Purple MediaWiki: Fine-Grained Addressability of Wiki Content

Kenneth Baclawski, Viral Gupta, Tejas Parikh
College of Computer and Information Science
Northeastern University
Boston, MA 02453, USA.
kenb, viral, tparikh@ccs.neu.edu

Peter P. Yim and Jonathan Cheyer
CIM Engineering, Inc. (CIM3)
San Mateo, CA 94402, USA.
peter.yim@cim3.com, jonathan.cheyer@cim3.com

Abstract

Purple MediaWiki (PMWX) is an extension of MediaWiki that supports fine-grained addressability. By making this feature available on MediaWiki it will be easier to support the many applications that require or can benefit from fine-grained addressability. The PMWX project engaged in a detailed study of related efforts, prepared a list of requirements, and developed a system architecture. The PMWX project has also developed a reference implementation that will be available as open source software. This paper reports on this project, including the architectural and design decisions that were considered.

Keywords: wiki, purple numbers, transclusion, collaborative work environments, fine-grained addressability, high-resolution addressability, indexing

1 Introduction

This paper describes Purple MediaWiki (PMWX), an extension to be integrated into MediaWiki that allows fine-grained addressability to the content of wiki pages. PMWX achieves its goal of fine-grained addressability by adding identifiers called "purple numbers" at the end of content sections on each wiki page. Unlike other web pages, content on a wiki is the result of a collaboration among the users of the wiki. As a result, the content on a wiki changes more frequently than most web pages. As more and more people add content to a web page and then refer to that content, it becomes important to pinpoint the location of the data for future reference or to provide a reference to someone else. Users generally do this by bookmarking a page for future reference or by sending a link to the article. The bookmarking option in a web browser allows one to bookmark the URL, but if this URL is the page as a whole it may be difficult for a user to locate the intended content when the amount of content on the page is large.

HTML allows one to create anchors to specific points in a web document. Using these links one can link to a particular point within a web document. The idea of directly accessing the information within a web document is an important hallmark of a knowledge management system. To build a system which empowers the user to access information precisely either the web site administrator or the author of the web document must manually create appropriate anchors within every web document. PMWX was developed to eliminate the need for web page authors to create these anchors. This allows a content developer to focus on developing the content; links to different parts of the document will be created and added automatically.

In this article, we talk about the history of Purple Numbers in Section 2.2, and some traditional uses of fine-grained addressability in Section 2.1. We then discuss the requirements that an implementation of Purple Numbers should satisfy in Section 3. The architecture and design are presented in Section 4, and we give an overview of the reference implementation in Section 5. There have been many implementations of Purple Numbers, including some for wikis. We give a review of these efforts in Section 6. In Section 7 we discuss some open research issues, we conclude in Section 8.

2 Background

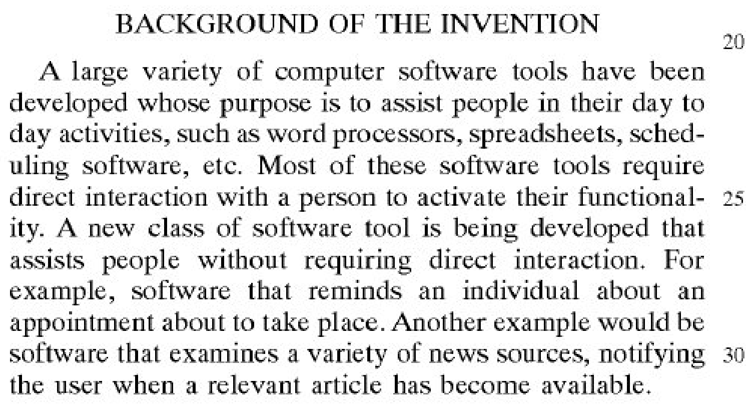
Fine-grained addressability has a long history, going back many centuries. Perhaps the oldest examples are in religious texts such as the Torah, Bible and Qur'ān which are indexed down to the level of individual verses. The Qur'ān has an especially complex organizational structure, and there are several competing divisions into verses.

In this section we discuss some of many traditional uses of fine-grained addressability. We then give a brief history of the notion of Purple Number which is the basis for our introduction of fine-grained addressability into MediaWiki.

2.1 Traditional uses of fine-grained addressability

There are many domains that make use of fine-grained text and image addressability. In the legal domain, laws and regulations are labeled with hierarchical identifiers, and these identifiers become terms in their own right. For example, officially recognized tax-exempt charitable organizations in the United States are often called 501(c)(3) organizations even in non-legal contexts.

Another important example of fine-grained addressability is in patents. In the United States patents make use of line numbers for fine-grained references to the text of the document as seen in Figure 1. This excerpt and those in Figures 2 and 3 were taken from [1].



