Revisiting Wikis: A Modular Structure for Textual Information Connectivity on Web

Chayapatr Archiwaranguprok
The School of Science and Technology
University of the Thai Chamber of Commerce
Bangkok, Thailand
pub@from.pub

Manachai Toahchoodee
The School of Science and Technology
University of the Thai Chamber of Commerce
Bangkok, Thailand
manachai_toa@utcc.ac.th

***Abstract*—As the advancement of the web and related technologies makes it possible to exceed the initial limitation imposed by the early-age World Wide Web, this paper concerns the revisitation of concepts and structure on textual data connectivity in wiki systems based on ideas around hypertext.**

***Index Terms*—Component-Based Open Hypermedia System, Wiki, Web Architecture**

I. Introduction

Since its birth at the end of the 20th century, World Wide Web has served as a large-scale information system utilizing the underlying internet infrastructure at ease by providing a relatively simple user interface in the form of documents referring to each other. Even though the system's design was motivated by the hypertext paradigm, it has sparked arguments within the field of whether it is a hypertext system or not, mainly for its data-centric approach, which the system does not, in fact, possess information of the structure, i.e., the linking technology based on hyperlinks and URIs is unidirectional, in contrast to the bidirectional in traditional hypertext systems.

However, this low-demand unidirectional workaround for the web has enabled the system to scale massively into a global network. Despite the disagreement, the hypertext-inspired notion contributes to the methodologies and views for handling interconnected data for the masses and, importantly, emphasizes the delinearization of textual data. While the technology has been diversely adopted, the core function of information organization remains vital; wikis have been one important practical implementation of the conceptual framework of interconnected textual information repositories based on the web infrastructure since WikiWikiWeb and Wikipedia were created in 1994 and 2001 [1] respectively. In the early 2000s, the concept of web semantics [2] focused on machine intelligibility and structured connections, leading to the development of semantic wikis [3], which enhanced traditional wiki interconnectivity with structural semantics.

The growing popularity of the web, in turn, refocuses hypertext research to circulate around the web. As a result, modern discussions around hypertextual connectivities are grounded around the limitation imposed by the limitation of web-based linkage, overshadowing other meaningful patterns. Yet, the advancement of the web and related technology in the recent decade makes it possible to exceed the common practices of the subject. By reexamining these foundational ideas through the lens of current technological capabilities, we aim to explore new possibilities for extending the structure and interconnectivity of wiki as a form of textual information repository while contribute to the ongoing dialogue about the around such topics for further developments.

II. Architecture for the System

To tackle the common practice of web functionalities, we first examine the discussion around hypertext structure outside the web sphere, which "can be described as successive steps of abstracting and opening up parts of their architecture" [4], from monolithic infrastructure which the layers, including application, link mechanism, and store are tied into one large system, toward the modular architecture that decouples the components and opens up for incorporation of different structure paradigms. One important development is the design of the Component-based Open Hypermedia System (CB-OHS) [5], which offers a philosophical design rationale for a system that "asserts the primacy of structure over data." [6] by separating the three layers and mechanisms within the layers. Importantly, the link mechanism is opened into an open set of "structure servers" for different functionalities. Conceptually inspired by CB-OHS, this section projects the rationale into real-world web technology to derive an architecture for wiki connectivities.

A. A Structure in Three Layers

Not only does the separation of layers lead to the ease of incorporation of different structures, but in the actual world, its modularity can address the fast-paced development of web technology by making it relatively simple to plug in, swap, remove, or update the module. In our design, the minimalistic core of our wiki system consists of three main layers, reflecting that of CB-OHS, as illustrated in Fig. 1. The components, including a possible minimalistic implementation, are described in subsection §II-B, § II-C, and §II-D.

