classes can retrieve Maps from this class.
2) **Parser** – Contains classes that parse input files. The core *FileParser* class parses XML topic map using TM4J [63] and creates the corresponding Map object. *FileParser* can be extended to tailor the parser for applications with different input file format. Configuration file parsers are also in this package. The default configuration file parsers included in TOMU are:
    - *NodeTypeFileParser, EdgeTypeFileParser* – Parse the files that define node types and edge types respectively, and create the corresponding *NodeType* and *EdgeType* objects. These parsers can be extended if *NodeType* and *EdgeType* were altered for domain-specific reasons.

- *HiddenNodeParser* – Parses the file that defines what nodes should remain hidden in the application.

Developers can add more parsers to provide more domain-specific configuration files that the users can work with.

3) **Graphic** - Contains classes that define how the topic map should be visualized as a graph. This package is only used by the UI classes that handle the graph portion of the interface. *Graph, GraphNode* and *GraphEdge* corresponds to *Map, Node* and *Edge* respectively, with extra parameters that describe how these entities should be visualized, such as the location of each node on the graph, the size of the node, the size of the graph, etc. These parameters are decided by *GraphManager*, which takes a map from the *MapDatabase* and translates it to a graph. Location of the nodes are initially determined by a random number generator; if the graph is not very large, the locations are then refined by using the force-directed method implemented in [40]. For very large graphs (e.g. >10,000 nodes), the force-directed algorithm would consume too much time to execute, and thus the system leaves the nodes in their initial randomly generated locations in exchange for performance. Furthermore, from our observations of subjects in the ENWiC experiment (see Section 4.1), we found that extremely large graphs tend to overwhelm the subjects too much to be useful, and thus it may not be worth the computation time to lay out the large graphs that subjects tend not to use. Along with laying out graphs, *GraphManager* also stores the graph generated, and UI classes can retrieve graphs from it.

4) **UI Components** – Contains all user interface components. The core of the UI classes is the *UIManager* class, which acts as the central point of communication to all UI components. All user events captured by the UI components is sent to *UIManager*, which then broadcasts the event to all registered UI components. The event is sent as a *TomuCommand*, which contains the type of event, and any parameters associated with the event. Each UI component then updates itself based on the event, if necessary. The UI components can also obtain data and objects from *MapDatabase* and

GraphManager through *UIManager*. The UI components available in TOMU are:

- *TomuGraph* – The core of the interface, which displays the visualization graph of the topic map. The interactive graph allows user to drag or click on the nodes. When the cursor is hovered over a node, a tooltip is displayed to show the topic information. Right clicking on a node brings up a context menu for additional features, such as hiding the node, or launching an associated URL of the node on a browser, if available.
- *TomuGraphControl* – Provides additional functionalities to manipulate the graph display, such as filtering edges, changing the size of graph, and changing the lengths of edges to widen or tighten graph.
- *TomuTopicList* – Lists all the nodes in the topic map. Users can also filter the topic list by entering regular expression search terms.
- *TomuMenu, TomuToolbar* – Provide basic functionalities, such as opening a topic map, refreshing the topic map, and exiting the application.
- *TomuHistory* – Keeps track of the nodes the user has visited during the lifetime of the application.

All UI components implement the interface *TomuComponent,* which describes the method notify() that is called by *UIManager* when an event occurs.

5) **Client** – Contains classes that are specific to the domain of each application. The naming and specification of this package is not fixed, as it depends entirely on the application using TOMU. There are steps that classes in this package should do to utilize TOMU:

- Initialize an appropriate file parser from the parser package, which helps build the topic map from some input stream.
- Initialize a *UIManager* from the uicomponent package.
- Initialize the desired UI components and add them to the *UIManager* so they can receive events.
- Display the UI components.

The default TOMU Client package contains the class *TomuFrame,* which lays out all the default TOMU UI components and allows user to load a topic map from a local XTM file.

To integrate TOMU into a system, we must write the client classes that utilize TOMU's UI components to create the client interface for the system, either for an applet or an application. Depending on the system and how topic map is stored, we may also need to extend the parser from TOMU to export the topic map into a XTM file.

3.2 The ENWiC System

ENWiC works with wikis in the resource domain, and deals with the scenario where we are not aware of the implementation details of the wiki software. It implements the HDR component and the VI component in Java.

ENWiC is designed to enhance wiki usage for e-Learning, with instructors acting as the administrators, and students acting as clients. The system's use of graphic organizers is designed to help students understand relationship between the topics in the content better, which can improve their learning progresses. The system is also designed to simplify instructors' work in producing course content; instructors just have to provide the crawler with a starting URL from the wiki, and the crawler automatically generates the topic map. The system also provides the instructors with an Administrator interface, where they can use wizards to manually add GOs to a topic map.

[Figure 4] shows the system architecture of ENWiC and how the components fit into the HDR and VI components. WikiCrawler and WikiCrawlerUI are used for document retrieval; WikiRecorder, AdminInterface and EduNuggetsDatabase are used for topic map generation, and EnwicViewer, EduNuggetsDatabaseClient, and the entire TOMU package are used for visualizations.

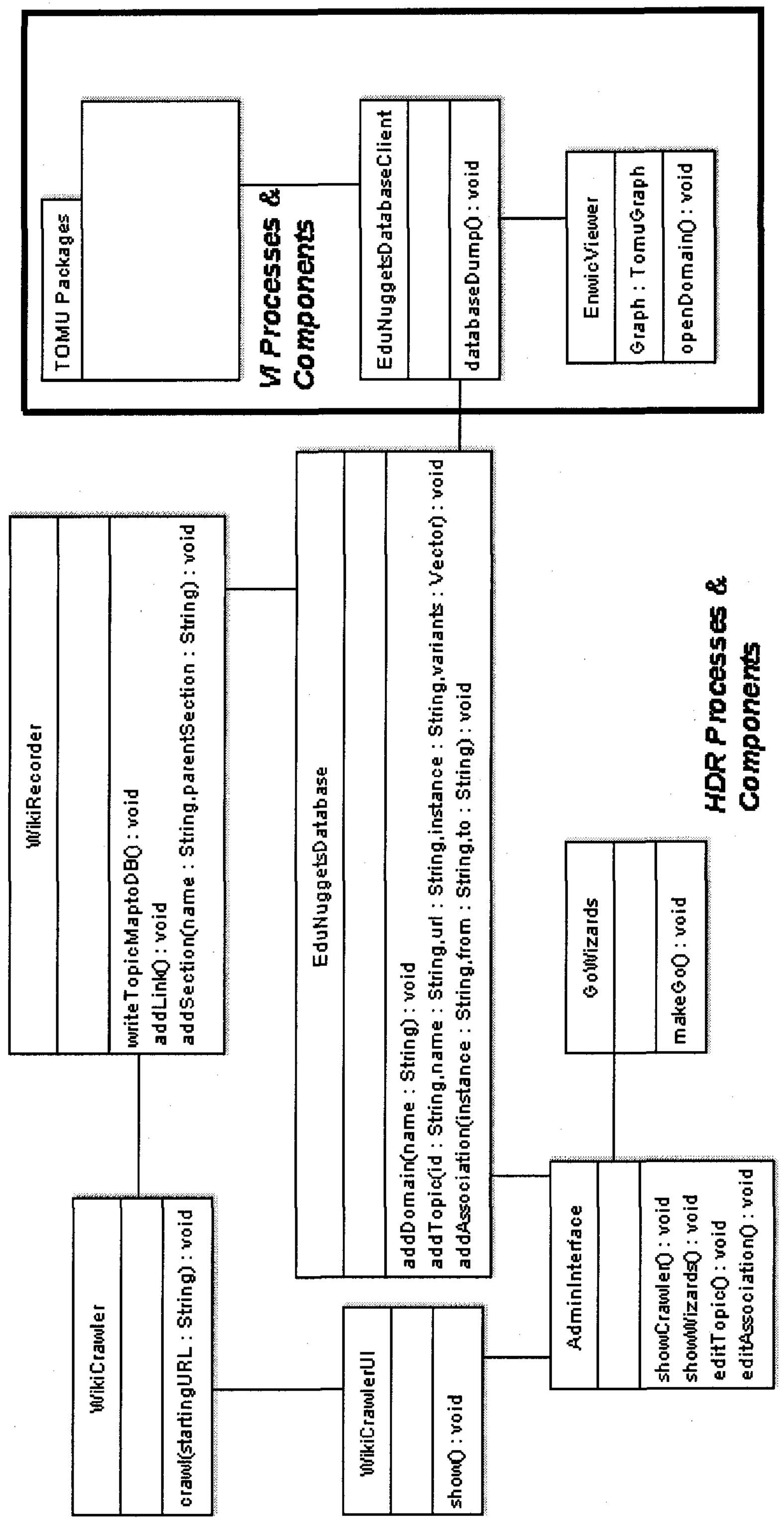

WikiRecorder
writeTopicMaptoDB() : void
addLink() : void
addSection(name : String,parentSection : String) : void
WikiCrawler
crawl(startingURL : String) : void
TOMU Packages
VI Processes & Components
EduNuggetsDatabase
addDomain(name : String) : void
addTopic(id : String,name : String,url : String,instance : String,variants : Vector) : void
addAssociation(instance : String,from : String,to : String) : void
EduNuggetsDatabaseClient
databaseDump() : void
WikiCrawlerUI
show() : void
AdminInterface
showCrawler() : void
showWizards() : void
editTopic() : void
editAssociation() : void
GoWizards
makeGo() : void
EnwicViewer
Graph : TomuGraph
openDomain() : void
HDR Processes & Components

Figure 4: Architecture of ENWiC, described with a UML class diagram (Note: only core classes and functionalities are illustrated). Classes that are implemented for VI are outlined with a bold rectangle; the rest of the classes are for HDR.

3.2.1 HDR Implementation for ENWiC

3.2.1.1 Document Retrieval

The crawler used in ENWiC is a hypertext page crawler. It takes a starting URL from the user as input and finds wiki pages in a breadth first search manner. The crawler examines the HTML markup of each page to find hyperlinks to other wiki pages, and the links are placed in a FIFO queue. The crawler continues visiting pages until the queue is empty. The ENWiC user also has the option to specify the maximum number of pages the crawler can visit to limit the size of the topic map. To prevent the topic map from expanding too far away from the wiki domain, when the crawler visits a page that is not from the wiki server (i.e. its URL does not share the same host as the starting wiki URL), the links on that page would not be placed in the queue. This ensures the queue only contains links from the wiki that the user is interested in, and the resulting topic map contains only associations relevant to that wiki domain.

All text matching and analyses are based on regular expressions. The crawler treats the markup as the query for regular expression string matches to identify HTML tags. The tags the crawler is interested in are:

1) "<a href=" tags, where new links can be found;
2) Tags that emphasize words, such as '<title>', and '<b>', '<i>'. Words in these tags can serve as keywords.
3) Texts that appear before tables and lists (i.e. text before a <table> tag or a <li> tag), which are potentially titles for tables and lists that may be interesting to users;
4) Header tags, which normally contain section headers. The crawler looks for them in descending order (<h1>, <h2>... <h6>) to preserve the order.

The crawler is called via the WikiCrawlerUI object, which provides a simple interface that takes user's input, such as the starting URL and a name for the domain to save the results in. Calling the crawler automatically calls the topic map generator as well.

3.2.1.2 Topic Map Generation

The topic map generation process is designed with the structure of a wiki in mind. Each page in the wiki is to be treated as a topic, and each intra-wiki link is to be treated as an association. The class WikiRecorder serves as the topic map and graphic organizer generator for the process, and EduNuggetsDatabase stores the topic map in a database. The EduNuggetsDatabase object and the database itself are adopted from the original EduNuggets [34] system.

WikiRecorder performs the following steps to generate the topic map:

1) The page's topic ID and base name are made using the title of a page, i.e. the text inside the <title> tags. The ID is prefixed with a number to ensure it is unique.
2) The emphasized words found on the page (e.g. words encapsulated in <b> tags) are made into variant names for the page's topic.
3) The link paths in the breadth-first search tree created by the crawler are preserved using instance-of relationships to link them together in the topic map.
4) All other page links are preserved using 'page-link' associations.

While processing each page to generate a topic, WikiRecorder also keeps track of the number of links leading into and out of the page to assign ranks (a 'link-in' rank and a 'link-out' rank, respectively) to the page. The rank may help students to identify useful and popular pages; a high link-in rank may indicate that the page contains useful information that are relevant to many different topics, while a high link-out rank may indicate that the page is an index page with links to relating topics.

WikiRecorder stores all topic map information into a SQL database through the EduNuggetsDatabase object. Using a database allows all instances of client programs to access any of the topic maps in the database. All the topics and associations generated by each execution of the crawler/topic map generator are listed under a single domain in the database. The name of the domain was specified by the user in WikiCrawlerUI at the start of the document retrieval step.

3.2.1.3 Graphic organizers generation

By following the instance-of relationships, the users can see the path taken by the crawler to get to each discovered page, and this path can serve as a linear string diagram that associates the starting page to the discovered page. Users can distinguish between the shortest path and alternative longer paths based on the instance-of relationships and page-link associations.

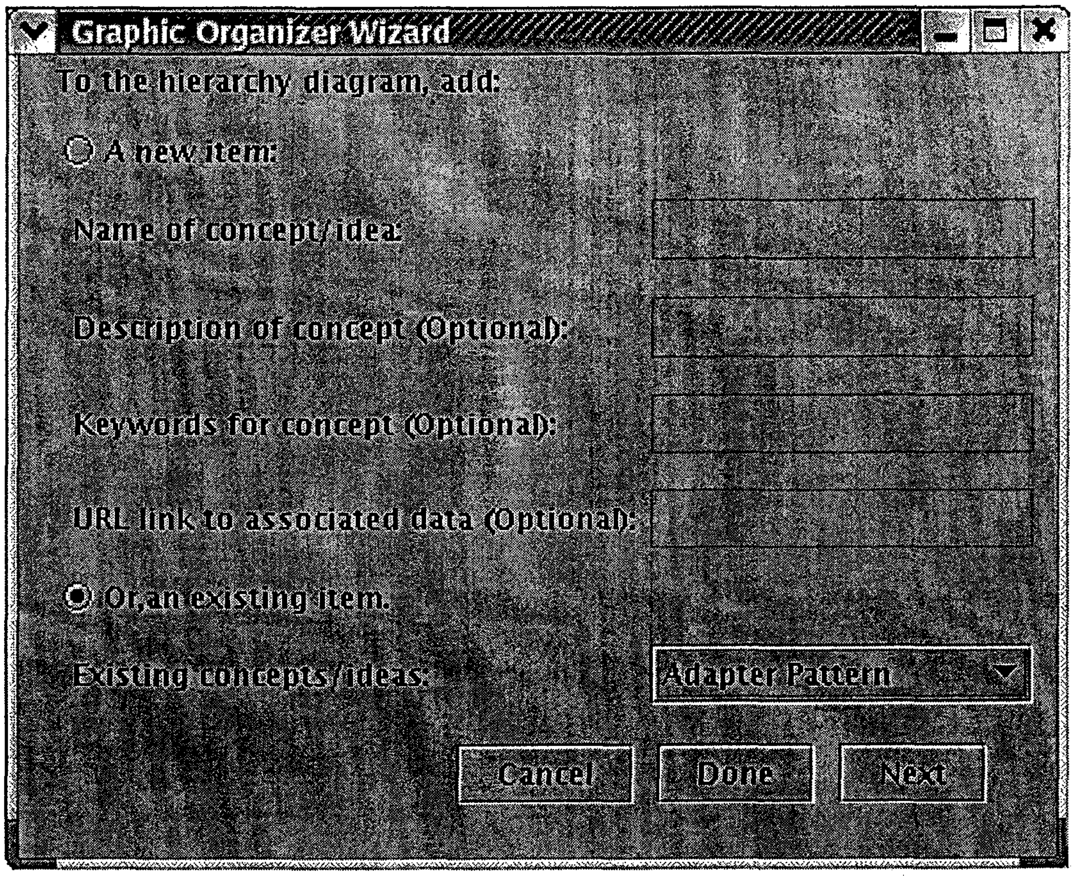
To better organize the links found on a wiki page, WikiRecorder divides the links into groups based on the sections they fall in to form a hierarchical-like structure. This is accomplished by using the header texts found by the crawler. Since sections of a page usually begin with the header texts, WikiRecorder uses the header texts to represent the sections. Each header text is made into a topic, with the text as the base name, prefixed by the word "SECTION:". The prefix helps indicate that the topic represents only a section of a page. For each link in the page, the system compares the link's location relative to the locations of the header texts to determine which section it falls in. In the topic map, links that fall under a section are associated to that section's topic.

Since the crawler preserved the order of the header texts, the topic map generator can also recognize section trees. Some sections may have sub-sections, and sub-sections themselves may have smaller sub-sections as well. In an HTML markup, core sections are usually headed by higher-level header tags (e.g. <h1>, <h2>), while sub-sections usually follows core sections with lower-level header tags (e.g. <h5>, <h6>). When examining header tags, WikiRecorder can distinguish between the different level header tags, and associate lower-level headers to the higher-level headers to preserve the hierarchical organization of the sections.

An Administrator interface is provided for instructors to add domain-dependent semantics to the topic map created by the crawler. Since the topic map generator can only analyze domain-independent hypertext structures to create a couple types of graphic organizers, the Administrator program allows instructors to add more GOs that are specific to the context of the domain. Wizards are provided to guide instructors through the GO creation process. The program is also a topic map editor as instructors are free to create their own set of topics and associations if they are familiar with the usage of topic maps, and they can also edit the topics and associations created by the crawler.

The Administrator interface can also act as a graphic organizer creator. Instructors can start with an empty domain and manually add topics and associations with the wizards or the topic map editor. [Figure 5] shows a screenshot of a graphic organizer wizard. This feature makes ENWiC applicable even outside the context of wikis as instructors can use it to build supplementary diagrams to their lecture materials.

[Figure A 1] in Appendix A shows a screenshot of the Administrator interface.


Graphic Organizer Wizard

To the hierarchy diagram, add:

A new item:

Name of concept/idea:

Description of concept (Optional):

Keywords for concept (Optional):

URL link to associated data (Optional):

[✓ Or,an existing item.]

Existing concepts/ideas: Adapter Pattern

Figure 5: A hierarchy diagram wizard from the ENWiC Administrator interface.

3.2.2 VI Implementation for ENWiC

The EnwicViewer object is the client interface. It integrates TOMU's default UI components to the ENWiC client interface, as well as creates some UI components not available in the default TOMU package, such as a browser for viewing the wiki page, a rank filter, and a history path view. EnwicViewer also uses EduNuggetsDatabaseClient object to obtain a database dump of a topic map in XTM format so that it could be used in by the TOMU visualization graph. [Figure 6] shows a segment of the XTM file produced by EnwicViewer.