BACKGROUND OF THE INVENTION

A large variety of computer software tools have been developed whose purpose is to assist people in their day to day activities, such as word processors, spreadsheets, scheduling software, etc. Most of these software tools require direct interaction with a person to activate their functionality. A new class of software tool is being developed that assists people without requiring direct interaction. For example, software that reminds an individual about an appointment about to take place. Another example would be software that examines a variety of news sources, notifying the user when a relevant article has become available.

Figure 1: Line numbers in a US patent

The line numbers are locally unique on each page of the patent. In addition to fine-grained addressability of text, all drawings in US Patents must have every element of the drawing labeled with an identifier as shown in Figure 2.

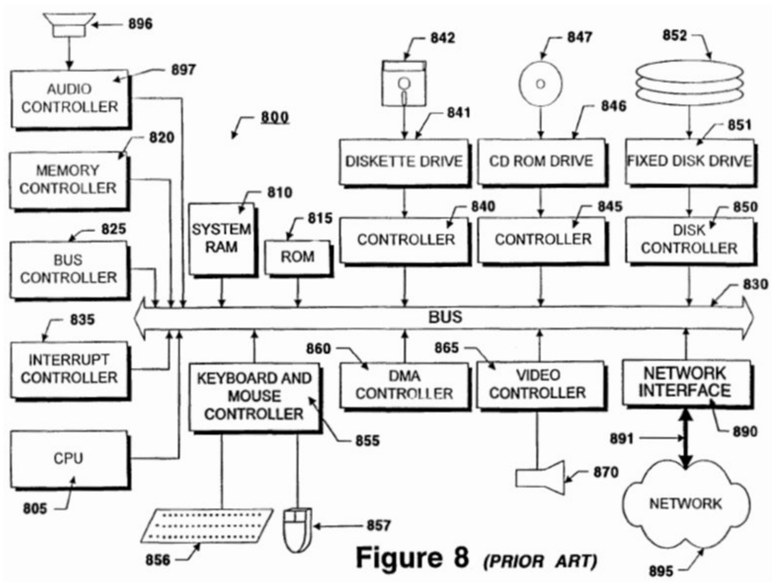


Figure 2: Diagram labels in a US patent

The identifiers for drawings are globally unique within the patent. The patent text refers to the elements of the drawings by using these identifiers as shown in Figure 3

Scientific research papers, such as this one, use a mix of hierarchical and sequential identifiers. The precise style that is used depends on the scientific domain. While the text of the document is organized hierarchically, other items such as figures, equations, theorems and literature citations often use sequential numbering.

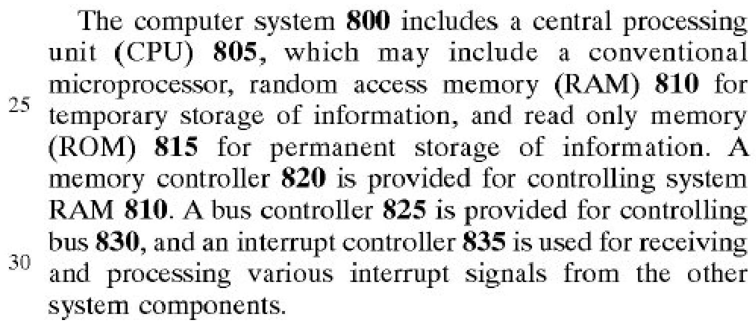

The computer system **800** includes a central processing unit (CPU) **805**, which may include a conventional microprocessor, random access memory (RAM) **810** for temporary storage of information, and read only memory (ROM) **815** for permanent storage of information. A memory controller **820** is provided for controlling system RAM **810**. A bus controller **825** is provided for controlling bus **830**, and an interrupt controller **835** is used for receiving and processing various interrupt signals from the other system components.

Figure 3: References to diagram labels in a US patent

Government and corporate archives are increasingly being digitized and indexed. There are also several large-scale efforts by libraries to digitize books that are no longer protected by copyrights. The Encyclopedia of Life [2] is an example of such an effort. These efforts include both the original images of the documents and information extraction using OCR techniques. Fine-grained identifiers will be of increasing important to link extracted information with its source.

Standards commonly employ fine-grained addressability. This is especially important for the standards review process which can involve a large number of organizations and people. For example in the standards review process employed by UN/CEFACT [3], each line in the technical specification is given a line number. Reviewers use these line numbers to refer to the topic of interest in their reports and communications.

In general, when a document is illocutionary (i.e., performs a function beyond just being a narrative), then precise identifiers serve an important role.