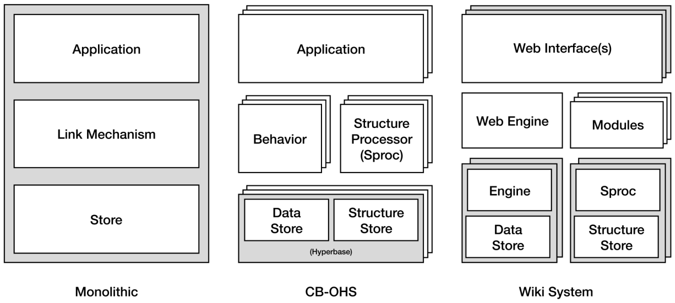

Fig. 1. Overall architecture of the Systems

B. Application Layer

This comprises the web interface(s) for the end user. The implementation can be trivially implemented using front-end web technologies and frameworks. Yet its design serves a crucial role in the usability of the system. For example, the choice of language used for textual information is directly linked to the ease of contribution. Most wikis use a markup language for representation; however, many modern applications use markdown syntax provided through the WYSIWYG editor, which, while reducing the complexity and possibility of representation, should be relatively more affable for end-users. The markdown texts are parsed through a parser for a web render in HTML format. Still, it is reasonably simple to swap this core to markup-based representation. The textual data are store in a component describe in §II-D.

C. Engine Layer

This layer refers to arbitrary services crafted to serve different purposes for different use cases. Implementation-wise, the system adopts a modular design, which can be plainly implemented as microservice architecture. Each non-core functionality is designed as a module that operates as an independent service. This facilitates the ease of updates, scalability, and maintenance. In simple cases, it may function as a medium for CRUD operations between the application and storage layers and as a platform for other functionalities, including the search engine.

D. Storage Layer

As grounded in CB-OHS, the storage is separated into two types: (1) Data Store and (2) Structure Store.

1) Data Store: In the context of wikis, this refers to a core textual repository for the wiki system. As mentioned in §II-B, the documents are in markup or markdown language. We may adopt the common practice, for example, that of MediaWiki, [7] for data storage, where the data is stored within a relational database [8]. As illustrated, another dedicated engine module may be plugged in to facilitate several exceptional use cases, but this is not commonly required.

2) Structure Store: Looking against the early-age constraint of the web, modern technologies can be integrated to extend the connectivity of the application, which has been criticized as data-centric, by providing structure information to the system [1]. CB-OHS introduced the notion of a structure processor (sproc) for the structural operation of the system. While it was a stand-alone component in CB-OHS, we reiterated the notion and combined it with a structure store, reflecting practicality for the adoption.

III. On Three Layers of Structural Connectivity

This section demonstrated a possible model derived from structure store (§II-D2) to extend the extend information connectivity of the core system. These modules can arguably be perceived as three hierarchical layers of system connectivity, i.e., Overtly (User-defined) Discrete (§III-A) → Covertly (Auto-generated) Discrete (§III-B) → Continuous (§III-C).

A. Extended Hyperlinks

Hyperlinks are the fundamental feature of the web and have been functioning as a core linkage for the system. Yet, as mentioned, the hyperlinks are unidirectional and possess no information, including the existence of the referred data. Incorporating structure stores and processors can generate structural information about the linkage, opening up several possibilities, including creating backlinks for entries that refer to each other, recognizing the existence of the referred data, and generating an illustrated site graph in the web interface. Implementation for such information varied, including relational database, triplestore, or simple JSON files.

B. Knowledge Graph

Semantic wiki extends traditional wiki systems by incorporating semantic notations into data points based on the web semantic framework. These structures provide a framework for representing information in a way that is both human-understandable and machine-processable. At the core of knowledge graph implementation is the Resource Description Framework (RDF), which expresses data as a set of triples, each consisting of a subject, predicate, and object. This enables the construction of semantic relationships from the knowledge encapsulated within the repository.

The integration of knowledge graphs into wiki systems enables more meaningful search and query capabilities that go beyond simple keyword matching or hyperlink navigation. However, coding these semantic relations is complex and subject to coders' inconsistency [9]. Maintaining consistency becomes increasingly challenging as the system scales, posing significant maintainability issues. Despite these challenges,

[1] It is necessary to highlight the difference between the whole world wide web system, and the system within the web, e.g., our wiki. While the criticism of the web being hypertext addresses the web itself, the constraints of the web still affect the common practices of systems implemented on the web

semantic wikis provide a powerful tool for more meaningful information organization [10].