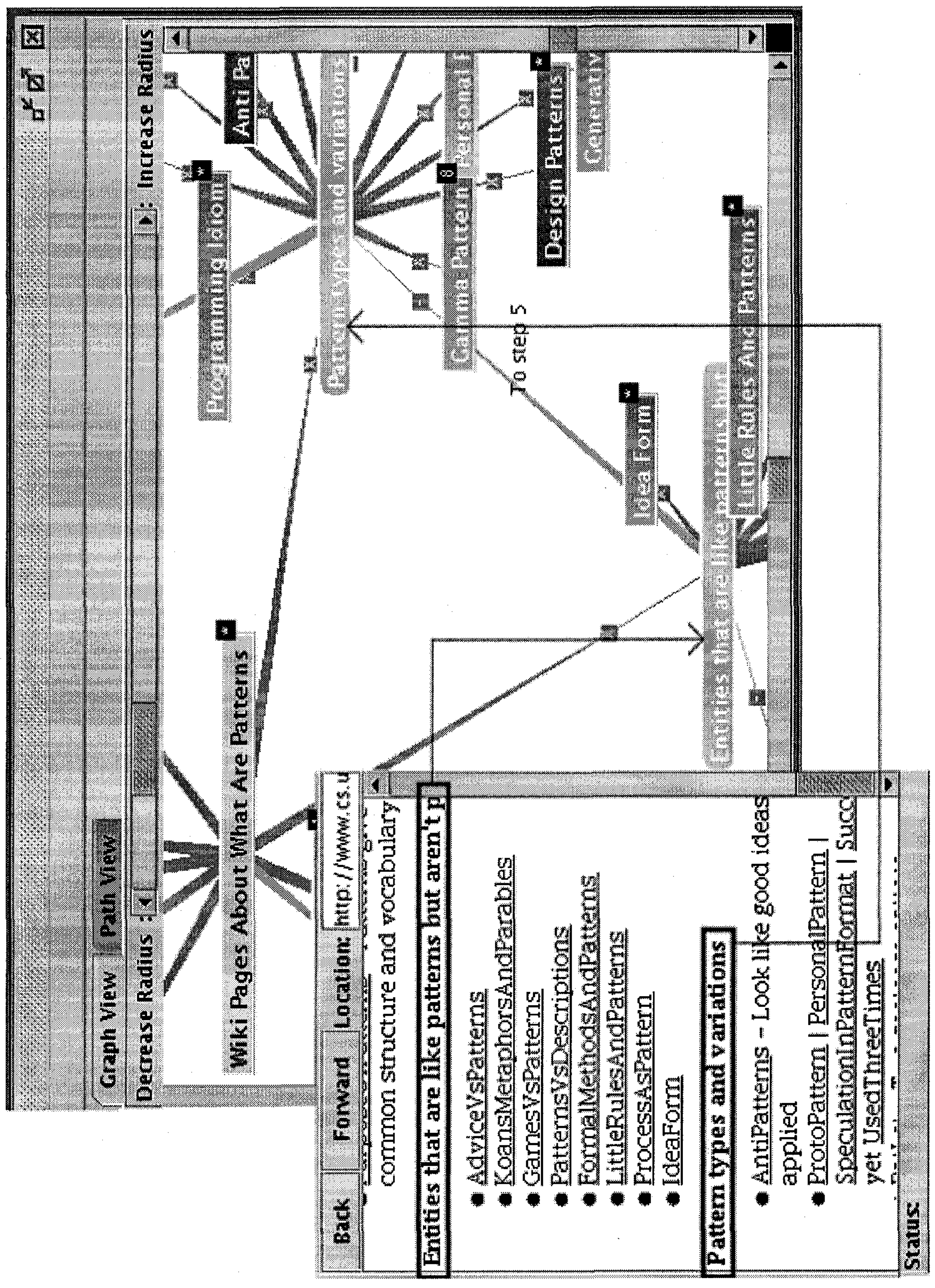
```
<topic id="t81-related_research">
<instanceOf>
<topicRef xlink:href="#t169-roomdisplayservice"/>
</instanceOf>
<baseName>
<baseNameString>
T81-SECTION: Related Research
</baseNameString>
</baseName>
</topic>

<topic id="t169-roomdisplayservice">
<instanceOf>
<topicRef xlink:href="#t148-webservices_homepage">
</instanceOf>
<baseName>
<baseNameString>
RoomDisplayService
</baseNameString>
<variant>
<variantName>
<resourceData id="index terms">
WebServices - Room Display Service
</resourceData>
<resourceData id="Nugget-6/26/2005 12:39.25+570">
text/html; charset=iso-8859-1, http://wikiURL/index.php/RoomDisplayService,
</resourceData>
</variantName>
</variant>
</baseName>
</topic>

<association id="35">

<instanceOf>
<topicRef xlink:href="#r0-linkbetween"/>
</instanceOf>

<member>
<roleSpec>
<topicRef xlink:href="#t169-roomdisplayservice"/>
</roleSpec>
</member>

<member>
<roleSpec>
<topicRef xlink:href="#t186-reservationserver"/>
</roleSpec>
</member>

</association>
```

Figure 6: Segment of an XTM file produced in ENWiC, showing a topic for a section, a topic for a page, and an association between two pages.

Like the EduNuggetsDatabase object from the topic map generation step, EduNuggetsDatabaseClient is adapted from the original EduNuggets [34] system.

Using EnwicViewer, users can load a domain available from the topic map database. We changed TOMU's text configuration files to draw nodes and edges in different shapes and colors; for example, page nodes are drawn as rectangles in a blue and purple gradient, while section nodes are drawn as round rectangles in pink. For page nodes, the intensity of the blue and purple gradient paint corresponds to the strength of their link-out rank and link-in rank respectively. Each type of graphic organizer has its own edge types, and we assigned each type its own color on the graph. This helps users identify the different GOs on the graph based on the different edge colors. [Figure 7] shows an example ENWiC visualization of the Portland Pattern Repository's wiki [55].

Users can filter the graph based on association types or ranks. Filtering by edge types allows users to concentrate on the desired types of graphic organizers. [Figure 8] shows two isolated graphic organizers obtained by filtering the edges for the node Façade Pattern. Users can also filter by ranks by specifying a rank value *r*. The users can then choose to show nodes that has a rank *r*, or higher than *r*, or lower than *r*.



common structure and vocabulary

Entities that are like patterns but aren't p

- AdviceVsPatterns
- KoansMetaphorsAndParables
- GamesVsPatterns
- PatternsVsDescriptions
- FormalMethodsAndPatterns
- LittleRulesAndPatterns
- ProcessAsPattern
- IdeaForm

Pattern types and variations

- AntiPatterns – Look like good ideas applied
- ProtoPattern | PersonalPattern | SpeculationInPatternFormat | Succ yet UsedThreeTimes

Figure 7: Portion from the ENWiC interface showing the graph and the page browser for a page in the Portland Pattern Repository's wiki. The sections detected by the crawler for the page are highlighted.

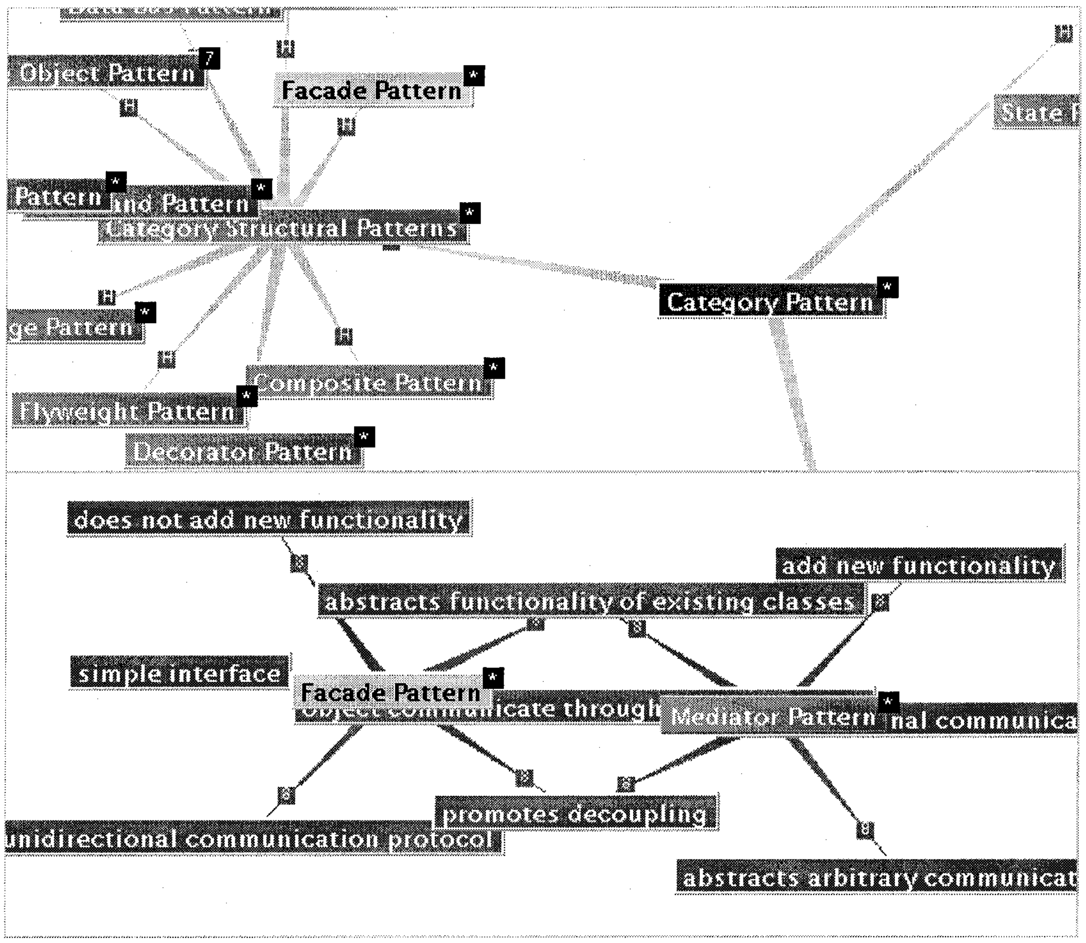
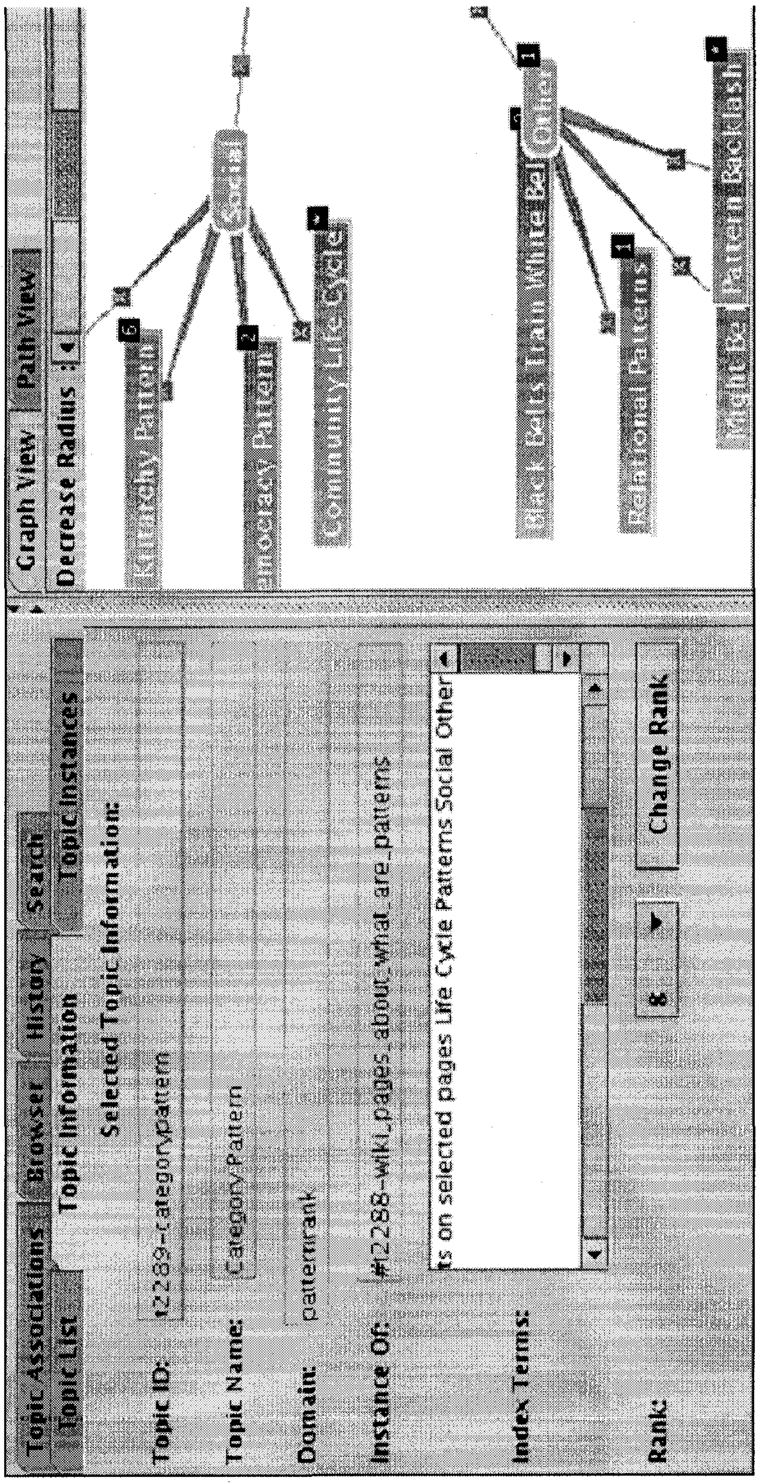


Figure 8: Two GOs related to Façade Pattern, obtained using filters. The top GO is a hierarchy diagram, and the second GO is a double cell diagram.



Selected Topic Information:

Topic ID: t2289-categorypattern

Topic Name: CategoryPattern

Domain: patternrank

Instance Of: #12288-wiki_pages_about_what_are_patterns

Index Terms: ts on selected pages Life Cycle Patterns Social Other

Rank: 8

Figure 9: The information panel alongside with the graph panel in the ENWiC interface.

Along with the visualization, the users are presented with a text information panel, which contains information such as the master list of topics, individual topic information (name, variants, associations, the corresponding wiki page, etc), a history list, and search form, as shown in [Figure 9]. The search function enables students to search topics by keywords, and they can click on a search result to focus on the corresponding node in the visualization. EnwicViewer also uses the history list to provide a visualization of the paths user has taken. This helps users recall where they are and how they arrived at certain pages. The users can also to see the diverging paths they have taken to backtrack to a previous node to take another path if necessary.

[Figure A 2] in Appendix A shows a screenshot of the full ENWiC interface.

3.3 The eNulog System

ENulog is designed to help market researchers to analyze blog data by providing them with visualizations of blog activities. To perform such analysis, the system needs to obtain a list of products of interest from an authoritative source; thus, we implemented specialized crawlers to retrieve information from source website(s). Each product is made into a topic, and when a product is mentioned in an entry, the system establishes an association between the product and the entry. The visualizations of the topic map helps market researchers identify any clusters forming around a product, and examine the blog entries associated with the product to see what the users are saying about it.

ENulog carries the similar design principles as ENWiC with modifications made in the HDR implementation to work with blogs instead; it uses multiple hypertext archive crawlers to retrieve data from blog archives and websites, and stores the information in local files that are used for further processing and analysis.

[Figure 10] shows an overview of the eNulog system, and the components for HDR and VI. We implemented the crawlers and the interface in Java, and a Perl script is used to execute the external program, Arrow [21], to perform the topic map generation process.

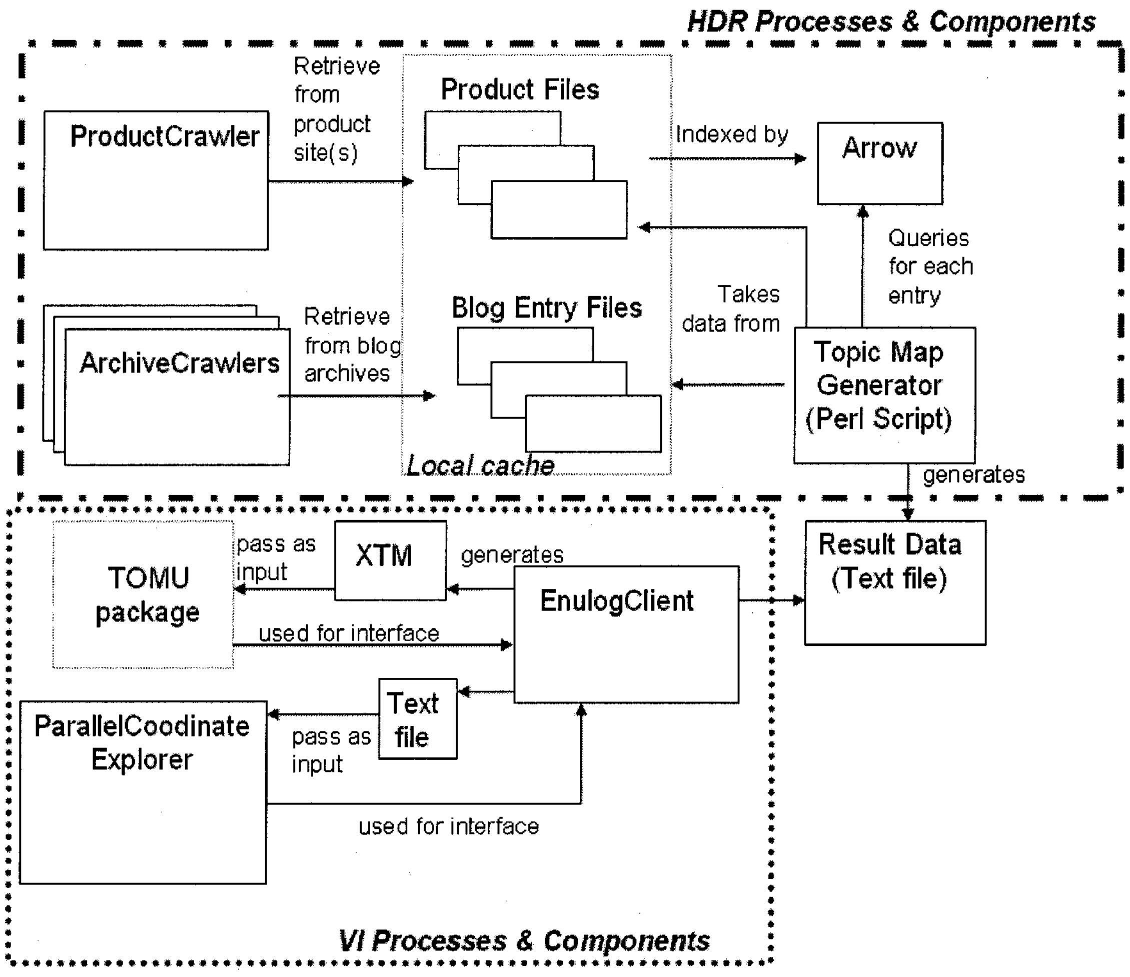


Figure 10: Architecture of eNulog illustrating how components interact with each other. Classes that are implemented for VI are outlined with a dotted rectangle; the rest of the classes are for HDR.