2.2 History of Purple Numbers

The concept of a "Purple Number" has its roots in the "oNLine System" (NLS) [4], which was a revolutionary collaboration system designed by Douglas Engelbart and his team in the Augmentation Research Center (ARC) at the Stanford Research Institute (now "SRI International") during the 1960s and 1970s. NLS was the first to employ the practical use of hyperlinked documents (hyperdocs), the mouse (co-invented by Engelbart and colleague Bill English), raster-scan video monitors, information organized by relevance, screen windowing, computer presentation, and other modern computer concepts. The ARC team used NLS to collaborate in ways that are just now becoming available with today's Web 2.0 social networking software. NLS was subsequently renamed as the AUGMENT system when it was commercialized.

The first use of "Purple Numbers" on web pages can be traced back to the mid-1990s when Doug Engelbart along with Bob Czech and Christina Engelbart at the Bootstrap Institute [5, 4, 6], came up with the notion of placing "Statement Numbers" on page elements such as headers, paragraphs and figures. The intention was to provide "Precision Browsing," mimicking the location number feature of the AUGMENT system. Christina, who developed the Bootstrap website, made those “Statement Numbers” purple in color, and as a result, “Purple Numbers” got its name. They were, however, just labels for reference purposes at the time.

Frode Hegland, who worked with Doug Engelbart around the time of the Bootstrap “Un-Rev2” Colloquium at Stanford (Q1/2000), suggested some major enhancements to the earlier implementation of the web-based Purple Numbers. In particular, the Statement Numbers were made active. With that, purple numbers are associated with the link information of the anchor to the particular element (heading, paragraph, figure, etc.), and this capability is supported in all current implementations.

Today, there are two kinds of Purple Numbers. A *hierarchical identifier* (HID) is the current name for a Statement Number. HIDs are stateless and give hierarchical information about the document element. A *node identifier* (NID) or statement identifier is a unique identifier for a document element (or “node”) that is independent of the placement of the element within the hierarchy of elements in the document. For a more detailed history of Purple Numbers see [5] and the links on this site.

Encompassed in Doug Engelbart’s ”bootstrap” philosophy [6] is the notion of a networked improvement community collaborating to develop a collective intelligence by improving on an improvement infratsructure. In particular, a virtual community using a collaboration tool to develop and continuously improve on their collaboration and their collaboration tools is one instantiation of ”bootstrapping”. Adhering to the ”bootstrap” philosophy, this paper, as well as the PMWX project, has been using a PMWX-enabled MediaWiki site both for the project [7] and the writing of the paper [8].

Purple Numbers are currently being used successfully in research, academia, government and commercial setting.

One such deployment is in the Ontolog community’s Collaborative Work Environment hosted on the CIM3 infrastructure. Ontolog (a.k.a. “Ontolog Forum”) is an open, international, virtual community of practice devoted to advancing the field of ontology, ontological engineering and semantic technology, and advocating their adoption into mainstream applications and international standards [9]. One of the co-authors of this paper, Peter Yim, is the founder of CIM3, as well as one of the founders of the Ontolog Forum.

There are a number of examples of very successful projects in (the US) government that made use of Purple Numbers. Susan Turnbull of the GSA Office of Intergovernmental Solutions has conducted a series of Collaborative Expedition Workshops with multiple Communities of Practice to advance government-to-government and government-to-citizen collaboration [10]. Another example was in their use to augment the development of the (US) Federal Enterprise Architecture, Data Reference Model v2.0 standard [11]. That development activity involved more than 300 documents, 585 people in 8 teams, and 5 workshops. This standard had an impact on virtually every US government agency. The new standard was developed in just 6 months, a pace that is rarely achieved in standards development activity. In both of these projects, Purple Numbers served as a mechanism for rapidly organizing a very complex series of discussions and negotiations.

3 Requirements

In this section we report on our conclusions regarding the key features required by Purple Numbers. We discuss the architecture, design and reference implementation in Sections 4 and 5.

3.1 Hierarchical Identifiers

Hierarchical Identifiers (HID) are Purple Numbers that represent a way to identify nodes in a structured document. As the name implies, each node in a structured document can be classified by its hierarchical position in the document. HIDs correspond to the "statement numbers" in the NLS system. As a node is added, removed, or reordered in the document, the hierarchical position of that node changes relative to other nodes. This change in the hierarchy is reflected by assigning a new HID to the node based on its position in the hierarchy of the document. The HIDs of other nodes may also change. This brings us to the main requirement of HIDs; namely that they must be stateless. A Purple Number is assigned to the hierarchy of the node and not the node itself. An HID is assigned to every node in the document that has a unique parent node. The second requirement is that HIDs should be unique for any given page.

3.2 Node Identifiers

Node Identifiers (NID) are Purple Numbers very similar to the "statement identifiers" of the NLS system. An NID goes beyond specifying a hierarchical location to a nodes or furnishing an anchor for the node. The key requirement of a Node Identifier is that it must be stateful. Once an NID is assigned to particular node in the document the NID remains with that node for the lifetime of the node. To ensure this property, NIDs are stored in a database together with the document to preserve their state. The key distinguishing point between HIDs and NIDs is that an HID specifies the hierarchical location of a node whereas an NID is assigned to the node itself.