Recent advancements in automatic knowledge graph generation [11] offer promising solutions to these scalability issues. Machine learning techniques, particularly in natural language processing, now allow for the automated extraction of entities and relationships from unstructured text. These technologies can significantly reduce the manual effort required to create and maintain knowledge graphs.richer data representation and understanding. In practice, when each page is edited, we can automatically generate a knowledge graph specific to that page, storing it in a triplestore with an RDF scheme based on a schema standards, e.g. schema.org [12]. We compare the generated knowledge graph with the existing database to assess sameness and compatibility. Conflicts may be flagged for manual resolution by users, ensuring data accuracy and consistency.

C. Text Embedding

Text embedding transforms textual data into continuous vector forms, bridging the gap between its discrete nature and machine learning models. Early methods like Bag-of-Words (BOW) ignored sentence context and excluded unknown words, missing neologisms, and misspellings. Recent transformer-based models with attention mechanisms [13] have significantly improved sentence embeddings by better context capturing. Similar to KG, each textual entry can be embedded, utilizing transformer-based language model, e.g. BERT [14], into a point stored in a vector database. The embedding offers several new possibilities for wiki system, including semantic similarity search, recommendation, and automated categorization, which are further discussed in §IV.

IV. On Functionalities

A. Search and Recommendation Engine

While a separate search and recommendation based on each possible data structure is possible and sensible to implement, incorporating multiple structures in our wiki system enhances search and recommendation capabilities for the system but also introduces challenges in ranking and presenting search results. This multi-faceted approach to search can be viewed through the lens of information retrieval, particularly the challenge of effectively ranking heterogeneous data.

The basic implementation of the search engine involves fusing scores or ranks from queries across multiple storage components. This includes plain-text search in the data repository, knowledge graph-based search, and embedding-based similarity search. The main process occurs in the engine layer, which initiates searches in the storage layer and then aggregates and processes the results. The results are combined using fusion algorithm which can be as simple as score-based CombMNZ [15] to ML-based algorithm. We noted that the choice of fusion algorithm is to be contextually chosen based on use cases, that is, can be posthumously defined and fine-tuned by adopters.

The incorporation of multiple search methods can be rationalized through analogous use cases. For instance, the integration of text embedding and knowledge graphs in our system mirrors Google's search methodology, which combines a SNA-based PageRank algorithm with a Knowledge Graph. This approach yields robust and consensual information retrieval. Beyond its primary function, this incorporated search structure can also serve as a recommendation engine. By leveraging the multidimensional relationships established through several structures, the system can suggest relevant content to users based on their current context or historical interactions. This dual functionality transforms the search feature from a mere query-response tool into a proactive knowledge discovery mechanism. For instance, as users navigate through wiki pages, the system can recommend related content that may not be explicitly linked but is semantically connected.

B. Auto-tagging and Categorization

Similar to knowledge graph generation, tagging and categorization are subjected to scalability caused by the tedious work required and coders' inconsistency. Yet, also similar to the mentioned possibilities and functionalities, it is possible to design an auto-tagging and categorization system based on storage components. The autonomy of this introduced process also results in the possibility of on-spot dynamically generated categorization based on usage contexts.

V. Possibilities and Future Works

This paper presents an architecture for a textual wiki system that incorporates discussion around a hypertextual linkage structure to extend the connectivity of information. While the paper is focused on wiki, the notion can still be developed to generalize overall web-based connectivities and relations. In addition, while the three layers introduced in §III are presented as a connected hierarchy with an implied idea of data flow, it is helpful to note that the systems are, in fact, modular and open to changes in structure based on use cases, including a spatial [16] or taxonomic [17] hypertext, while the many ideas are aged, it can still be useful when reiterating in contemporary contexts. The future works for our wiki system are primarily regarding (1) the completion of surrounding components, e.g., authentication and version control, and emphasis on scaling factor, e.g., the more optimized design for storage and query, and (2) the extension of structural concepts and modules.