3.3.1 HDR Implementation for eNulog

3.3.1.1 Document Retrieval

Our crawlers for blogs are archive crawlers; we manually examined the HTML markup of each blog's chronological archive and devised regular expression matches so that the crawlers can extract entry links and text based on where they are relative to the HTML markup. Since blog archives organize entries by dates, we implemented our archive crawlers to look at the archives by months.

Using the regular expressions matches we created, an archive crawler for a blog generally performs the following steps:

1) Take a month value and a year value as user input

2) Find the list of blog entries for the specified month/year from the blog's archive. Depending on the blog software, this can usually be done by:
    a. Using the URL similar to the form http://blogURL/archive/YYYY/MM; or
    b. If each month's archive is not separated into individual pages, examine the entire archive and look for the first/last occurrence of an entry posted in that month. This serve as the start/end of the entry list for that month.
3) For each entry link found in (2), retrieve the entry page. From the page, extract
    a. *entry_title*, the title of the entry,
    b. *yyyy/mm/dd*, the date the entry is posted
4) Save the entry page locally, in */entrycache/x/yyyy/mm/dd/entry_title.html*, where *x* is the name of the blog.

To obtain products from official authoritative sources, we have to look for an authoritative website that provides a list(s) of products, along with their release dates. Like with the archive crawlers, we manually write regular expressions to match and extract product-related texts, such as the name of product and the release date, based on our examination of the website's product list page HTML markup. Based on our configurations, the product crawler generally performs the following steps:

1) Retrieve the product list(s) via a URL(s).
2) For each product in the list, if the product information is located in the product's page that is linked to in the list, retrieve the page. From the product list or the product's page, extract:
    a. The name of the product, *product_name*.
    b. The release date of the product, *YYYY/MM/DD*.

   Other fields may be extracted as well based on crawler configuration.
3) Save the extracted text in a text file locally, in */productcache/YYYY/MM/DD/product_name.txt*. Each extracted field is separated by new line characters. The name of the product is always in the first line.

3.3.1.2 Topic Map Generation

Arrow [21], a program for document retrieval, is used to identify associations between products and entries. When provided with a directory path, Arrow recursively descends into the directory to find text files; it builds an index of all the files it found in the directory. When users provide Arrow with a query, Arrow examines the file index it built and determines which file(s) is similar to the query. A heuristic value in the range of (0,1) is given for each resulting file to indicate how similar the file is to the query. eNulog takes only results that have a heuristic value higher than a threshold value. In our implementation, the threshold is set at 0.3.

Generating the topic map with Arrow involves the following basic step:

1) By giving Arrow the directory path to the product file cache, an index of the product files is built.
2) For each entry saved in the local cache:
    a. Submit the entry's text to Arrow as a query.
        i. For each result with a heuristic value greater than threshold, use the name of the corresponding product file as the topic name for the product.
            1. Extract the product release date from the directory path to the product file, and use the date as the topic name for the release date.
    b. Use the entry title as the topic name for the entry.
    c. Extract the entry post date from the directory path to the entry file, and use the date as the topic name for the post date.
    d. Associate the product topic(s) with the entry topic.
    e. Associate the entry's date with the entry topic.
    f. Associate the product's date with the product.

Treating the product/entry dates as topics and associating them with products/entries helps preserve the temporal relationships in the domain. The entire process is automated using a Perl script, which acts as the topic map generator. The script can be changed manually based on domain-dependent information to create more topics and associations.

The script writes the results in a format accepted by the Parallel Coordinate Explorer [31] as a text file. Each line of text in the file would typically contain the entry name, entry post date, related product's name, and product release date, with each parameter separated by a tab. If an entry is not related to any product, an empty string is used for the related product name parameter. Additional parameters can be added by modifying the script.

[Figure 11] shows a segment of the resulting data file from the topic map generator. Each column is separated by tabs in the text file. Date parameters prefixed with 'E' indicate they are part of the entry's post date, while date parameters prefixed with 'P' indicate they are part of the product release date. Entries are listed under the blog in which they appear.

BLOG 1	BLOG 2	EYEAR	EMONTH_DAY	PRODUCT	PYEAR	PMONTH_DAY
"news from slackerwood getting high"	""	2005	12.23	"The Alamo"	2004	4.09
"boetticher in boston"	""	2005	11.27	""	""	""
""	"haden church as sandman picture"	2005	11.06	"Venom"	2005	9.16

Figure 11: A portion of the text data file produced by eNulog's topic map generator. First row contains the parameter names. Dates are written in MM.DD format.

3.3.2 VI Implementation for eNulog

ENulog provides more views than ENWiC, since market researchers need to examine the data in different aspects, such as by time, or by volume.

The client interface for eNulog is the EnulogClient class, and it integrates both the TOMU system and the Parallel Coordinate Explorer [31] for the interface, producing three views of the data:

1) **Graph view** – Like in ENWiC, TOMU is used to visualize the topic map as a graph with nodes and edges. Modifications were made to the TOMU graph so that at startup, only entry post date and product release date nodes are shown on the graph. Each set of dates are linked together in chronological order to form two timelines. The user can then select the dates of interest on each timeline to see the product(s) released or entries posted on those dates. The user can further expand the product nodes or the entry nodes to see their relationships with other entry nodes and product nodes, respectively. Colors are used to distinguish different types of nodes; each of the two timelines gets its own color, and each blog is

assigned its own color, which is also used by all entry nodes associated with that blog. Edges are also colored based on the type of association they represent. In the graph view, users can determine popularity by observing the number of edges linked to a node; a heavily discussed product will have relatively more edges linked to it.

2) **Temperature map** – The temperature map is an extension of the graph view. Blog nodes and product nodes are drawn with different sizes and opacities. The criterions that determine the size and opacity of a node are configured manually in the TOMU graph source code. Generally, these criterions are based on the number of a specific type of associations related to the node; for example, in our configuration, the relative size of a product node is determined by the number of associations it has with entry nodes, and the relative opacity is determined by the number of blogs that contains at least one entry that relates to the product. In this configuration, the user can see two product-blog relationship metrics – the number of times the product is talked about, and the number of blogs that are interested in the product - simply by observing how the node is drawn. We also created another temperature map that focuses more on blog activity; each node in this view represents one day's activity of one blog, and the relative size of the node indicates the number of entries made on that blog within that day. With this view, users can observe the update activities of each blog. Nodes are not labeled in this view, but user can hover over a node to see a tooltip with the node information. Initially, all edges are hidden in this view so that users can focus solely on the nodes. Like the graph view, the node can be expanded to show any related associations and nodes.

3) **Parallel slider view** – This view is produced using the parallel coordinate explorer [31]. The data file from our topic map generator organizes parameters in the following order: entry -> entry posted date -> related product -> product release date. The parallel coordinate slider treats each parameter as an axis and places the values from the tuples onto the axes. Each individual tuple is then represented by a line graph connecting its respective values across the parallel axes. Users can select a range(s) on any axis to focus on the set of lines that fall

into that range. This helps users establish a set of criterions to obtain desired results. The parallel axes allow users to focus on different dimensions; for example, they can alter the range of values on the product axis to focus on a certain product and its relationship across different dimensions. This is useful for market researchers as they can visually specify the parameters they are interested in to observe for any patterns or clusters in the data.

The parallel coordinate slider uses the text data file from the topic map generator directly, while EnulogClient produces the XTM file for the graph view and the temperature map. EnulogClient takes each line from the text data file and makes each relevant parameter as a topic (e.g. an entry parameter becomes an entry topic, and a product parameter becomes a product topic). It establishes associations between topics from the same tuple, e.g. between the entry topic to entry post date. The EnulogClient also detects and discards repeated values so that the topic map would not have duplicated topics; for example, if there are multiple entries posted on the same date, the date appears in multiple tuples. The EnulogClient makes only one topic for the date, and associates all entries to that one date topic.

The graph view allows users to see an overview of what is happening in the domain. The temperature map allows users to quickly identify the volume of activities surrounding a product/blog. The slider view allows users to see patterns and trends that occur through time.

The slider view and the temperature map are shown in [Figure A 3] and [Figure A 4] in Appendix A, respectively.

3.4 The Annoki System

While HDR enables the systems to obtain a fairly good topic map representation of the resource domain, we believe that we can improve the quality of the topic map even further if we have control over the implementation of the domain software. Along with the ability to incorporate better semantics to the domain software, we also have the ability to create a live feed from the domain to the client interface so that users always have up-to-date topic map visualization.

Annoki is developed to work in such scenario, and it integrates the LTM component and the VI component. On the domain end of the system, we have our Annoki wiki, which is an enhanced version of MediaWiki [53] with extra semantics and functionalities. The topic map generation process involves exporting the wiki database into a topic map, and the client interface uses this topic map to produce the visualization.

Having a synchronous topic map interface with the wiki can be beneficial with respect to content management. When users are planning to contribute new content to the wiki, the topic map can serve as a guide to the domain to help them decide where their new content fit in. In a conventional wiki, users can write a new page, but they may not be aware of what intra-wiki links they could establish for related topics. The topic map can help them visually identify any clusters of related topics that are currently present in the wiki that they can associate the new content with; thus, when using the topic map as a guide, the users can potentially reduce the number of orphaned pages or sparsely linked page that should not be orphaned or sparsely linked. This leads to better content authoring and management scenarios.

The Annoki wiki database provides data that are not immediately apparent in the wiki to the topic map, such as information that pertains to user activities. Since the database can provide a list of registered users and the revisions each user has made, users are included in the topic map as topics, and associations are made between a user and the pages he/she has edited. When the topic map is visualized, users can see if there are patterns in each user's activity, and identify the possible subjects that each user may be specialized on, based on any clusters of related pages that appear in the user's edit history.

A simple and intuitive metadata for a wiki page is the property/value pair. In Annoki, users can associate a wiki page with a set of property/value pairs of their choice to categorize the pages better. In the topic map, users can then first look for a desired property, and then they can identify all possible values for that property alongside with the pages that contains those values. Thus, the property/value pairs help create hierarchical-like structures in the topic map, which helps users further understand the relationships of the topics in the domain.

The property/value pairs are also used to improve the interface of the wiki itself in order to help users locate information better. Associated property/value pairs are displayed on a wiki page, and when users click on a pair, all associated pages for that property/value pair are displayed to let users see the related pages. Users can also search for specific property/value pairs to locate associated pages.

One of the biggest criticism of wikis is that the ability for any users to edit pages often lead to the submission of incorrect or inappropriate content. To help correct this problem, Annoki introduces an annotation layer in the editing process, which treats all edits as pending edits until they receive approval to become permanent. The approval can be given by administrators, but to preserve the collaborative nature of wikis, Annoki also allows users to give the approval. The intuition behind the use of the annotation layer is that edits that are deemed useful by the majority of the users should become permanent, while edits that are considered unhelpful by most users should be removed to save future readers from taking the time to go through it. By allowing users to rate edits, Annoki can encourage stronger collaborative content management where users work together to keep unsuitable content apart from the official revisions of a page, so that official revisions only contains useful and credible information.

[Figure 12] summarizes the structure of Annoki and how the LTM and VI components are integrated into the system.

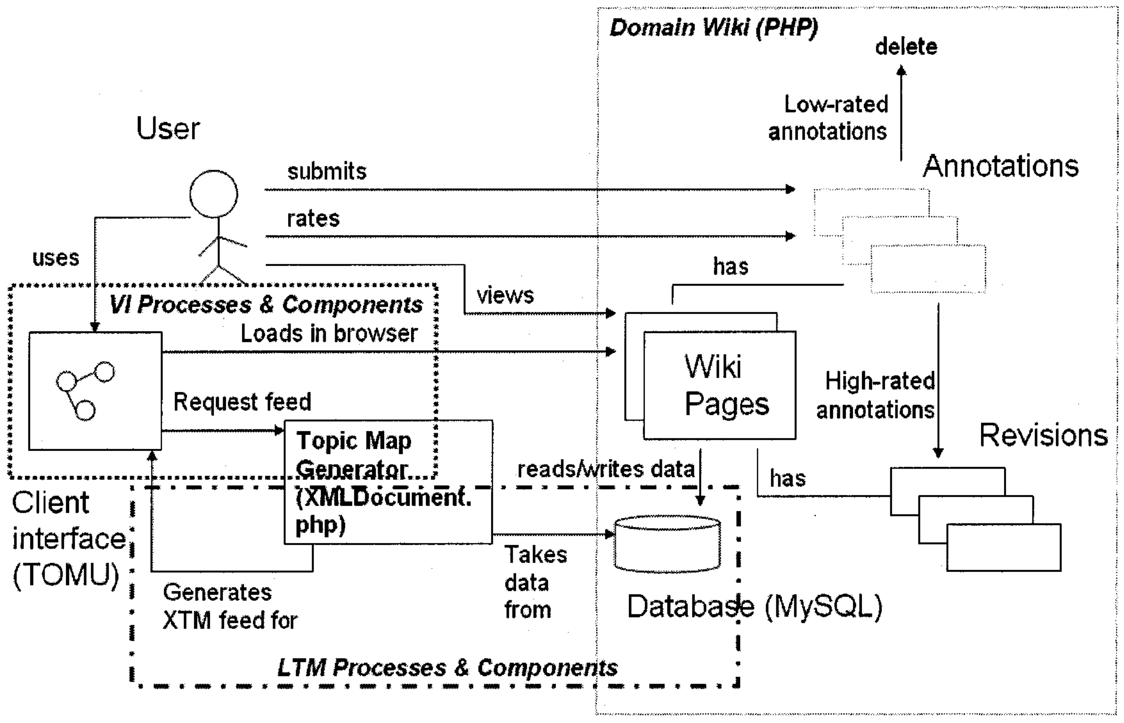


Figure 12: The Annoki Architecture.

3.4.1 LTM Implementation for Annoki

To generate the topic map, Annoki translates the tables and entities in the SQL database to topics and associations that convey the same information and structure. Since MediaWiki [53] was used as the base of Annoki wiki, all of its tables are carried over. Additional tables are used to store Annoki-specific information.

The tables from the original MediaWiki that are of interest in the topic map generation process are:

1) **Page** – This table stores the properties for each wiki page, such as page ID, page namespace, title, current revision, and hit counter. Each page is made into a topic in the topic map, using the page ID as the topic ID, and the page title as the base name of the topic.
2) **Revision** – Each revision made to the wiki is stored in this table with a foreign key that refers to the ID of the page the revision is for.
3) **Pagelinks** – Every link between two wiki pages is stored in this table. In the topic map, each link from this table is interpreted as an association between the corresponding page topics.

4) **User** – This table stores information on all registered users of the wiki, such as user ID, username, and email. Each user is represented as a topic in the topic map. If a user made changes to a page, which can be determined by examining the Revision table and the Annotation table, an association is established between the user and the page.

Additional tables made specifically for Annoki are:

1) **Annotation** – This table is similar to the Revision table, with extra fields added to store additional properties that are specific to annotations only. These fields include a hit counter, and a heuristic value that helps determine if an annotation should be promoted. Each annotation becomes a topic in the topic map. All page edits are stored in the Annotation table until they obtained a score high enough to warrant a promotion. When the annotation is promoted, its data is moved to the Revision table.
2) **Annotation_User_Rank** – This table stores the votes registered users gave to each annotation. A user can rank an annotation as Useful, Average, or Not Useful. The votes are used to calculate a heuristic value for each annotation.
3) **Annotation_Association** – This table keeps track of the annotation tree for each page by storing the parent of each annotation. If a user makes a new annotation by editing a page revision, the parent of the new annotation is the page. If a user makes a new annotation by editing an existing annotation, the new annotation's parent is the existing annotation. All parent-child relationships are interpreted as associations in the topic map to preserve the structure of the annotation tree.
4) **KeyValue** – This table stores unique property/value pairs that can be found in the wiki. Each property/value pair is made into a topic in the map.
5) **Page_KeyValue_Pairs** – This table lists relationships between a page and a property/value pair. They are made into associations in the map.

We implemented an XMLDocument object in PHP on the Annoki wiki to handle the topic map generation.

3.4.2 VI Implementation for Annoki

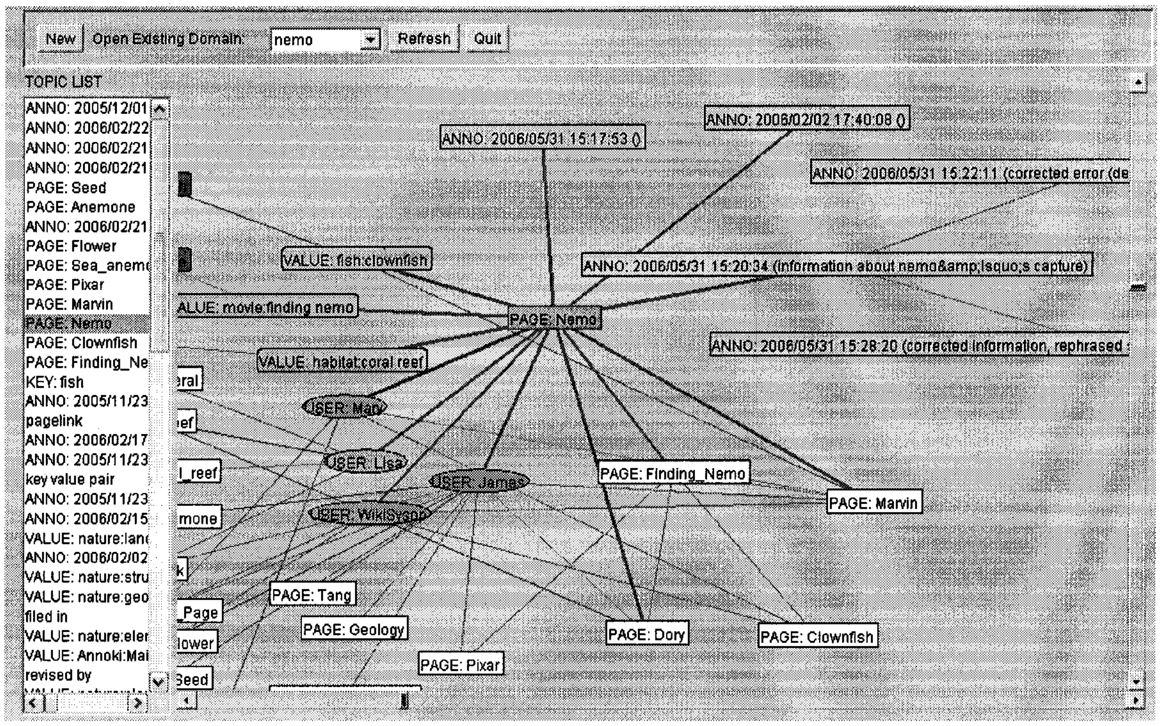


Figure 13: The Annoki Visualization interface.