Each time the wiki page is edited and saved, the content of the wiki page is scanned for new nodes. Each new node gets assigned a new unique NID which will identify that particular node only. One note that we would like to make at this point is that when a node is edited, e.g., a line is added in middle of a paragraph which already has NID assigned, it should not be considered to be a new node even though the content in that node has changed. If a user wishes to assign a new NID to a node whose content is changed, the user must delete the NID assigned to that node. A new NID will be assigned when the page is saved. In particular, when a user deletes a node, the NID for the node should also be deleted.

Another requirement of NID is that it is only unique to the page and not to the entire wiki. This requirement differs from the convention used in PurpleWiki. The reason for this requirement is to ensure that the use of NIDs in a large wiki with many pages and many concurrent users will not encounter performance problems. Creating unique NIDs for the whole wiki degrades the performance of the wiki because every update must compete for the "next NID" data value in the database.

3.3 Viewspec

A common complaint about Purple Numbers is that they can be a distraction or may even be regarded as intrusive. This is especially true of data-intensive wiki pages. Users may complain even though they like the added capabilities that fine-grained addressability gives them. To address this issue we decided that PMWX must have the ability to switch between showing and hiding Purple Numbers. We call this functionality *Viewspec*. This feature must be available both at the system administration and user levels. Thus Purple Numbers should be visible only when the administrator and user wants them to be. Another reason we provide this functionality is to avoid any confusion that Purple Numbers can create for new users who might not want to make use of this feature during their initial visits.

3.4 Transclusion

Our fourth requirement has to do with the support for transclusion. Transclusion is defined as the inclusion of the content of a document into another document by reference. The goal of transclusion is to get the latest referential data. In other words, if the referenced page is changed, then the transcluded information will be updated on the page even when the page was not otherwise changed. To support transclusion, there must be a consistent way of specifying the content which will be transcluded when one refers to either an HID or NID. MediaWiki currently supports transclusion up to a certain level. Users can transclude a whole page or a section of the page. An extension was also developed that would allow users to transclude any node that they want provided that markup has been added to the section for this purpose. Purple Numbers should provide this additional markup without any effort by the creator of the page. In addition, advanced users should be able to transclude a node from a particular version of the page if they do not want the transcluded data to change or if they wish to refer to an older version of the page. This feature is similar to a "cut-and-paste" operation except that the transcluded information is not duplicated in the database.

4 System Architecture and Design

4.1 System Architecture

There are multiple ways in which Purple Numbers can be added to MediaWiki. In this section we give the architectural details of how PMWX integrates with MediaWiki. We then discuss design aspects, especially the design decisions we had to address.

Purple Numbers can be added to the wiki content in two ways, either by a server-side script or a client-side script. Client-side scripting can be used when Purple Numbers can be generated on the fly every time page is rendered. It is also necessary for scripting to be available and enabled on the client web browser. While HIDs could be added by a client-side script because they are generated every time the page is rendered, the criterion may not always be satisfied. Accordingly, it is more appropriate to use server-side scripts. Furthermore, NIDs are stateful, so they could never be supported with a client-side script. For the sake of uniformity in the architecture and design and also to allow for users that may not have client-side scripting, we decided to use only server-side scripting. The only feature that is appropriate for client-side scripting is Viewspec.

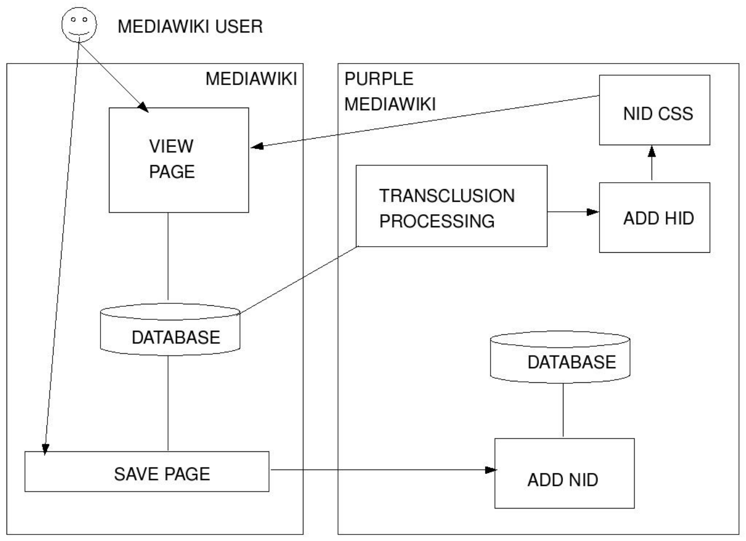


Figure 4: PMWX System Architecture

The system architecture for PMWX is shown in Figure 4. The left-hand side represents the functionality of MediaWiki that is used by PMWX. The right-hand side of the figure shows the main parts of PMWX. Because NIDs are stateful (persistent), they are stored in a database. This database would normally use the same database server as the one used by MediaWiki, but they can be different if desired. As shown in the diagram, PMWX adds transclusions, HIDs and NIDs in a pipeline whenever a page is viewed. Like MediaWiki, PMWX is written in PHP and has been tested using the MySQL database server. Support for other databases will be added later.