The minimalistic system presented here is roughly a proof-of-concept focus on relation, yet more work needs to be done for the entire working system that scales. In addition, for such a system presented in this paper, a web-based architecture that serves end-user performance testing methods in information retrieval or management through user testing needs to be revised or more meaningful. The need for the design of a suitable test methodology is exerted.

References

[1] G. Moody, "This time, it'll be a Wikipedia written by experts — theguardian.com," https://www.theguardian.com/technology/2006/jul/13/media.newmedia, [Accessed 20-05-2024].

[2] T. BERNERS-LEE, J. HENDLER, and O. LASSILA, "The semantic web," *Scientific American*, vol. 284, no. 5, pp. 34–43, 2001. [Online]. Available: http://www.jstor.org/stable/26059207

[3] S. Schaffert, A. Gruber, and R. Westenthaler, "A semantic wiki for collaborative knowledge formation," 01 2005.

[4] C. Atzenbeck, T. Schedel, M. Tzagarakis, D. Roßner, and L. Mages, "Revisiting hypertext infrastructure," in *Proceedings of the 28th ACM Conference on Hypertext and Social Media*, ser. HT '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 35–44. [Online]. Available: https://doi.org/10.1145/3078714.3078718

[5] D. B.-L. C. A. P. Peter J. Nürnberg, Kaj Grønbæk and O. Reinert, "A component-based open hypermedia approach to integrating structure services," *New Review of Hypermedia and Multimedia*, vol. 5, no. 1, pp. 179–205, 1999. [Online]. Available: https://doi.org/10.1080/13614569908914713

[6] P. J. Nürnberg, J. J. Leggett, and E. R. Schneider, "As we should have thought," in *Proceedings of the eighth ACM conference on Hypertext*, 1997, pp. 96–101.

[7] "Manual:Managing data in MediaWiki - MediaWiki — mediawiki.org," https://www.mediawiki.org/wiki/Manual:Managing_data_in_MediaWiki, [Accessed 25-05-2024].

[8] "Manual:Database layout - MediaWiki — mediawiki.org," https://www.mediawiki.org/wiki/Manual:Database_layout, [Accessed 24-05-2024].

[9] O. Hassanzadeh, "Building a knowledge graph of events and consequences using wikipedia and wikidata," in *Proceedings of the Wiki Workshop at The Web Conference*, 2022.

[10] K. Kutt and G. J. Nalepa, "Loki – the semantic wiki for collaborative knowledge engineering," *Expert Systems with Applications*, vol. 224, p. 119968, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417423004700

[11] L. Zhong, J. Wu, Q. Li, H. Peng, and X. Wu, "A comprehensive survey on automatic knowledge graph construction," 2023.

[12] R. V. Guha, D. Brickley, and S. Macbeth, "Schema.org: evolution of structured data on the web," *Commun. ACM*, vol. 59, no. 2, p. 44–51, jan 2016. [Online]. Available: https://doi.org/10.1145/2844544

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017. [Online]. Available: http://arxiv.org/abs/1706.03762

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[15] E. Fox and J. Shaw, "Combination of multiple searches," *NIST special publication SP*, pp. 243–243, 1994.

[16] C. C. Marshall and F. M. Shipman, "Spatial hypertext: designing for change," *Commun. ACM*, vol. 38, no. 8, p. 88–97, aug 1995. [Online]. Available: https://doi.org/10.1145/208344.208350

[17] H. Van Dyke Parunak, "Don't link me in: set based hypermedia for taxonomic reasoning," in *Proceedings of the Third Annual ACM Conference on Hypertext*, ser. HYPERTEXT '91. New York, NY, USA: Association for Computing Machinery, 1991, p. 233–242. [Online]. Available: https://doi.org/10.1145/122974.122998