The client interface of Annoki is produced using TOMU. An AnnokiClient object is implemented to integrate all UI components from TOMU, which allows users to navigate the wiki with the graph, and control the look of the graph using the filter and zoom tools. [Figure 13] and [Figure A 5] from Appendix A show two different topic maps as seen in the Annoki visualization interface.

We implemented an extended version of the TOMU file parser, named AnnokiFeedParser, which retrieves an XTM feed from Annoki first to put it through the default TOMU XTM file parser. The feed is located at http://annokiURL/index.php?action=xml. Requesting for the feed triggers the topic map generation process, so that an XTM of the wiki can be produced. [Figure 14] shows a segment of the feed.

```
<topic id="pg1283">
<baseName>
<baseNameString>PAGE: Marvin</baseNameString>
```

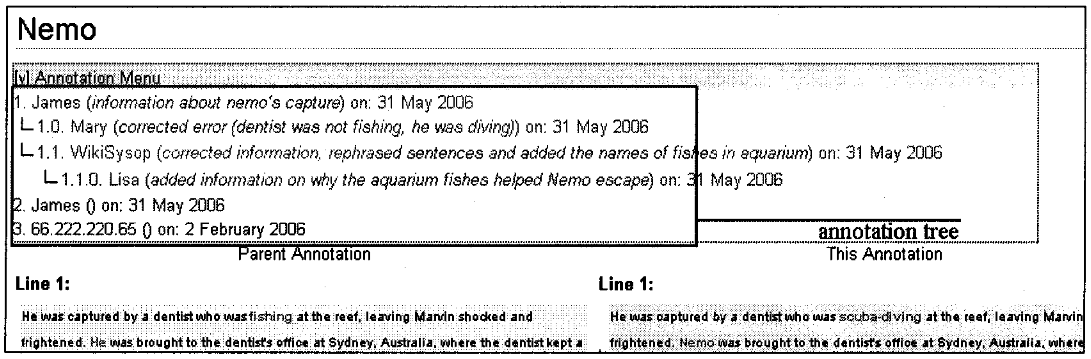
```
<variant><variantName>
<resourceData id="Nugget">text/html,
http://www.annokiURL/index.php/Marvin,
</resourceData></variantName></variant>

<variant><variantName>
<resourceData id="RevisionAnnotationCounters">RevAnn: 5: 3</resourceData>
</variantName></variant>

<variant><variantName>
<resourceData id="HitCounter">Hits: 71</resourceData>
</variantName></variant>

</baseName>
</topic>

<topic id="usr2">
<baseName>
<baseNameString>USER: Mary</baseNameString>
<variant><variantName>
<resourceData id="Nugget">text/html,
http://www.annokiURL/index.php/User:Mary,
</resourceData></variantName></variant>
</baseName>
</topic>

<association id="a48">
<instanceOf><topicRef xlink:href="#at-revby"/></instanceOf>
<member><roleSpec><topicRef xlink:href="#pg1283"/></roleSpec></member>
<member><roleSpec><topicRef xlink:href="#usr2"/></roleSpec></member>
</association>
```

Figure 14: Segment of the XTM file produced by the topic map generator in Annoki, showing a topic of a page (Marvin), a topic for a user (Mary), and an association that indicates Mary made revisions to Marvin.

3.4.3 The Annoki wiki

Along with the features from the original MediaWiki [53] interface, Annoki wiki users are presented with additional menus and forms that mainly deal with annotations or property/value pairs. Administrators are provided with additional forms to moderate the wiki.

Like a regular wiki, the users see the latest revision of a page by default. They can choose the annotation they like to view by clicking on the corresponding link on the annotation tree. To allow users to easily identify the edits made in an annotation, when viewing an annotation, a differencing table is displayed to show the changes between the annotation being viewed and its parent. [Figure 15] shows the annotation menu on the Annoki interface. Users can rate the annotation using a voting form that appears at the end of the page. Annoki administrators are provided with an additional form, which enables them to promote or delete an annotation. When an annotation has been promoted to a revision, the administrators are also allowed to demote the revision back to an annotation if they deemed necessary. [Figure 16] shows the property/value pairs and the rating forms as seen on an Annoki wiki page.



Nemo

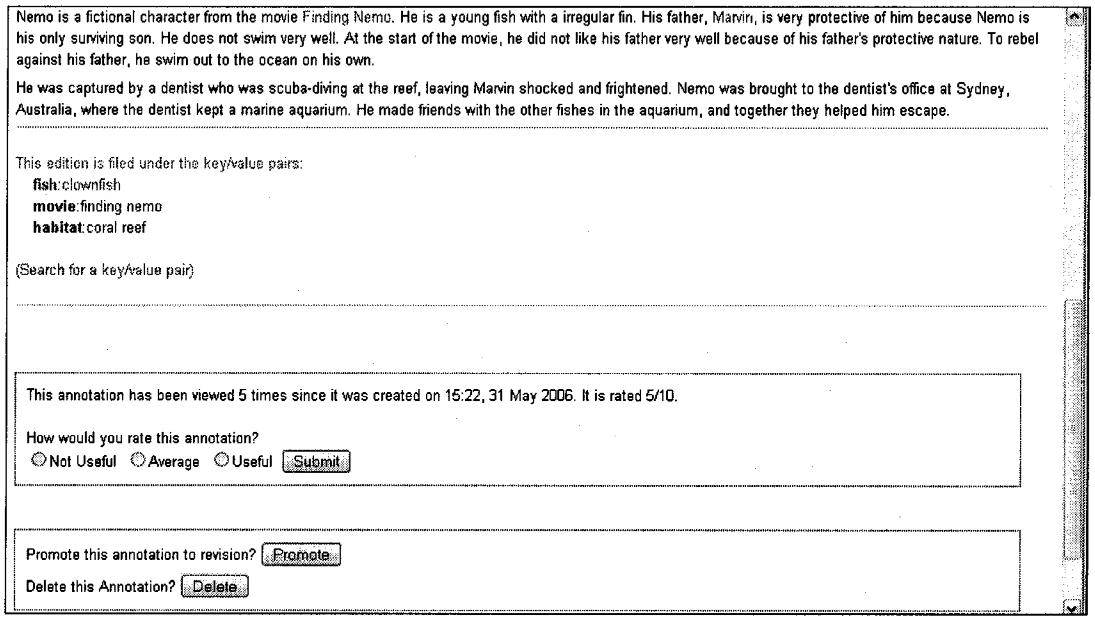
[v] Annotation Menu

1. James (*information about nemo's capture*) on: 31 May 2006
 └ 1.0. Mary (*corrected error (dentist was not fishing, he was diving)*) on: 31 May 2006
 └ 1.1. WikiSysop (*corrected information, rephrased sentences and added the names of fishes in aquarium*) on: 31 May 2006
  └ 1.1.0. Lisa (*added information on why the aquarium fishes helped Nemo escape*) on: 31 May 2006
2. James () on: 31 May 2006
3. 66.222.220.65 () on: 2 February 2006

annotation tree

Parent Annotation	This Annotation
Line 1:	**Line 1:**
He was captured by a dentist who was fishing at the reef, leaving Marvin shocked and frightened. He was brought to the dentist's office at Sydney, Australia, where the dentist kept a	He was captured by a dentist who was scuba-diving at the reef, leaving Marvin frightened. Nemo was brought to the dentist's office at Sydney, Australia, where

Figure 15: The annotation menu on the Annoki interface, with a differencing table following the menu.



Nemo is a fictional character from the movie Finding Nemo. He is a young fish with a irregular fin. His father, Marvin, is very protective of him because Nemo is his only surviving son. He does not swim very well. At the start of the movie, he did not like his father very well because of his father's protective nature. To rebel against his father, he swim out to the ocean on his own.

He was captured by a dentist who was scuba-diving at the reef, leaving Marvin shocked and frightened. Nemo was brought to the dentist's office at Sydney, Australia, where the dentist kept a marine aquarium. He made friends with the other fishes in the aquarium, and together they helped him escape.

This edition is filed under the key/value pairs:
fish:clownfish
movie:finding nemo
habitat:coral reef

(Search for a key/value pair)

This annotation has been viewed 5 times since it was created on 15:22, 31 May 2006. It is rated 5/10.

How would you rate this annotation?
○ Not Useful ○ Average ○ Useful [Submit]

Promote this annotation to revision? [Promote]
Delete this Annotation? [Delete]

Figure 16: A segment of an Annoki page showing the body text, the property/value pairs, and the annotation rating form.

When editing a page, users are presented with a form to add or remove property/value pairs. When a property/value pair is submitted with the page edits, it is listed on the page after the body text. However, like with all edits, if a property/value pair is added or removed in an annotation that is never promoted, the change does not appear on the official revision of the page.

A search form is provided to let users search for property/value pairs by entering a keyword for the property field or the value field, which returns pairs that matches or partially matches the keyword. If users enter the entire property/value pair in the search, Annoki returns page results instead, which are divided into three categories: pages that are currently associated with the property/value pair, pages with an old revision that is associated with the pair, and pages with an annotation that is associated with the pair.

Administrators can restrict access to certain pages by specifying user groups that are allowed to view and edit these pages. Administrators can set up user groups using the special purpose page provided by the original MediaWiki [53].

[Figure A 6] in Appendix A shows an annotated wiki page as it appears in the Annoki wiki.

3.4.3.1 The annotation life cycle

When a user rates an annotation, a heuristic value for the annotation is calculated to determine if it gained enough approval to award a promotion. The heuristic value h is determined with the weighted formula:

$$w1*u + w2*a + w3*n + w4*r = h \quad \textbf{[Eq. 1]}$$

where u is the percentage of 'Useful' votes received out of all votes cast for the page, a is the percentage of 'Average' votes received, n is the percentage of 'Not Useful' votes received, and r is the ratio of views the annotation received relative to the highest viewed annotation in the wiki. The weights w_1, w_2, w_3, w_4 are constant values that can be changed in the PHP file they reside in during installation. By default, $w_1 > w_2 > w_3$, to ensure higher scores are given to annotations that received more 'Useful' votes than 'Not Useful' votes. The importance of the hit ratio can be altered by changing w_4; the intuition of adding the hit ratio to the formula is to compensate for cases where users do not vote enough regularly. If users are constantly referring back to the same annotation, which is signified by a high number of views, it implies that the annotation contains useful information; thus, it is likely that it deserves a promotion.

After the calculation is performed, if the annotation is in its 'trial period', the h value is normalized and then compared to a threshold value t. If $h > t$, the annotation receives a promotion. The threshold value and the start and end of the trial period can be changed in the same PHP file where the weight values are specified. The trial period is implemented to avoid the accumulation of annotations over a long period of time, and to prevent an annotation from being promoted prematurely. An annotation should not be promoted before a reasonable number of users had a chance to review and rate it. Conversely, if an annotation has been around for a long time without receiving a promotion, then it is a sign that the annotation has been viewed by most users and that they probably did not find the annotation useful enough to grant a promotion. The start and end of the trial period is determined by one of the following three variables *i, d,* and *v*, where:

1) *i* is the number of views the annotation received. A low number suggests that not many users have a chance to see the annotation, and thus it should not be promoted just yet. A high number suggests that most users have already seen the annotation, and if it has not earned enough support for a promotion, it is likely not useful.
2) *d* is the number of days elapsed since the annotation was created. The system does not want to promote an annotation that has only been around for a very short period of time (e.g. an hour), nor does the system want to consider an annotation that is very old, as the information on the annotation may no longer be up-to-date. An annotation that has been around for a long time without getting a promotion also suggests that the annotation is no longer in used.
3) *v* is the number of votes it received. If only a few percentages of users have rated the annotation then the system should wait for more votes. If most users have already rated the annotation then the system has sufficient information to determine if the community found the annotation useful.

Only one of the variables must be satisfied to trigger the start and end of the trial period. For *v* and *i*, what constitute as a 'high' number a 'low' number depend on the size of the community, thus when users define the start and end of the trial period for these variables in the PHP file, the values they enter are scaled by the number of registered users:

- in terms of *v*, users are asked to specify a percentage of registered users that needs to vote to start/end the trial; for example, if *v* is set at 0.5 for the start of trial and there are 4 registered users, the start trial condition for *v* is satisfied when the annotation receives 2 votes.
- in terms of *i*, users are asked to specify the average number of times each registered user need to view the annotation to start/end the trial; for example, if

the *i* value is set at 100 for end of trial and there are 4 registered users, when an annotation receives 400 views, the end trial condition for *i* is satisfied. When setting the start/end trial condition for *i*, one must keep in mind that if anonymous users are allowed to view the wiki, they affect the hit counters as well, and thus *i* should be set at a higher value based on the amount of traffic the wiki is expected to receive outside of the wiki.

When setting the start/end values for *d*, one must consider the importance of time to the domain of the wiki; if the wiki contains information on current or rapidly changing events like the news, then the *d* values should be small enough to reflect the shorter life cycle of the information. Administrators have the power to control the life cycle of annotations by promoting or deleting them at any time when they deem necessary.

Chapter 4: Empirical Evaluation

This section discusses the experiments and case studies we carried out using ENWiC, eNulog and Annoki. The usability of ENWiC was tested against a conventional wiki with an experiment of how well subjects performed tasks on each interface. A case study in the movie domain was performed with eNulog to examine whether the amount of blog activities surrounding a movie is related to its success. Finally, a pilot study was conducted to examine how well Annoki can assist young students in storytelling activities.

4.1 ENWiC usability experiment

ENWiC was designed to help learners access and understand wiki content better. To test its effectiveness, a usability experiment was conducted to determine if students were able to find information faster with ENWiC. Students were given questions that they must find the answers to from wiki pages; they were asked to use a normal wiki interface for some questions, and the ENWiC interface for the rest. Their performances on each interface were used to evaluate which interface was more effective in helping them locate the necessary information.

4.1.1 Experiment Design

In this experiment, our objective was to evaluate the following two hypotheses:

- Hypothesis 1: the visualization of the overall structure of an information-dissemination wiki enables *faster* access to the wiki content.
- Hypothesis 2: the visualization of the overall structure of an information-dissemination wiki enables *better* access to the wiki content.

The ENWiC experiment was setup as an across groups experiment:

- In the *control condition*, the subjects used a conventional wiki on its own to answer questions based on information they could find in it;
- In the *experimental condition,* the subjects used the ENWiC interface to the same wiki to answer the questions.

The subjects were divided equally into two groups: one started off in the control condition to perform three tasks, and then switched to the experimental condition to perform the remaining tasks; vice versa for the other group. The subjects were third or fourth year computing science students, and the problems they were given involve design patterns concepts. The Portland Pattern Repository's wiki [55] was used because they have a 'road map' page which links to design patterns-related pages on the Wiki. We used ENWiC to crawl the road map page to obtain a topic map of the design pattern pages. We then used the Administrator program to add in hierarchy diagrams, double cell diagrams and comparison matrices to reflect relationships that are specific to the design-pattern domain.

The subjects were given a short tutorial on how to use the wiki interface and the ENWiC interface. Under each condition, they answered one easy question involving one design pattern, and two harder questions that involved the knowledge and/or analysis of two or more patterns. The subjects were given a maximum of fifteen minutes for each problem. [Table 1] contains the questions the subjects were given.

Table 1: The ENWiC experiment tasks

easy	1. Give an example of how credit card service can use the Strategy pattern.
	2. Consider a chat room application. What is the subject and what are the observers?
difficult	3. What is the class of design pattern that Decorator belongs to? What is the class of design pattern that Memento belongs to?
	4. List two patterns that promote decoupling by using a central object for communication.
	5. Consider a pager service. Why is the façade pattern inadequate for this scenario? Which pattern would be more suitable?
	6. You would like to make a deposit to your personal accounts at the bank. The bank has a group of tellers; you are called to the teller that can handle deposit to personal accounts, who will process your request. What pattern does this resemble?

For each question, we recorded the following metrics for each subject:

1) time to completion;
2) number of unique pages visited;
3) number of mouse clicks on the interface (on nodes/buttons for ENWiC, and on links and back/forward buttons for the Wiki); and

4) correctness and completeness of the answer.

To determine the users' experience and impression of ENWiC and the Wiki, we asked the subjects to fill out a user-satisfaction questionnaire at the end of the experiment.

4.1.2 Results and analysis

We found that the "time to answer" variable depended more on the subjects than the interfaces. Subjects who could read and comprehend the information quicker got shorter completion times regardless of the interfaces when compared to other subjects. On average, subjects did achieve shorter completion times with ENWiC; the average time used per question on ENWiC is 3:17 minutes, while the average time used per question on the wiki is 3:31 minutes. Since ENWiC's loading time per page at the time of the experiment was noticeably longer than the Wiki, the completion time could potentially be higher.

Table 2: Number of mouse clicks and unique pages used by each subject (A-F) for each question (1-6).

	1	**2**	**3**	**4**	**5**	**6**
A	5,E:3,2	2,W:5,5	1,W:8,6	4,E:1,1	6,E:7,3	3,W:43,26
C	2,W:4,2	5,E:3,3	4,E:5,3	1,W:40,14	3,W:33,17	6,E:3,2
E	2,W:2,2	5,E:3,3	4,E:4,4	~~1,W:24,7~~	~~3,W:9,4~~	6,E:2,2
B	5,W:4,3	2,E:1,1	1,E:8,7	~~4,W:19,16~~	~~6,W:29,18~~	3,E:6,6
D	2,E:3,3	5,W:2,2	4,W:10,6	1,E:3,3	3,E:3,3	6,W:9,7
F	2,E:4,3	5,W:2,2	**4,W:2,2**	1,E:13,8	3,E:4,4	6,W:2,2

We also examined the number of mouse clicks made by the subjects to complete each task. [Table 2] summarizes the activity of the subjects. Each cell is written in the format (O,C:M,P), where O represents the order the subjects answered the question in, C is the tool used to answer the question (E=ENWiC, W=Wiki), M is the number of mouse clicks, and P is the number of unique page visited. From the table, we could see that subjects used significantly less number of mouse clicks on ENWiC for the difficult questions. The number of clicks was not a factor with the easier questions as the answers can be found on a single page; however, when subjects have to combine information from different pages to solve the problem, the ENWiC interface eliminated the need for subjects to click on back buttons multiple times to re-visit links to different pages, as they could just click on a graph node or use the history graph to go directly to the desired page. The number of unique pages used by the subjects also decreased when they used the ENWiC interface for the harder questions. From our observations, we found that ENWiC reduced the amount of content that the subjects needed to sift through to find potential solutions. For example, when answering question (5), after the subjects searched for Façade pattern on ENWiC, they could see from the graph that the Mediator pattern shared some associations with the Façade pattern, and thus they realized that the Mediator pattern may be the answer. For wiki users, the relationship between Façade and Mediator patterns were not as apparent, and thus they had to read about each pattern from the Patterns Category page until they reached the Mediator pattern page to realize it contained the answer. This increased the number of mouse clicks and the number of unique pages used by the subjects, which indicated that the subjects went through more irrelevant information than ENWiC users to reach the answer.