4.1.1 Hierarchical Identifiers

The notion of a hierarchical identifier is one of schemes of achieving fine-grained addressability. Hierarchical identifiers point to a particular location on a page. As we mentioned in the requirements, HIDs are not immutable i.e. they are assigned to the hierarchical location of the node and not the node itself. If a node moves, its hierarchical information changes and thus its assigned HID changes. HIDs are useful when a page is static or is not changed very often. HIDs are added at the end of the node in a special font using a purple color.

HIDs map logically to the physical layout of the document, making it easier to understand. For example HID value (4E8) will always point to the eighth sub statement of the fifth sub statement of the fourth statement on the first level of the document. Note, however, that this hierarchy is the one perceived by the user, not the HTML element containment hierarchy. This is discussed in more detail in Section 5.1.

4.1.2 The HID Numbering Scheme

PMWX provides two different numbering schemes for HIDs. The scheme to be used can be selected at the time of installation or later by the administrator who installs and maintains PMWX.

The first numbering scheme is the NLS numbering scheme for HIDs. In the NLS numbering scheme, the number for the HID starts with a digit followed by a letter in the alphabet, followed by a digit, and so on. In this scheme, the first node of the document will be numbered "(1)." The first child of this node will be numbered "(1A)" and the first child of this child node will be numbered "(1A1)".

The second numbering scheme is exactly the opposite of the NLS numbering scheme; namely, the roles of letters and digits are reversed. In this scheme, the first node of the document will be numbered "(A)". The first child of this node will be numbered "(A1)" and the first child of this child node will be numbered "(A1A)".

4.1.3 Node Identifiers

NIDs are very useful for fine-grained addressability on dynamic documents because they are are assigned to a particular node in the wiki page and they stay with it for its lifetime. NIDs should stay with the node even if the node is moved around in the document. However, they should not be moved with the node when the node is being copy-pasted into another document, as NIDs are unique to that page only and such an operation could create duplicate NIDs on the other page. Currently we do not have a mechanism for detecting NID duplication but it can be integrated into the extension at a later stage.

NIDs looks similar to HIDs, and when both are being shown, the NID comes before the HID, which follows the node content it is identifying. The shade of purple color used by the NID helps the user distinguish NIDs from HIDs. Figure 5 shows a simple wiki page that has both HIDs and NIDs.

4.1.4 The NID Numbering Scheme

There is only one numbering scheme available for NIDs in PMWX. The numbering schemes for NIDs follow what is effectively a base-62 numbering system whose "digits" are digits, lower-case letters and upper-case letters (in this order). NIDs are incremented sequentially, and only the largest NID used on each page is stored in the database. This allows one to obtain the next NID whenever a new one is required.

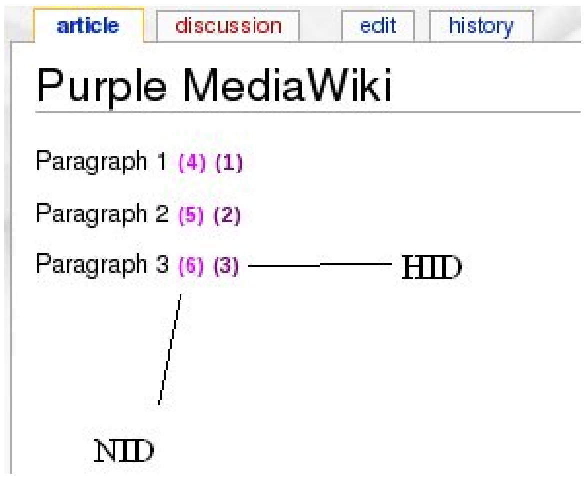


Figure 5: Screen capture of a simple wiki page with Purple Numbers

5 Reference Implementation

The reference implementation developed by the PMWX project uses the object-oriented capabilities of PHP. The class diagram is shown in Figure 6. Note that NIDs and HIDs were decoupled from one another to allow an administrator the option of supporting just NIDs or just HIDs on the system. In the class diagram for HIDs, each type of element has its own subclass. This was done so that the assignment of HIDs at the top level can treat all types of elements uniformly. The specific behavior for each type of element is encapsulated in its class. This is not necessary for NIDs because there is no element type specific behavior in this case.

Viewspec is implemented in JavaScript to provide a user with the option to switch between showing and hiding HIDs and NIDs. If the user's web browser does not have a script capability, or if the user has disabled the script feature, then the user will not be able to change the setting established by the administrator.

5.1 The HID Implementation

HIDs are generated every time the page is requested by the client. HIDs are added after MediaWiki has converted the wiki markup into HTML markup. This was done to avoid the need for parsing wiki markup since MediaWiki already does this. It has the added bonus that the HTML provided to the PMWX HID processor has a structure that is more easily analyzed. HIDs are assigned to any node that affects the hierarchy of the wiki page as perceived by the user. As we mentioned earlier, each node that is to be assigned an HID must be structured properly. It must either be a child of the overall wiki page or be a child of another node. Nodes (in the HTML markup) that are assigned an HID convey user-perceived hierarchical information for rather than presentation information. For example,

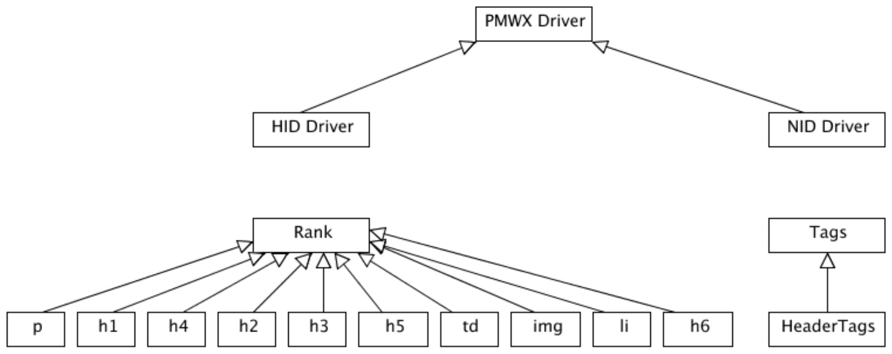


Figure 6: Class Diagram of the PMWX Implementation

HTML elements with the tags `<p>`, `<li>` and `<h1>` to `<h6>` have hierarchical significance to the user viewing the page, but elements with tags such as `<b>` and `<i>` only affect the font of the text and are not normally perceived as being of hierarchical significance.

HIDs are assigned as intuitively as possible. For example, `<h1>` hierarchically supersedes the `<h2>`-`<h6>`, `<p>`, `<img>`, `<li>`, `<td>` tags, and similarly `<h6>` supersedes the `<p>`, `<img>`, `<li>`, `<td>` tags. Consider this example of a small part of the HTML of a wiki page:

```
<h1>Heading 1</h1>
<p> Paragraph </p>
<h2> Heading 2 </h2>
```

Here the `<p>` tag is the child of `<h1>` and so is `<h2>`. The HIDs assigned to the above elements will be (1), (1A) and (1B), respectively. Now if we have the HTML structure shown here:

```
<h1>Heading 1 </h1>
<p> Paragraph 1</p>
<h2> Heading 2 </h2>
<p> Paragraph 2</p>
```

then the HIDs assigned to the nodes above will be (1), (1A), (1B) and (1B1), respectively. The numbering scheme is implemented using a local ranking array. Each tag gets its HID by locating the nearest parent node. This operation depends on the type of element, so it was implemented by using a polymorphic method of the subclass corresponding to the element type.

5.2 The NID Implementation

The NID implementation differs from that of HID, because NIDs are stored in the database along with wiki content. NIDs have to implemented in such a way that every node that is assigned an NID is also assigned an HID but the difference is that while the HID parser has the luxury of parsing HTML tags, the NID parser cannot avoid parsing the wiki markup. The NID parser is a bit less complicated compared with the HID parse due to fact that the NID parser is not concerned about the hierarchy. The NID parser simply needs to identify the nodes which should be assigned an NID and then assign one. Consequently, the getNextNID() function is much simpler than the getNextHID() function because all that is needed is to increment the most recently used NID by one. NIDs are stored along the content of the wiki page and can be easily identified when editing the page, as in the following example:

```
This is a paragraph in a wiki page. <nid value="1">
```

The only complication is in how one deals with the NIDs for Header tags. The WikiMedia parser does not convert header markup to HTML if there is an HTML tag on the same line. For example, the wiki markup `== The World == <nid value="1"/>` would not be parsed into `<h1>The World</h1> (1)` as one would expect. Placing the NID inside the header like this `== The World <nid value="1"/> ==` would be undesirable since headers are used in the table of contents and other contexts in which Purple Numbers would not be appropriate. To deal with this difficulty, we handle this special case in a different way. We add the NID to the next line of the document when it is stored in the database, but when the page is rendered for viewing by the user, we adjust the HTML for the header so that the NID is on the same line as the header.