We found that subjects were also able to recognize 'good' links better with ENWiC; for example, in question (1) where subjects were asked to write an example of the strategy pattern, the answer can be found by visiting a link at the bottom of the Strategy Pattern wiki page, which led to a page that has an 'Example' section about strategy patterns. On ENWiC, the subjects could quickly identify a node named 'Example' that is associated with the Strategy Pattern node to find that page, while on the Wiki, the subjects had to read the entire page until they see the link at the bottom of the page, and some subjects did not think of visiting the links at all. This also increased the reading times for some subjects, as they had to read more text to identify good links to other pages on the Wiki.

We used t-statistics to evaluate whether the trend we observed was statistically significant. We believed that the number of mouse clicks used on ENWiC is less than on the Wiki. Thus, we set up our hypotheses as:

$$H_0 : \mu_{Wiki} = \mu_{ENWiC} \text{ (Null Hypothesis) } \textbf{[Eq. 2]}$$

$$H_1 : \mu_{Wiki} > \mu_{ENWiC} \text{ (Alternate Hypothesis) } \textbf{[Eq. 3]}$$

$$\text{Selected significance level: } t_{0.05,4} = 2.132 \textbf{ [Eq. 4]}$$

Table 3: ENWiC hypothesis testing result for each task. A value marked with * denotes it is significant enough to reject the null hypothesis.

Question	T_0 (# of clicks)	T_0 (# of pages)
1	0	-0.707
2	0.555	0.5547
3	0.372	0
4	2.997*	2.428*
5	2.526*	2.138*
6	1.127	1.121

[Table 3] shows the hypothesis testing result for each question. Since questions (1) and (2) were easy questions, we did not expect any significant decreases. For the difficult questions, we concluded that the number of mouse clicks used for questions (4) and (5) were significantly less with ENWiC. The Patterns Category page on the wiki helped subjects with locating the answer for question (3) and thus the decrease in clicks were not as significant. For question (6), we found that most subjects were able to recall some of the information they have seen earlier on the Wiki, so they were able to locate the necessary pages faster through memory.

We carried out another hypothesis testing to show that the number of clicks needed to complete all tasks was less in ENWiC than in the Wiki. Our hypotheses for this test are:

$H_0 : \mu_{Wiki} = \mu_{ENWiC}$ (Null Hypothesis) **[Eq. 5]**

$H_1: \mu_{Wiki} > \mu_{ENWiC}$ (Alternate Hypothesis) **[Eq. 6]**

Selected significance level: $t_{0.05,34} = 2.032$ **[Eq. 7]**

[Table 4] shows the result for this test, which supports our hypothesis that subjects can complete the tasks in ENWiC with less effort.

Table 4: Hypothesis testing result for all tasks. A value marked with * denotes it is significant enough to reject the null hypothesis.

Question	T_0 (# of clicks)	T_0 (# of pages)
1-6	2.611*	1.790

Since the ENWiC interface is considerably different from conventional web interfaces, we found that some subjects needed time to adjust and learn the interface. Some subjects were able to adapt to the interface quickly and utilized the graph to find the answers, while a couple of subjects were not able to learn as quickly, and ended up relying on text searches to find their answers. Since the experiment was short in duration, the subjects who preferred text searches did not have time to familiarize themselves with the ENWiC environment, such as what types of graphic organizers were available and how the GO filtering function worked. These subjects did not achieve a notably better performance in ENWiC, as opposed to subjects who were able to work with the ENWiC graphs to discover information faster. The subjects who used the ENWiC graphs also appreciate the fact that the ENWiC search results showed them a 'neighborhood' of related pages on the graph, as opposed to a single discrete page from a text search result.

The correctness and completeness of the tasks were not significantly different under the two experimental conditions; this result falsifies our second hypothesis that subjects can complete the tasks in ENWiC better. This is not very surprising, however, since ENWiC nodes refer to information from the Wiki. Once the subjects found the relevant pages, the information they obtain is identical under either condition, and thus they arrived at the same answer. The difference between the two conditions lies in how the subjects find the information, and not what information the subjects found.

From the subjects' responses in the questionnaires, all subjects thought ENWiC was an interesting tool to use, and it helped highlight the relationships between different design patterns. They did not feel as lost when using ENWiC, and they were able to identify paths to relevant pages better when using the ENWiC graphs. The main disadvantage they saw with ENWiC is its loading time; while the transition between pages on the wiki is almost instantaneous, on ENWiC, every time the subjects focus on a new node, a graph for the node has to be retrieved and drawn on screen, which made the subjects notice that the program is slower than the Wiki. A few subjects who are used to switch between pages quickly on an Internet browser found the loading time to be a hindrance that made the program not as satisfying to use.

ENWiC is designed to be a visual learning tool and our experimental results indicated that it is beneficial to wiki users. We believe that this is the case because it engages the visual intelligence of the wiki users in addition to their verbal intelligence, however, a subsequent experiment, differentiating between these two types of learners, would be necessary to investigate this hypothesis.

4.2 The movie case study using eNulog

The information disseminated through blogs tends to be more "opinionated" than the information contained in Wikis. There are a multitude of politics blogs with commentary and opinions on recent events and political personalities and blogs on products and services. This fact has led marketing analysts to view blogs as an obvious marketplace to survey for recognizing trends in the consumers' opinions and preferences.

4.2.1 Case-study Design

In this case study, we wanted to investigate whether blog 'buzz' reflects the success of a product. The movie domain is especially interesting for this investigation because a movie's popularity is highly dependent on 'word-of-mouth' advertising. People tend to share opinions regarding the latest movies, and a person is more likely to watch a movie if his/her friends gave good comments on the movie. Blogs have helped widen people's scope, as they can now turn to blogs and collect many opinions about a movie to help them decide whether they should watch the movie or not. Movies are released on a weekly basis, which makes the domain even more interesting on a research perspective, as data are constantly being updated regularly, giving us a large set of varying data over a timeframe.

Our overall objective was to examine whether blog activities correlate to movie success and how. Furthermore, to the extent that some aspects of blog activities correlate with movie success, we were interested in examining which eNulog visualizations can better communicate these correlations visually.

We conducted a study that investigated the correlations between the information and the activities in several movie-related blogs, and the movies released during November and December 2005. Our 'product' in this study is movies, and we used IMDB [49] as the authoritative source of official movie-release data. At the time of study, IMDB provided a monthly release schedule from 2003 to 2006 (accessible via http://imdb.com/nowplaying/YYYY/MM/), and the product crawler in eNulog was implemented to retrieve the monthly schedule and extract:

- the title of the movie

- the release date of the movie

To determine movie success, the crawler also visited the individual movie pages through the links provided in the monthly schedule to look for the 'Box office & business' page for the movie. From this box office page, the crawler extracted:

- the opening weekend gross
- the three-weeks cumulative gross

Next, we implemented five archive crawlers for eNulog to extract all blog entries and their associated comments posted in the months of November and December 2005 from five regularly updated movie-related blogs: Cinematical [46], The Movie Blog [61], Cinemarati [45], World Mag Movie Blog [70], and Cinecultist [44].

After the eNulog topic map generator created the topic map by relating the entries and the movies together, we retrieved the following types of data about the blog activity related to a particular movie:

- the number of entries posted about a movie, which is the number of associations between the movie and blog entries on the topic map;
- the *lifespan* of the movie in the blog; i.e. the number of days between the first and the last posted entry about the movie, which was found using the Slider view;
- the *posting regularity*, i.e. the density of entries posted about a movie. For our analysis, we used the average number of posts over a movie's lifespan, and the average number of posts over a fixed time period. We chose four weeks as our fixed time period, which spans over two weeks before a movie's release date, and two weeks after the release.

We also examined the different effects of blog activity before and after a movie's release by running two sets of calculations: one for the blog activity of a movie before its release, and the other for the total blog activity of the movie before and after its release.

4.2.2 Study results and statistical analysis

The three views in eNulog helped viewers identify the 'popular' movies in the domain. [Figure 17] shows the graph view for a box office hit, *Harry Potter and the Goblet of Fire*, versus *Casanova*, which was not as successful. The difference in the number of edges clearly showed Harry Potter had a lot more blog entries dedicated to it than

Casanova. Harry Potter was also mentioned in more blogs, as indicated by the colors of the blog entry nodes.

[Figure 18] shows the slider view for the movie case study, with different parameters and ranges selected in the parallel axes. We can see that the popular movie *King Kong* had a longer 'lifespan'; i.e. a longer timeframe in which the movie was still talked about in the blogs. The movie *Jarhead* had a significantly shorter lifespan. The density of blog activities for *King Kong* is also higher than the movie *Wolf Creek*, which had a long lifespan, but had an extremely sparse set of blog entries dedicated to it.

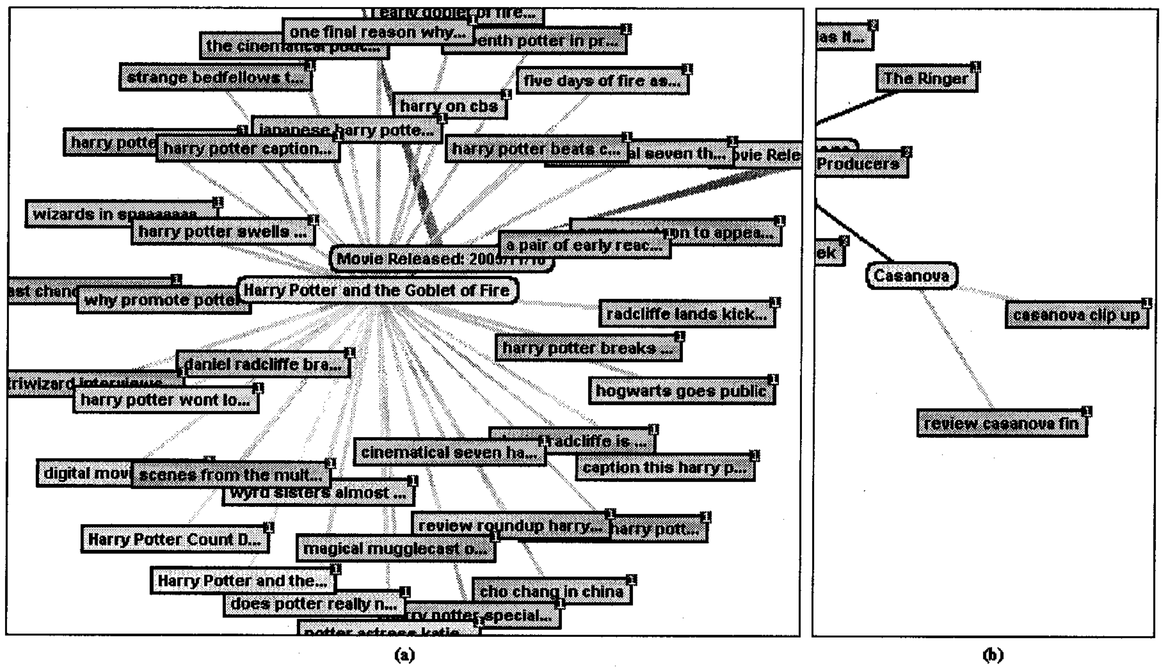


Figure 17: Graph view of eNulog showing the movies Harry Potter and the Goblet of Fire (a) and Casanova (b).

Other trends can also be identified using the slider view; users can select entries from one single blog and determine what movies are covered in these entries, as shown in [Figure 19](a); or, users can select a certain timeframe and examine which blog is updated the most frequently within that timeframe, as shown in [Figure 19](b).

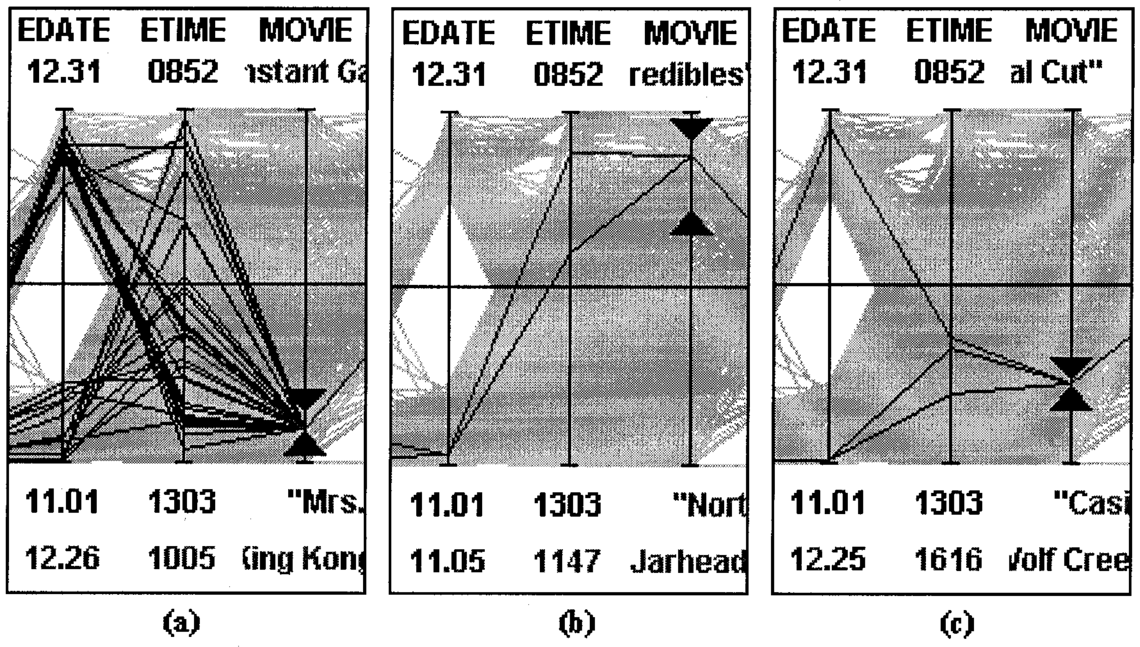


Figure 18: Slider view with the axes for Entry date (EDATE), Entry time (ETIME) and Movie title (MOVIE). Each line represents an entry for the selected movie (*King Kong* in (a), *Jarhead* in (b), and *Wolf Creek* in (c)).

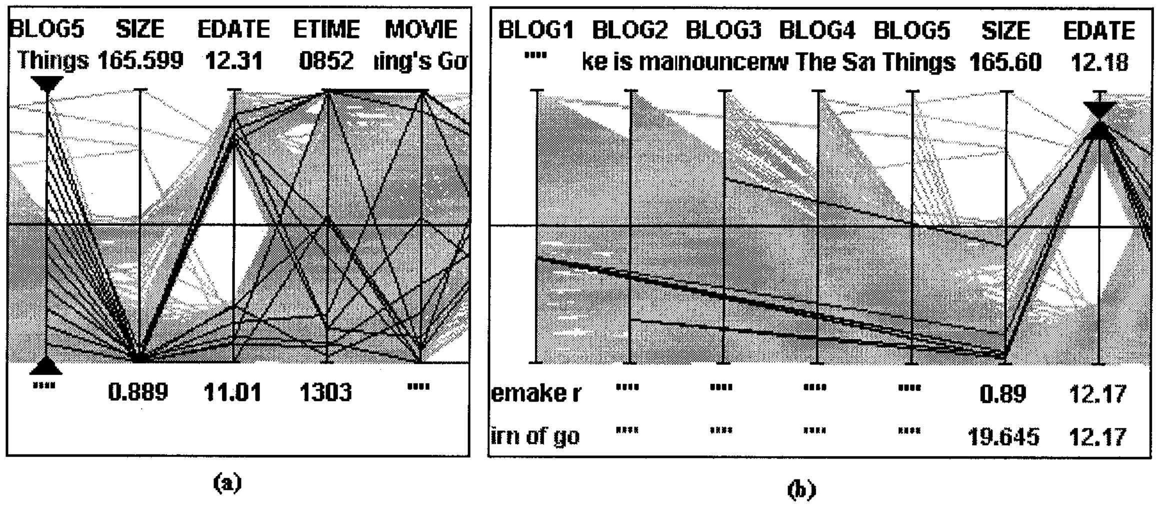

BLOG5 SIZE EDATE ETIME MOVIE
Things 165.599 12.31 0852 ıing's Go
"" 0.889 11.01 1303 ""
(a)
BLOG1 BLOG2 BLOG3 BLOG4 BLOG5 SIZE EDATE
"" ke is mainouncenw The Sa Things 165.60 12.18
emake r "" "" "" "" 0.89 12.17
irn of go "" "" "" "" 19.645 12.17
(b)

Figure 19: Entries from a single blog (Cinecultist) is selected in (a), while entries from a single date (12/17) are selected in (b).

[Figure 20] shows the temperature map view. Based on the relative sizes of the nodes as seen in [Figure 20](a), users can immediately see that *Chronicles of Narnia* was more popular than movies with smaller nodes, such as *Saw II*. The node for *Chronicles of Narnia* is also more opaque, indicating that the movie was covered in more blogs. Users can also examine blog activities with the temperature map view; [Figure 20](b) shows that there were many updates on Cinematical on November 29th. Users can expand the nodes, as shown in [Figure 20](c), to see further relationships between blogs and movies.