6 Related Work

There have been many attempts to develop tools to support and to popularize Purple Numbers. These tools were an important influence on the design and development of PMWX. In this section we give a brief overview of the various tools that have support for Purple Numbers in online documents such as web pages, blogs and wikis.

6.1 XLink

XLink allows elements to be inserted into XML documents in order to create and describe links between resources [12]. XLink was created by Jon Bosak and Tim Bray toward the end of the 1990's. XLink is a powerful mechanism for linking XML documents. XLink allows one to specify bidirectional links, embedded links (i.e., transclusion) and links that have more than two targets. When combined with XPointer [13] and XPath [14], XLink can link and transclude parts of documents, with granularity down to the level of individual characters.

6.2 Purple

Purple [15] is a small suite of quickly hacked tools inspired by Doug Engelbart's attempt to bootstrap the addressing features of his Augment system onto HTML pages. Its purpose is simple: produce HTML documents that can be addressed at the paragraph level. It does this by automatically creating name anchors with static and hierarchical addresses at the beginning of each text node, and by displaying these addresses as links at the end of each text node. Purple is relatively easy to use. The flip side is that it was built for static webpages consisting mainly of text, the web page developer has to run the tool on each web page before it is uploaded on the website as well as whenever it is updated.

6.3 Plink

Murray Altheim came up with an implementation similar in function to Purple (which was mainly in perl), but done in Java around the same time (April 2001) [16].

6.4 PurpleSlurple

Matthew Schneider developed the first HID implementation which generates purple numbers on-the-fly for HTML and text documents which he published in 2002. He had actually worked on flavors of PurpleSlurple for other document formats like word doc's and pdf's, but they did not seem to have been officially released [17].

6.5 PurpleWiki

PurpleWiki [18] is a WikiWikiWeb implementation derived from UseModWiki. It was written by Eugene Kim of Blue Oxen Associates and Chris Dent, and was first released in January 2003. It adds several features to Purple, and modularizes the code for easier development. In addition to support for Purple Numbers, it has a parser that supports pluggable output formats, RSS feeds of recent changes, and transclusion of content between pages. The downside of PurpleWiki is that it only supports NIDs and lacks many features that MediaWiki provides.

6.6 Purple Numbering on Blogs

Tim Bray was the first to put purple numbers (purple hash marks which acted as permalinks, to be exact) to a blog, back in May 2004 [19]. One of the co-authors of this paper, Jonathan Cheyer, has written a plugin to add Purple Numbers for Wordpress [20]. It only supports HIDs. NID are not supported because permanent node identifiers are not stored for each paragraph.

6.7 HyperScope

HyperScope has been a project that Doug Engelbart has been driving over the last decade or so, as part of his OHS effort [21]. A more recent effort has been the NSF funded project (with development work done by Eugene Kim, Brad Neuberg, Jonathan Cheyer et al.) which implements a subset of the functionality of the original NLS system mapped onto the modern-day web paradigm. In particular, it supports many of the original NLS viewspecs for viewing documents in different ways. It also supports a number of jump commands which allow you to move between different nodes in a document. The power of the viewspecs and jump commands is that they can be embedded directly into a URL. This allows a user to pass along URLs to another user which point to specific locations in a document with a particular view of that document. The HyperScope site describes the tool as a "high-performance thought processor that enables you to navigate, view, and link to documents in sophisticated ways."

7 Future Work

The requirements and architecture for transclusion have been developed, but we have not yet designed and implemented it. Support for general XPath expressions adds a significant degree of complexity to transclusion. The plan is to begin with transclusion for NIDs which is relatively easy to add. We will later add support for HIDs. One of the fundamental drawbacks of transclusion using HIDs is that HIDs depend on the structure of the document which can change as the document is updated. We intend to allow users to specify a version number for a transclusion which can mitigate this problem to some extent. Finally, we will add support for XPath expressions. XPath expressions gives the extra power to transclude any element on the web. Thus one can transclude parts of a document that are not addressed by either HIDs or NIDs alone and one can transclude parts of documents that do not have fine-grained accessibility at all, such as documents that are not in a wiki.

Because there is currently a great deal of wiki content in PurpleWiki and other implementations of purple numbers, we plan to develop tools to migrate the content to PMWX. Differences in formats and conventions complicate migration, especially if the history of documents is to be migrated as well as the current version.

Once the infrastructure for fine-grained accessibility is in place, we plan to begin work on applications that build on this infrastructure. One such application is the notion of a semantic wiki [22]. Our plan is to support annotations using either folksonomic tagging (such as the del.icio.us web site) or formal ontologies written in RDF or OWL. Existing semantic wikis and tagging mechanisms are limited to annotations at the document level. Fine-grained identifiers allow one to annotate at a much more precise level. We have already developed an initial prototype wiki that allows one to tag purple numbers in the PurpleWiki. We plan to develop a WikiMedia version of this prototype and to extend it to more powerful annotation capabilities.

8 Conclusion

An extension of MediaWiki that supports fine-grained addressability was developed, which we call Purple MediaWiki Extension or PMWX. The intention was to remain firmly in the spirit of the Doug Engelbart's vision for fine-grained addressability, but also to be as compatible as possible with the needs of MediaWiki developers and users. We made a case that fine-grained addressability has been a feature of many types of documents for centuries and that many applications today can benefit from it.

The PMWX project engaged in a detailed study of related efforts and prepared a list of requirements. One outcome of this process was the conclusion that both hierarchical and persistent identifiers have their uses and that both should be supported at the same time, but that users could choose to hide them if desired, a feature we call Viewspec. The project then developed a system architecture for supporting both HIDs and NIDs as well as transclusion and Viewspec. We then prepared a detailed design, addressed the design issues that arose, and reported on these design decisions. Finally, to establish that the architecture and design are realistic and effective, we developed a reference implementation. In "bootstrap" fashion we used the reference implementation as the project proceeded to continue the development effort.

While we do not have user experience other than our own with PMWX, related systems do have such experience. These experiences provide evidence that fine-grained addressability in general, and Purple Numbers in particular, provide important capabilities for many large-scale and time-limited collaboration efforts. By making fine-grained addressability available to MediaWiki, these capabilities will be more generally available to collaborative work environments.

References

[1] K. Baclawski. Distributed computer database system and method, February 20 2001. United States Patent No. 6,192,364. Assigned to Jarg Corporation, Waltham.

[2] EOL staff. Encyclopedia of Life, 2008. `http://www.eol.org`.

[3] UN/CEFACT. Core components technical specification (CCTS) version 3: Second public review., 2007. `http://xml.coverpages.org/ni2007-04-20-a.html`.

[4] K. Gust. References from the NLS/AUGMENT project at the Computer History Museum, 2006. `http://www.softwarepreservation.org/projects/nlsproject`.

[5] P. Yim and C. Engelbart. Introduction and brief history of Purple Numbers, 2008. `http://community.cim3.net/cgi-bin/wiki.pl?PurpleNumbers`.

[6] D. Engelbart. The bootstrap vision and mission, 1999. `http://bootstrap.cim3.net/vision_mission.html` and `http://www.bootstrap.org`.

[7] K. Baclawski et al. PMWX Project homepage on Purple MediWiki, 2007–2008. `http://project.cim3.net/wiki/PMWX`.

[8] K. Baclawski et al. Work-in-progress version of this paper on Purple MediaWiki, 2008. `http://project.cim3.net/w/index.php?title=PMWX&oldid=906#hid1B2`.

[9] P. Yim et al. Ontolog collaborative work environment site, 2002–2008. `http://ontolog.cim3.net/wiki`.

[10] S. Turnbull. USA Services: COLAB-collaborative-work-environment, 2008. http://colab.cim3.net.

[11] M. Daconta, S. Turnbull, M. McCaffery, et al. Process for community and public comments on the FEA-DRM document, 2008. http://colab.cim3.net/cgi-bin/wiki.pl?DataReferenceModel#nid2RH3 and http://colab.cim3.net/cgi-bin/wiki.pl?DataReferenceModel.

[12] S. DeRose, E. Maler, and D. Orchard. XML linking language (XLink) version 1.0, 2001. http://www.w3.org/TR/xlink.

[13] S. DeRose, R. Daniel, P. Grosso, E. Maler, J. Marsh, and N. Walsh. XML pointer language (XPointer), 2002. http://www.w3.org/TR/xptr.

[14] J. Clark and S. DeRose. XML path language (XPath) version 1.0, 1999. http://www.w3.org/TR/xpath.

[15] E. Kim. Purple web site, 2003. http://www.eekim.com/software/purple.

[16] M. Altheim. Plink web site, 2001. http://www.bootstrap.org/dkr/ohs-dev/0588.html and http://collab.blueoxen.net/forums/tools-yak/2004-02/msg00085.html.

[17] M. Schneider. Purpleslurple web site, 2002. http://www.purpleslurple.net.

[18] Blue Oxen Associates. PURPLEWIKI web site, 2007. http://www.blueoxen.com/tools/purplewiki.

[19] T. Bray. Purple number signs, 2004. http://www.tbray.org/ongoing/When/200x/2004/05/29/PurpleNumbers.

[20] J. Cheyer. Purple Number Wordpress plugin site, 2007. http://cms.cheyer.biz/software/purple.

[21] D. Engelbart. Draft OHS-Project Plan, 2000. http://www.bootstrap.org/augdocs/bi-2120.html.

[22] M. Krötzsch, S. Page, and D. Vrandecic. Semantic Media Wiki, 2008. http://ontoworld.org/wiki/Semantic_MediaWiki.