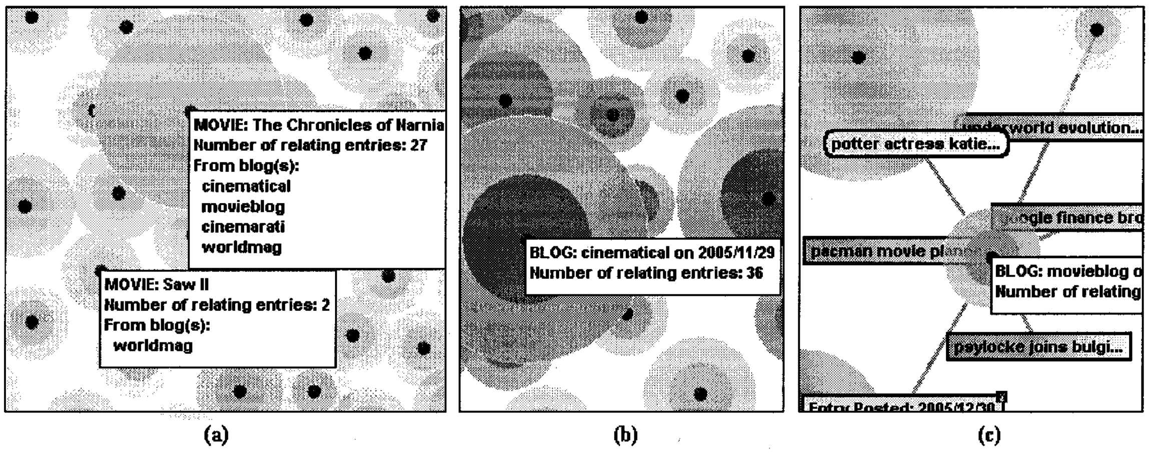


Figure 20: Temperature Map of eNulog. (a) shows the movie nodes, (b) shows the blog nodes, and (c) shows an expanded blog node, which has entries linking to two related movie nodes (Harry Potter, which is represented as a large blue node to indicate greater popularity, and Underworld: Evolution, which is represented as a small blue node).

We examined the visualizations to obtain the blog activity data described in Section 4.2.1 for the statistical analysis. We paired the movie success metrics with the blog activity data, and we computed their correlation coefficients to determine if the two parameters do correlate with each other. The correlation coefficient (CC) is calculated using [Eq. 8], which determines the strength of relationship between variables X and Y. The CC can range from -1 to 1, and a higher magnitude signifies a stronger correlation.

$$r_{xy} = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\ \sqrt{n\sum y_i^2 - (\sum y_i)^2}}. \qquad \textbf{[Eq. 8]}$$

[Table 5] summarizes our calculation results. We see that all CCs with the exception of one have high magnitudes, as highlighted by the bolded values in the table. We observed that the correlation between cumulative gross and blog activities are notably higher, indicating that blog activities are highly related to a movie's overall success. The strongest correlation is shared between the cumulative gross and the average posts per day over a fixed time period, and since our time period spans the two weeks before and after a movie's release, we can claim that the blog 'buzz' surrounding the time of the movie's release is extremely important in determining the movie's success in the box office.

Table 5: Correlation coefficients (CC) between success values (X) and blog activity values (Y)

X(Success)	Y (blog activity metric)	CC
Opening	Number of blog entries posted about the movie before its release	**0.77**
Weekend	Number of blogs that contain at least one entry about the movie before its release	0.53
Gross	Average post per day over the movie's lifespan before its release	**0.74**
3-Weeks	Number of blog entries posted about the movie	**0.89**
Cumulative	Number of blogs that contains at least one entry about the movie	**0.73**
Gross	Average post per day over the movie's total lifespan	**0.72**
	Average post per day over four weeks	**0.94**

We also examined a movie's success based on how well professional critics received it. Movies with positive critical reviews can generate hype in the blogosphere. Although they are likely not as highly discussed as box office hits, they should receive more blog activities over movies that are neither box office success or critically acclaimed. We used the number of award nominations from the two major movie awards in Hollywood, the Oscars and the Golden Globe, as an indicator of how well received a movie is with critics. We manually obtained the number of award nominations from IMDB [49] for each movie that appeared in our data set in the statistical analysis.

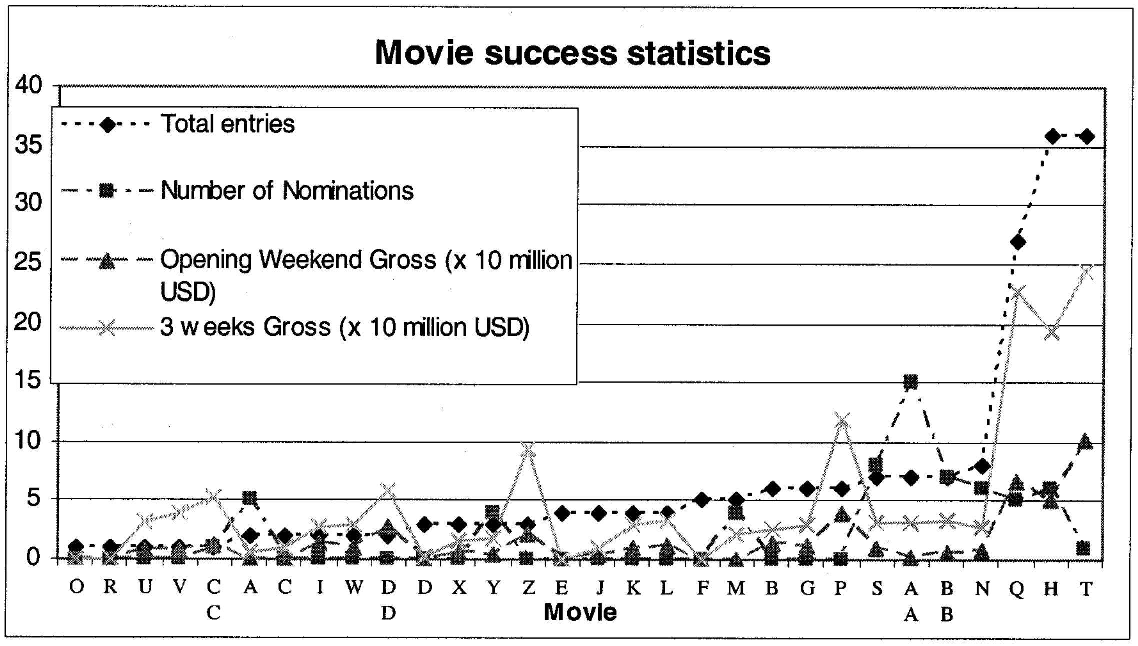


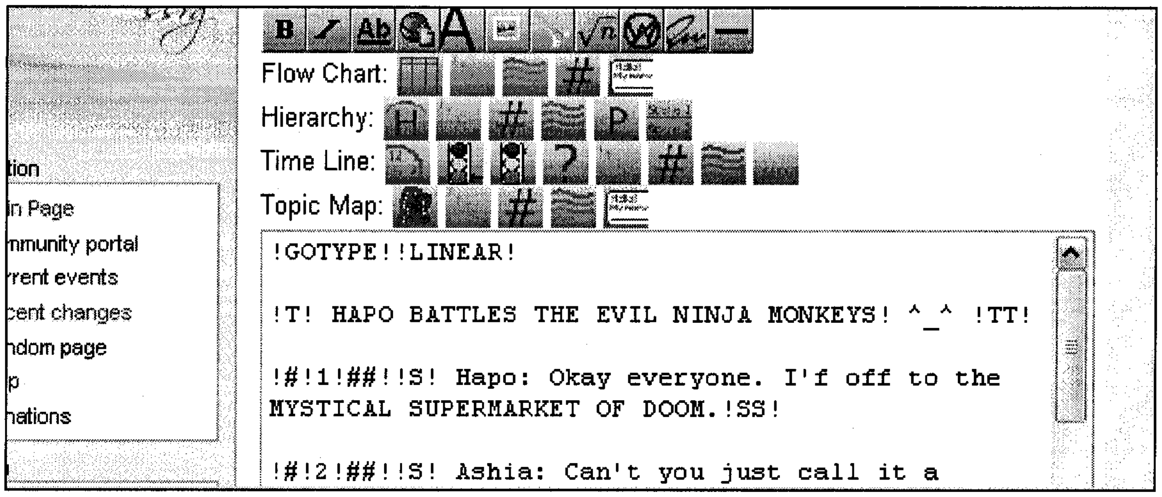
Figure 21: Number of related entries, nominations, and gross values for each movie. Movie titles are represented with letter ID on the X-axis. All movies to the right of Movie S have at least six nominations, or a three-week cumulative gross of 150 million dollars.

[Figure 21] graphs the total number of related entries, the number of award nominations, and the gross values for each movie. From the graph, we could see that movies that were mentioned in more than six entries have either a high number of award nominations, or a high cumulative gross. We could also see that movies with the highest cumulative grosses have the most number of related entries, followed by 'critically acclaimed' movies, then movies that could be considered as less successful.

4.3 AnnokiBlooms: An Annoki-based tool for story planning and understanding

We took advantage of the ability to manipulate the domain interface and the graphical interface of the Annoki system to tailor it to a specific usage. We examined how we could utilize graphic organizers to help young students write and/or understand a story at a different cognitive level as stated in Bloom's Taxonomy [3]. The idea is to have teachers and/or students write or plan a story on an Annoki wiki page using special templates. The Annoki system makes a special feed in XTM format for the page based on the template, which is then visualized as a graphic organizer in the client interface. Like in ENWiC, each type of GO has its own association and node types, which are rendered accordingly in the interface.

The benefit of using the wiki to write a story is that it is easy for users to learn how to make and edit wiki pages, which is especially important when we are dealing with young children with little experience. Wikis are also easily accessible, thus students and teachers are able to use it at school or at home. Teachers can also keep track of students' progress and make comments and edits to the stories if necessary. Visualizing the stories as graphic organizers helps students review the stories, and having visual pictures on hand may also simulate the students' motivation and interest in learning from and working with the stories.



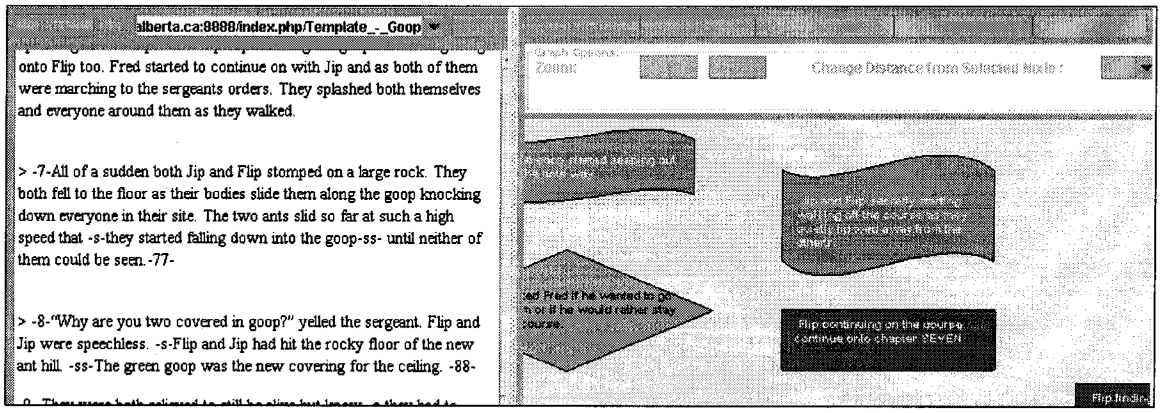
```
!GOTYPE!!LINEAR!

!T! HAPO BATTLES THE EVIL NINJA MONKEYS! ^_^ !TT!

!#!1!##!!S! Hapo: Okay everyone. I'f off to the
MYSTICAL SUPERMARKET OF DOOM.!SS!

!#!2!##!!S! Ashia: Can't you just call it a
```

Figure 22: Example script written using GO templates. The toolbars on top of the text editors help add the tags to the GO template.

To support the story writing process in Annoki, special tags are added to represent graphic organizer elements. New toolbars are added to the wiki page editor with buttons for each GO element. Each type of GO has its own toolbar; currently, the supported GOs are flow charts, hierarchy diagrams and linear string/timeline. Users can also build a generic topic map instead of a GO. Each template starts with the tag "!GOTYPE!" followed by the type of GO (e.g. LINEAR, FLOWCHART), and this tag must be in the first line of the page for Annoki to recognize it is a GO page. Users can use the special tags to indicate what should be made into topics and what should be made into associations. [Figure 22] shows an example of a script for a story written in the linear string template. Each step is numbered so that Annoki recognizes the order and establishes the associations appropriately, so that step *n* is associated to step *n*+1. For other types of GO, it may be necessary to specify the association manually; e.g. in a flow chart, if a decision can make step *x* jumps to step *y*, the tag "-sp-*x*_*y*-" is used to specify such path.

When a feed for the page is requested, Annoki first determines what type of GO template is used by checking for the GOTYPE tag in the first line. It then looks for the appropriate tags to set up the topics and associations for the topic map, and the result is exported in XTM format.

AnnokiBlooms is the client interface based on TOMU, which is implemented by Michelle Annett, as a summer USRA project. It is designed for young students to view the GOs created in Annoki. AnnokiBlooms extends the standard Annoki interface as it provides a more pictorial interface, with emphasize on expressing a story than acting as a content management tool. This makes the interface more appealing and engaging for students. [Figure 23] shows a screenshot of the AnnokiBlooms interface. The browser on the side panel allows students to access a story on the Wiki; when the desired story is found, the GO for the story can be loaded to the right visualization panel.


alberta.ca:8888/index.php/Template_-_Goop

onto Flip too. Fred started to continue on with Jip and as both of them were marching to the sergeants orders. They splashed both themselves and everyone around them as they walked.

> -7-All of a sudden both Jip and Flip stomped on a large rock. They both fell to the floor as their bodies slide them along the goop knocking down everyone in their site. The two ants slid so far at such a high speed that -s-they started falling down into the goop-ss- until neither of them could be seen.-77-

> -8-"Why are you two covered in goop?" yelled the sergeant. Flip and Jip were speechless. -s-Flip and Jip had hit the rocky floor of the new ant hill. -ss-The green goop was the new covering for the ceiling. -88-

Graph Options:
Zoom:
Change Distance from Selected Node :
ed Fred if he wanted to go
n or if he would rather stay
course.
Flip continuing on the course
continue onto chapter SEVEN
Flip finding

Figure 23: The AnnokiBlooms interface displaying a flow chart.

4.3.1 Pilot study

We performed a pilot study on how AnnokiBlooms can help students write a story. Our subjects were 12-15 years old students attending a computing science summer camp. They were given a project to create a short computer animation movie, and to start the project, they were asked to plan their stories on Annoki. We gave a brief tutorial on how to use the Wiki, and the linear string template. The students were asked to write their story in a script format, with each line of the script set up as a numbered step in the template. After they completed their scripts, we showed them how to use AnnokiBlooms, which could be used to review their stories as they move on to create the actual animation. When the students completed their projects, they were asked to fill out a user satisfaction questionnaire.

[Table 6] summarizes the results from the questionnaire. Based on the questions that obtained a score of at least 4.0/5.0, we could see that the students found the Annoki wiki and AnnokiBlooms easy to use and understand, and that the system has helped them plan their stories faster. Since this study is performed in an informal classroom setting, we will still have to investigate how the system can benefit students in a regular classroom environment, especially in language classes where story writing is important.

Table 6: Average scores (out of 5.0) subjects gave for each question in the satisfactory questionnaire.

Question	Avg. Score
1. The AnnokiBlooms diagrams have helped me review my materials.	3.6
2. It is easy to add information to Annoki.	3.9
3. The steps needed to make a diagram are easy and straightforward.	3.6
4. Writing a story on Annoki is easy.	4.5
5. There is enough information displayed on the AnnokiBlooms diagrams.	3.8
6. The story I wrote on Annoki is displayed correctly in AnnokiBlooms.	4.1
8. The AnnokiBlooms program is easy to use.	4.0
9. I understand the diagrams in AnnokiBlooms.	4.0
10. Using the diagrams has helped me with my story planning.	3.8
11. The speed of this software is fast.	4.2
12. Annoki and AnnokiBlooms helped me create and plan my story faster.	4.4
13. The AnnokiBlooms diagrams are clear and concise.	4.1
14. The program has attractive presentations.	4.1
15. The program has done better than I expected.	3.8
16. The program makes story planning more interesting.	3.4
17. Working with the Annoki and AnnokiBlooms is satisfying.	3.7

Chapter 5: Contributions, Conclusions and Future Work

The web has become an invaluable tool for the sharing of information. Students can utilize the collection of online documents as an information repository for their studies, while market researchers can see the Internet as a source for collecting public opinions. With so much data available on the web, this thesis focused on providing a method to organize these data better to help users navigate and locate the information of interest. This is especially important for Wikis and blogs; with the increased usage of Wikis and blogs over the recent years, they have accumulated many useful data, but the lack of organization and navigation support in these tools has made it difficult for users to efficiently discover helpful information.

5.1 Contributions

In this thesis, we have presented components and systems that help represent the structures of Wikis and blogs. The work of this thesis focused on generating topic maps that convey the content present in Wikis and blogs, and producing visualizations to improve users' understanding and perception of the domain. The effectiveness of our methodology was evaluated using the ENWiC and eNulog systems; the ENWiC experiment demonstrated that the topic map obtained from its hypertext document retrieval component was beneficial in helping users locate information with less effort; the eNulog case study showed that there is correlation between blog activities and product success, and thus the topic map visualizations of blog activities can be valuable for market researchers to examine what is popular in the market today. Our empirical studies supported this thesis' hypotheses that semantic structures and visualizations help users better perceive information from wikis. The Annoki system was built to further improve the quality of the semantic structures, and its live topic map feed component enables users to obtain up-to-date visualizations of the domain. Our application of Annoki to storytelling for young students demonstrated its usability for educational purposes.

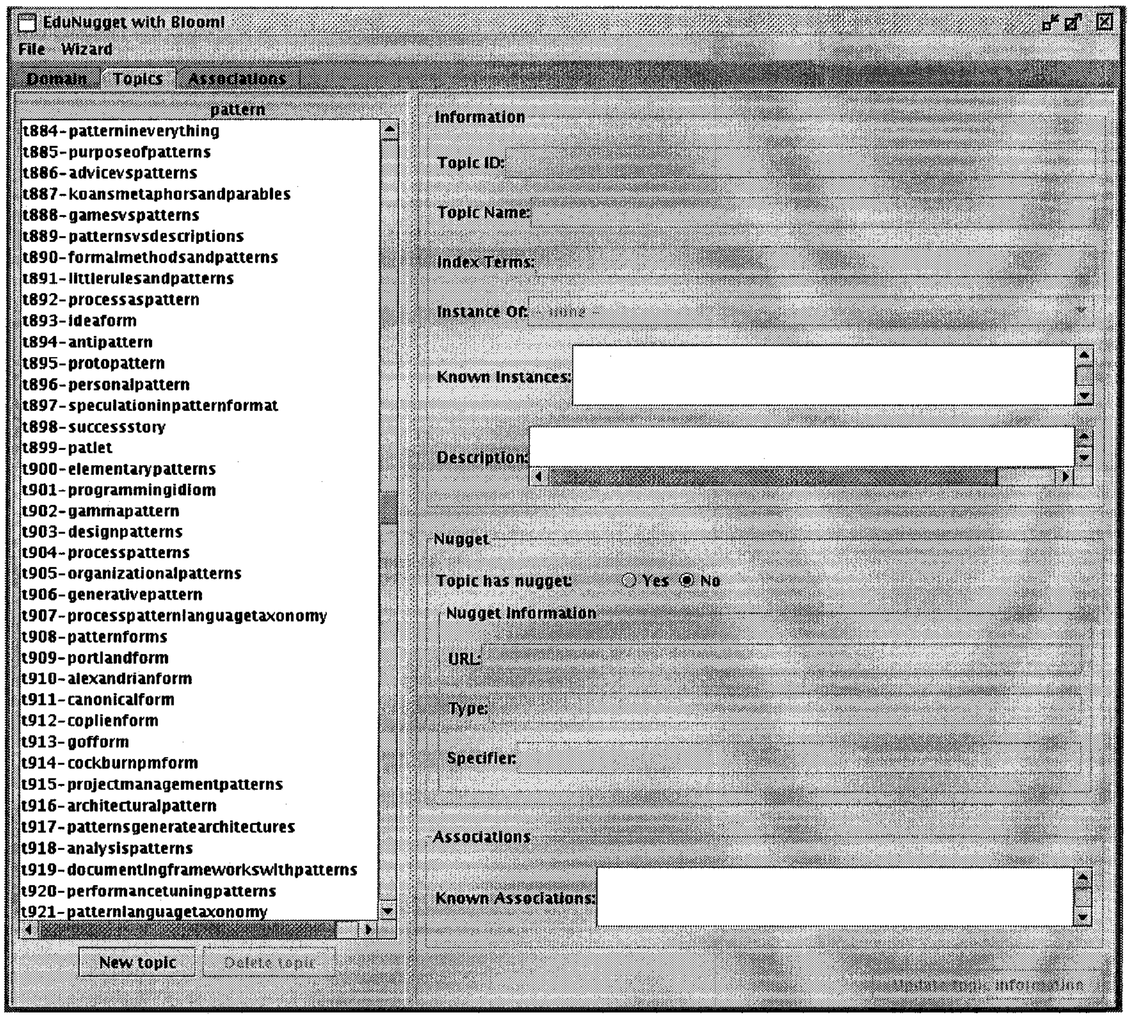
5.2 Future Work

We look to improve the visualization client of Annoki by implementing more types of graphic organizers. From our ENWiC experiment, we observed that GOs are useful in helping users identify relationships faster. We look to implement an automatic GO generator similar to the one in ENWiC, which will utilize information from the wiki database to produce GOs in the topic map; for example, a double cell diagram of two pages can be generated by examining what links they have in common, and what links they have that are different from each other. We can perform a usability experiment similar to ENWiC where subjects will perform tasks using the GOs, and we can observe if the GOs help improve the subjects' performance.

We can also apply the Annoki philosophy to blogs. Blog entries are usually organized in chronological order, which makes it difficult for readers to locate entries that deal with a specific topic. Like with the Annoki wiki, we can implement better semantics into the blog software to help structure the entries better. We can also implement a GO generator to produce more views in the client interface; GOs that compare entities, such as the double cell diagram and the comparison matrix, can be very useful for market researchers to evaluate different products.

Since it is important for market researchers to recognize current events happening in the market, we can examine the use of RSS or Atom feeds to obtain up-to-date blog data for eNulog. This will provide us with a larger data set that allows us to further expand our statistical analysis, as well as apply the system to different domains with frequently changing data, such as music news, and political news. When the system has up-to-date data, it can also make a prediction on what may happen in the future, based on the activities currently happening in the blogs. We can observe these predictions through time and determine if the system is effective in foreseeing what will happen in the market.

Appendix A: Interfaces of ENWiC, eNulog and Annoki



pattern

t884-patternineverything
t885-purposeofpatterns
t886-advicevspatterns
t887-koansmetaphorsandparables
t888-gamesvspatterns
t889-patternsvsdescriptions
t890-formalmethodsandpatterns
t891-littlerulesandpatterns
t892-processaspattern
t893-ideaform
t894-antipattern
t895-protopattern
t896-personalpattern
t897-speculationinpatternformat
t898-successstory
t899-patlet
t900-elementarypatterns
t901-programmingidiom
t902-gammapattern
t903-designpatterns
t904-processpatterns
t905-organizationalpatterns
t906-generativepattern
t907-processpatternlanguagetaxonomy
t908-patternforms
t909-portlandform
t910-alexandrianform
t911-canonicalform
t912-coplienform
t913-gofform
t914-cockburnpmform
t915-projectmanagementpatterns
t916-architecturalpattern
t917-patternsgeneratearchitectures
t918-analysispatterns
t919-documentingframeworkswithpatterns
t920-performancetuningpatterns
t921-patternlanguagetaxonomy

New topic | Delete topic

Information

Topic ID:

Topic Name:

Index Terms:

Instance Of: - none -

Known Instances:

Description:

Nugget

Topic has nugget: Yes [✓ No]

Nugget Information

URL:

Type:

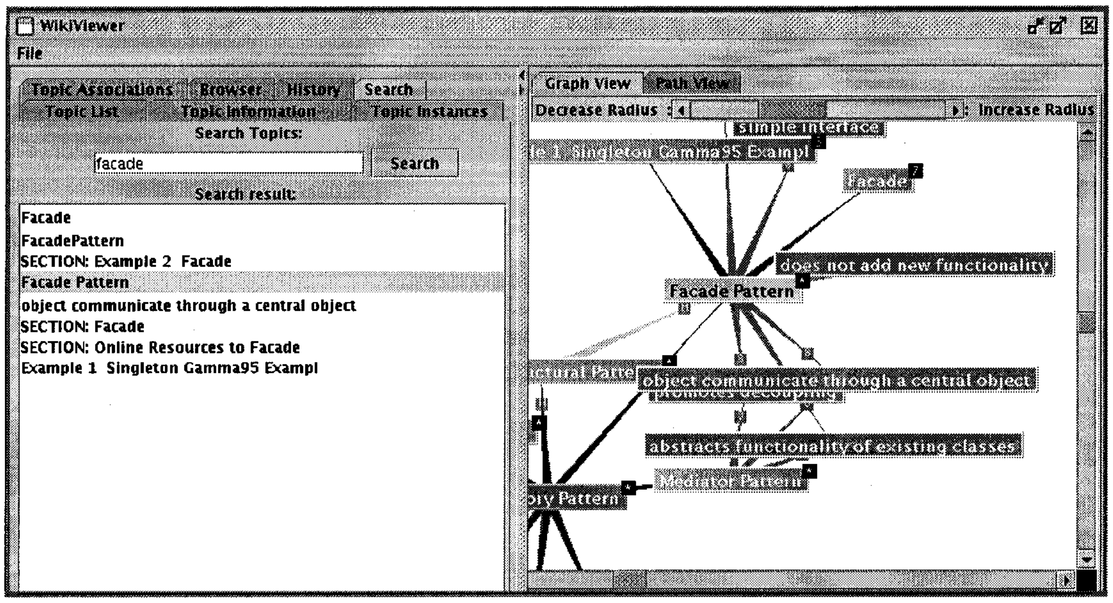
Specifier:

Associations

Known Associations:

Update topic information

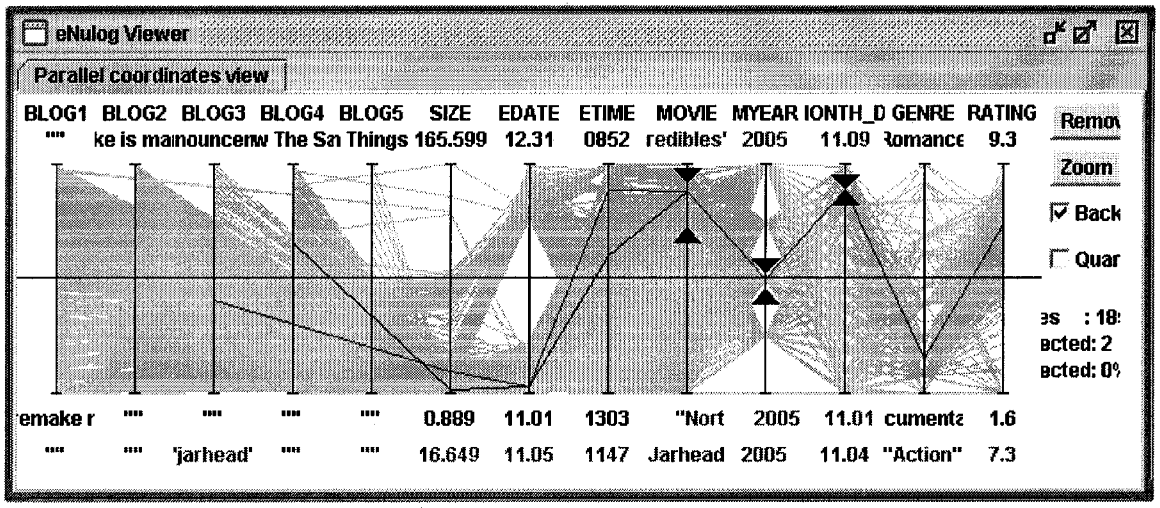
Figure A 1: Administrator Interface of ENWiC. The left panel shows the topics currently in the domain, and the right panel enables users to view and edit topic information. Users can also use the tabs under the menu to access the Association panel, which allows them to edit associations, or the Domain panel, which allows them to add/delete domains.



facade

Facade
FacadePattern
SECTION: Example 2 Facade
Facade Pattern
object communicate through a central object
SECTION: Facade
SECTION: Online Resources to Facade
Example 1 Singleton Gamma95 Exampl

Figure A 2: Client interface of ENWiC, showing the search result of the keyword 'façade' in the search panel on the left. The right panel shows the topic map that is centered on the Façade pattern node. The tabs on the left panel allow users to see other information on the topic/domain.



```
BLOG1   BLOG2          BLOG3      BLOG4   BLOG5   SIZE      EDATE   ETIME   MOVIE       MYEAR   IONTH_D   GENRE      RATING
""      ke is ma       announcem  w The Sa n Things 165.599 12.31   0852    redibles'   2005    11.09     Romance    9.3
emake r ""             ""         ""      ""      0.889     11.01   1303    "Nort       2005    11.01     cumenta    1.6
""      ""             'jarhead'  ""      ""      16.649    11.05   1147    Jarhead     2005    11.04     "Action"   7.3
```

Figure A 3: Slider view in eNulog with "Jarhead" selected in the MOVIE axis, and Jarhead's movie release date (2005/11/04) selected in the MYEAR (Movie year) and MMONTH_DAY (Movie month/day) axes. The other axes are: BLOG1-BLOG5, which contains entries from blogs; SIZE, which is the size of the entry file; EDATE/ETIME, which is the posted date/time of the entry; GENRE, which is the genre of the movie, and RATING, which is the user's rating of the movie from IMDB [49]. The red line is the current selected line; its value for each axis is written in the bottom row (in red).

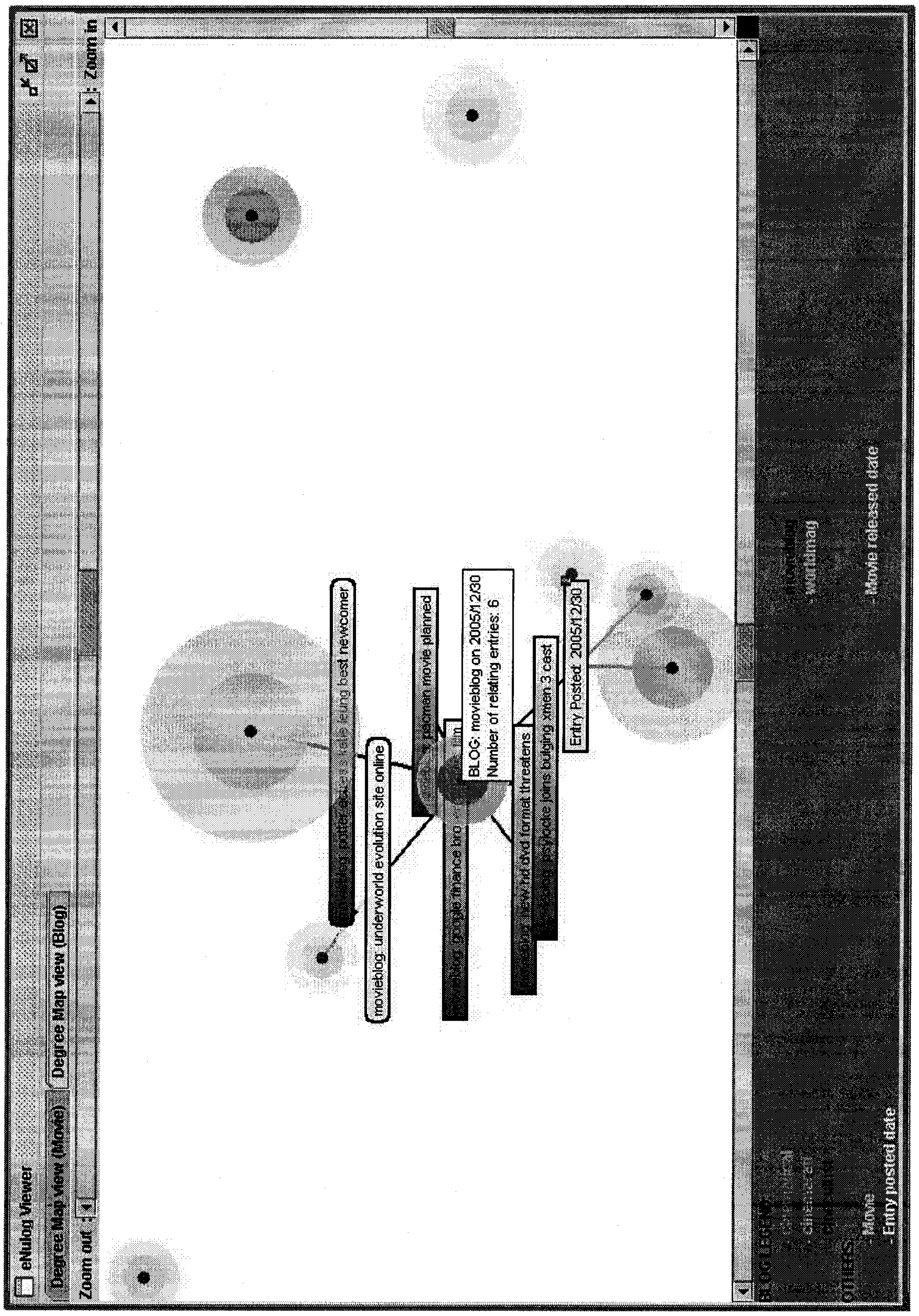
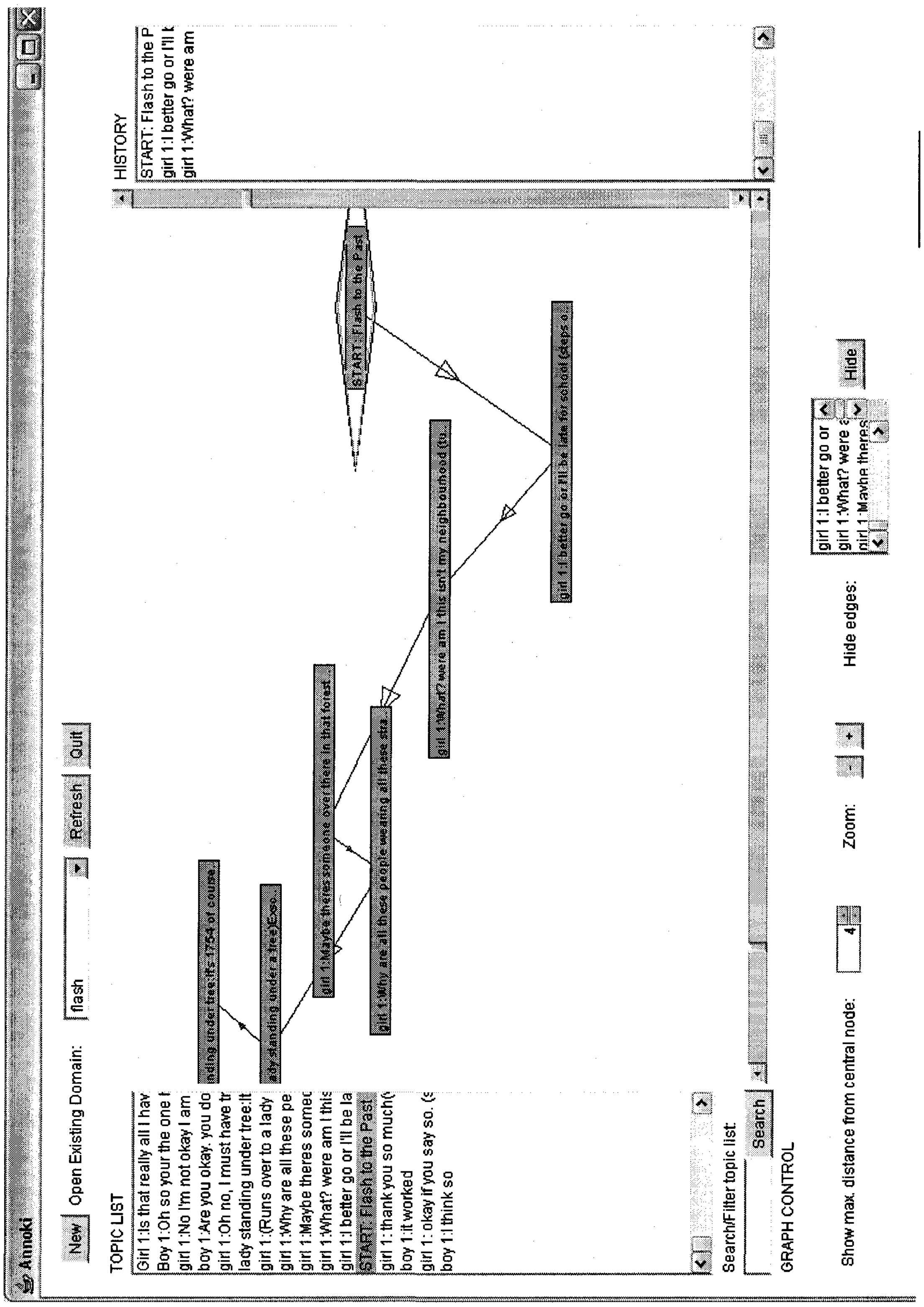


Figure A 4: The temperature map in eNulog for blogs, showing an expanded node for a day (2005/12/30) in The Movie Blog [88]. The entries are the red rectangular nodes; the two blue circular nodes are movie nodes that are associated with some of the entries made on that day on the Movie Blog.

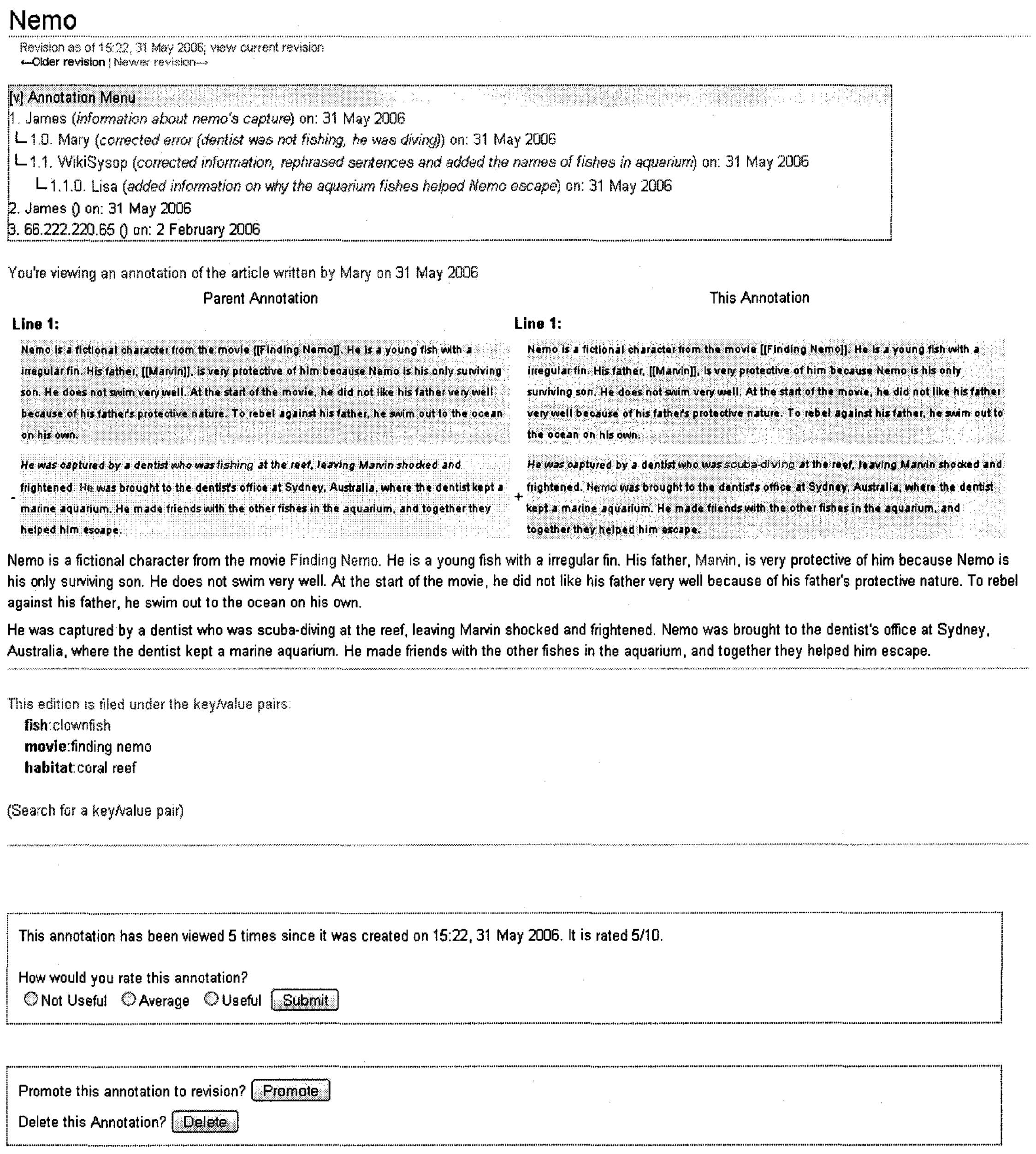

Annoki
New
Open Existing Domain:
flash
Refresh
Quit
TOPIC LIST
HISTORY
Search/Filter topic list:
Search
GRAPH CONTROL
Show max. distance from central node:
4
Zoom:
-
+
Hide edges:
Hide
START: Flash to the Past
girl 1:I better go or I'll be late for school (steps o
girl 1:What? were am I this isn't my neighbourhood (lo
girl 1:Maybe theres someone over there in that forest
girl 1:Why are all these people wearing all these stra
lady standing under a tree)Exe
nding under tree:It's 1754 of course.

Girl 1:Is that really all I hav
Boy 1:Oh so your the one t
girl 1:No I'm not okay I am
boy 1:Are you okay, you do
girl 1:Oh no, I must have tr
lady standing under tree:It
girl 1:(Runs over to a lady
girl 1:Why are all these pe
girl 1:Maybe theres somec
girl 1:What? were am I this
girl 1:I better go or I'll be la
START: Flash to the Past
girl 1: thank you so much(
boy 1:it worked
girl 1: okay if you say so. (s
boy 1:I think so

START: Flash to the P
girl 1:I better go or I'll b
girl 1:What? were am

girl 1:I better go or
girl 1:What? were a
girl 1:Mavbe theres

Figure A 5: The Annoki interface, showing a script for a story as a topic map. The left panel shows a list of topics; the bottom panel contains functions to control the graph's appearance. The right panel shows the history of nodes visited.



Nemo

Revision as of 15:22, 31 May 2006; view current revision
←Older revision | Newer revision→

[v] Annotation Menu
1. James (*information about nemo's capture*) on: 31 May 2006
└ 1.0. Mary (*corrected error (dentist was not fishing, he was diving)*) on: 31 May 2006
└ 1.1. WikiSysop (*corrected information, rephrased sentences and added the names of fishes in aquarium*) on: 31 May 2006
  └ 1.1.0. Lisa (*added information on why the aquarium fishes helped Nemo escape*) on: 31 May 2006
2. James () on: 31 May 2006
3. 66.222.220.65 () on: 2 February 2006

You're viewing an annotation of the article written by Mary on 31 May 2006

	Parent Annotation		This Annotation
	Line 1:		**Line 1:**
	Nemo is a fictional character from the movie [[Finding Nemo]]. He is a young fish with a irregular fin. His father, [[Marvin]], is very protective of him because Nemo is his only surviving son. He does not swim very well. At the start of the movie, he did not like his father very well because of his father's protective nature. To rebel against his father, he swim out to the ocean on his own.		Nemo is a fictional character from the movie [[Finding Nemo]]. He is a young fish with a irregular fin. His father, [[Marvin]], is very protective of him because Nemo is his only surviving son. He does not swim very well. At the start of the movie, he did not like his father very well because of his father's protective nature. To rebel against his father, he swim out to the ocean on his own.
-	He was captured by a dentist who was fishing at the reef, leaving Marvin shocked and frightened. He was brought to the dentist's office at Sydney, Australia, where the dentist kept a marine aquarium. He made friends with the other fishes in the aquarium, and together they helped him escape.	+	He was captured by a dentist who was scuba-diving at the reef, leaving Marvin shocked and frightened. Nemo was brought to the dentist's office at Sydney, Australia, where the dentist kept a marine aquarium. He made friends with the other fishes in the aquarium, and together they helped him escape.

Nemo is a fictional character from the movie Finding Nemo. He is a young fish with a irregular fin. His father, Marvin, is very protective of him because Nemo is his only surviving son. He does not swim very well. At the start of the movie, he did not like his father very well because of his father's protective nature. To rebel against his father, he swim out to the ocean on his own.

He was captured by a dentist who was scuba-diving at the reef, leaving Marvin shocked and frightened. Nemo was brought to the dentist's office at Sydney, Australia, where the dentist kept a marine aquarium. He made friends with the other fishes in the aquarium, and together they helped him escape.

This edition is filed under the key/value pairs:
fish:clownfish
movie:finding nemo
habitat:coral reef

(Search for a key/value pair)

This annotation has been viewed 5 times since it was created on 15:22, 31 May 2006. It is rated 5/10.

How would you rate this annotation?
○ Not Useful ○ Average ○ Useful [Submit]

Promote this annotation to revision? [Promote]
Delete this Annotation? [Delete]

Figure A 6: A page from the Annoki wiki that shows an annotation of the page 'Nemo'. It has: an annotation menu, a differencing table, the page text, the property/value pairs, a rating form, and a form for administrators to manage the annotation.

Bibliography

1. Amin, A.B.M. “Using Graphic Organisers to promote active e-learning” [Electronic version]. In *Proc. of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2005*, pages 4010-4015, AACE, Norfolk, VA, 2005.
2. Aumueller, D. “SHAWN: Structure Helps a Wiki Navigate”. In *Proc. Of the BTW-Workshop “WebDB Meets IR”,* 2005.
3. Bloom, B.S. *Taxonomy of Educational Objectives, Handbook I: The Cognitive Domain*, Addison Wesley, 1956.
4. Campanini, S.E., Castagna P. & Tazzoli R. “Platypus Wiki: a Semantic Wiki Wiki Web” [Electronic version]. In *Proc. Semantic Web Applications and Perspectives– 1st Italian Semantic Web Workshop*, 2004.
5. Canas, A.J., Carff, R., Hill, G., Carvalho, M., Arguedas, M., Eskridge, T.C, Lott, J. & Carvajal, R. “Concept Maps: Integrating Knowledge and Information Visualization” [Electronic version]. In *Knowledge and Information Visualization 2005*, pages 205-219, 2005.
6. Dennis, B.M. and Jarrett, A.C. “NusEye: Visualizing Network Structure to Support Navigation of Aggregated Content”. In *Proc. of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05)*, pages 107c-, 2005.
7. Dicheva D., Dichev, C. & Wang, D. “Visualizing Topic Maps for e-Learning” [Electronic version]. In *Proc. of the Fifth IEEE International Conference on Advanced Learning Technologies (ICALT'05)*, pages 950-951, 2005.
8. Donelan, C.I. “From Spider Maps to Double-Cell Diagrams: Graphic Organizers Support Student Learning”. *ENC Online.* Accessed May 2005, from http://www.enc.org/features/focus/archive/graphic/document.shtm?input=FOC-003559-index
9. Espiritu, C., Stroulia, E. Tirapat, T. “ENWiC: Visualizing Wiki Semantics as Topic Maps”. In *Proc. Of the 8th International Conference on Enterprise Information Systems (ICEIS 2006),* Paphos, Cyprus, May 2006.
10. Gardner, H. *Frames of Mind: The Theory of Multiple Intelligence.* Basic, 1983.

11. Garshol, L.M. "What are Topic Maps". Accessed August 2006, from http://www.xml.com/pub/a/2002/09/11/topicmaps.html

12. Herring, S.C., Kouper I., Paolillo J.C., Scheidt L.A., Tyworth M., Welsch P., Wright E., Yu N. "Conversations in the Blogosphere: An Analysis "From the Bottom Up"". In *Proc. of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05),* pages 107b-, 2005.

13. Jakubovitz, J. & Paul, D. "Wide World of Wikipedia". *The Emory Wheel.* Accessed May 2006, from http://www.emorywheel.com/media/storage/paper919/news/2006/04/21/News/Wide-World.Of.Wikipedia-1865022.shtml

14. Kurashima, T., Tezuka, T. and Tanaka, K. "Blog Map of Experiences: Extracting and Geographically Mapping Visitor Experiences from Urban Blogs." In *Proc. of the 6th International Conference on Web Information Systems Engineering (WISE 2005),* pages 496-503, Springer, 2005.

15. Lamb, A. "Learning Resources: Graphic Organizers". *The Teacher Tap*, May, 2001. Accessed August 2005, from http://eduscapes.com/tap/topic73.htm

16. Lavik, S. and Nordeng, T.W. "BrainBank Learning – Building Topic Maps-Based E-Portfolios." In *Proc. of the First International Conference on Concept Mapping, Pamplona,* Spain, 2004.

17. Lee, Y.J. "Facilitating Web Search with Visualization and Data Mining Techniques" [Electronic version]. *Knowledge and Information Visualization 2005*, pages 326-342. 2005.

18. Lesser, E.L and Storck, J. "Communities of practice and organizational performance". *IBM Systems Journal*, Vol. 40, Num. 4, 2001. Accessed August 2006, from http://researchweb.watson.ibm.com/journal/sj/404/lesser.html.

19. Librelotto, G., Ramalho, J.C. and Henriques, P.R. "TM-Builder: An Ontology Builder based on XML Topic Maps". *CLEI Electronic Journal*, 7, 2. Accessed January 2006, from http://www.clei.cl/cleiej/papers/v7i2p4.pdf.

20. Liu,Y. & Stroulia,E. "A Lightweight Project-Management Environment for Small Novice Teams". In *Proc. of the 3rd International Workshop on Adoption-Centric Software Engineering*, pages 42-48, Portland, OR, USA, 2003.

21. McCallum, A.K. "Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering." 1996. Accessed December 2005, from http://www.cs.cmu.edu/~mccallum/bow.

22. Nanno, T., Fujiki, T. and Suzuki, Y. “Automatically Collecting, Monitoring, and Mining Japanese Weblogs.” In *Proc. of the 13th International Conference on World Wide Web – Alternate Track Papers & Posters (WWW 2004),* ACM, pages 320-321, 2004

23. Nickols, F. “Communities of Practice Resources”. Accessed August 2006, from http://home.att.net/~discon/KM/CoPs.htm.

24. O’Neil, M. “Automated Use of a Wiki for Collaborative Lecture Notes”. In *Proc. of the 36th SIGCSE technical symposium on Computer science education*, pages 267-271, St. Louis, Missouri, USA, 2005.

25. Oren, E. “Semantic Wikis for Knowledge Workers”. *CDH Seminar*, 2005. Accessed May 2006, from http://m3pe.org/seminar/oren.pdf.

26. Power, R. “Topic Maps for Context Management”. In *Proc. of the 1st international symposium on Information and communication technologies*, Dublin, Ireland, 2003

27. Riddell, R. “Wikis Test Students' Research Skills”. *eSchool News Online*, 2006. Accessed May 2006, from http://www.eschoolnews.com/news/showStory.cfm?ArticleID=6069

28. Robinson, D.H. & Skinner, C.H. “Why Graphic Organizers Facilitate Search Processes: Fewer Words or Computationally Efficient Indexing?” [Electronic version]. *Contemporary Educational Psychology*, 21, pages 166-180, 1996.

29. Schaffert, S. “IkeWiki - A Semantic Wiki for Collaborative Knowledge Management”. In *STICA06*, Manchester, UK, 2006.

30. Seigenthaler, J. “A false Wikipedia 'biography'”. *USA Today*, 2005. Accessed May 2006, from http://www.usatoday.com/news/opinion/editorials/2005-11-29-wikipedia-edit_x.htm

31. Siirtola, H., “Combining Parallel Coordinates with the Reorderable Matrix”. In *Jonathan Roberts (Ed.), In Proc. of the International Conference on Coordinated & Multiple Views in Exploratory Visualization* (*CMV 2003*). Pages 63-74, IEEE Computer Society, London, UK, 2003.

32. Smith M.A. & Fiore A.T. "Visualization components for persistent conversations" [Electronic version]. In *Proc. of CHI 2001,* 2001.
33. Sowa, J.F. "Semantic Networks". Accessed August 2006, from http://www.jfsowa.com/pubs/semnet.htm.
34. Stroulia E. & Jari K. "EduNuggets: an intelligent environment for managing and delivering multimedia education content". In *Proc. of the International Conference on Intelligent User Interfaces 2003*, 2003.
35. Tirapat, T., Espiritu, C. and Stroulia, E. "Taking the Community's Pulse, one Blog at a Time". In *Proceedings of the Sixth International Conference on Web Engineering*, Palo Alto, CA, USA, 2006.
36. Viégas, F.B., Wattenberg M. & Dave, K. "Studying Cooperation and Conflict between Authors with history flow Visualizations". In *Proc. of SIGCHI*, Vienna, Austria, 2004.
37. Wang, C. and Turner, D. "Extending the Wiki Paradigm for Use in the Classroom". In *Proc. Of the International Conference on Information Technology: Coding and Computing,* pages 255-261, Las Vegas, Nevada, 2004.
38. Wenger, E., et al. "Technology for Communities". *CEFRIO.* Accessed August 2006, from http://technologyforcommunities.com/CEFRIO_Book_Chapter_v_5.2.pdf.
39. "Blog Mining Gets Real". Accessed January 2006, from http://www.crmbuyer.com/story/43483.html
40. Graph::Layouter (Perl). Accessed June 2006, from http://search.cpan.org/~pasky/Graph-Layderer-0.02/lib/Graph/Layouter.pm
41. BlogDigger. Accessed August 2006, from http://www.blogdigger.com
42. Bloglines. Accessed August 2006, from http://www.bloglines.com
43. BlogPulse. Accessed January 2006, from http://www.blogpulse.com
44. Cinecultist. Accessed January 2006, from http://www.cinecultist.com
45. Cinemarati. Accessed January 2006, from http://www.cinemarati.com
46. Cinematical. Accessed January 2006, from http://www.cinematical.com
47. Emacs Wiki. Accessed October 2005, from http://www.emacswiki.org/cgi-bin/wiki
48. Google. Accessed August 2006, from http://www.google.com

49. Internet Movie Database (IMDB). Accessed January 2006, from http://www.imdb.com
50. Javapedia. Accessed May 2006, from http://wiki.java.net/bin/view/Javapedia/WebHome
51. LinuxWiki. Accessed May 2006, from http://www.linuxwiki.org
52. Meatball Wiki. Accessed October 2005, from http://www.usemod.com/cgi-bin/mb.pl
53. MediaWiki. Accessed December 2005, from http://www.mediawiki.org/
54. Nielson BuzzMetrics. Accessed January 2006, from http://www.nielsenbuzzmetrics.com
55. Portland Pattern Repository's Wiki. Accessed October 2005, from http://c2.com/cgi/wiki?WelcomeVisitors
56. Semantic MediaWiki. Accessed May 2006, from http://wiki.ontoworld.org
57. Technorati. Accessed August 2006, from http://www.technorati.com
58. TikiWiki. Accessed May 2006, from http://TikiWiki.org
59. TikiWiki For Education. Accessed May 2006, from http://edu.tikiwiki.org
60. The Graphic Organizer. Accessed August 2006, from http://www.graphic.org
61. The Movie Blog. Accessed January 2006, from http://www.themovieblog.com
62. Topic Maps. Accessed May 2005, from http://www.topicmaps.org/xtm/1.0
63. Topic Maps 4 Java. Accessed May 2005, from http://tm4j.org/
64. TouchGraph LLC. Accessed May 2005, from http://www.touchgraph.com
65. TWiki. Accessed May 2006, from http://www.TWiki.org
66. Wikimedia Commons. Accessed August 2006, from http://commons.wikimedia.org
67. Wikipedia. Accessed May 2005, from http://www.wikipedia.com
68. WikiTravel. Accessed May 2006, from http://www.wikitravel.org
69. WikiWikiWeb: Wiki History. Accessed August 2006, from http://c2.com/cgi/wiki?WikiHistory
70. WorldMagBlog. Accessed January 2006, from http://www.worldmagblog.com