A Knowledge Management System based on Wikis from a Software Architecture Perspective

Luciano Straccia*, Brenda Baruzzini, Benjamín Alegre and María F. Pollo-Cattaneo

Universidad Tecnológica Nacional, Buenos Aires, Argentina

Abstract

A wiki enables collaboration among team members and is an important resource for knowledge management. A wiki is not a complete knowledge management system (KMS), but it is one of its most widely used components. The objective of this article is to consider the use of wikis as part of a knowledge management system from a software architecture perspective and to propose a KMS based on wikis, considering other needed components. The proposal architecture is based on a model with the following layers: presentation, access and authorization, knowledge management, taxonomy and knowledge maps, information management and repository, and shows details for each layer.

Keywords

Knowledge Management, Wiki, Architecture, KMS

1. Introduction

Previous research carried out by the research group explores different perspectives on knowledge management (KM) [1, 2] and analyzes the application of technology for knowledge management [3, 4, 5]. A wiki is an important resource for KM, which enables collaboration among team members and allows knowledge exploitation and representation [6, 7]. The wiki is well-established in the literature but has been little studied from a Software Architecture perspective [8]. The architecture of knowledge management systems (KMS) is the bridge between knowledge management consultants and technologists, providing the possibility of implementing, in a technological and reliable way, the ideas proposed from a management-oriented perspective [9]. A wiki is not a complete KMS, but rather a part of it. Wikis are discussed in the literature as an isolated component, without exploring their potential integration with other tools that enable more efficient use. Authors such as Cordova and Cerchione [10, 11] even consider wikis to be user interface tools.

2. Objectives and Methodology

This work accepts as a hypothesis the wide use of wikis for knowledge management and raises the need to expand the results that wikis provide through the incorporation of other tools. The objective of this article is to propose the use of wikis as part of a knowledge management system from a software architecture perspective. Section 3 presents the theoretical framework, and a discussion of the software architecture topics for KMS is presented in Section 4. The Section 5 presents the results (a model for KMS based on wikis from a Software Architecture perspective); for identifying the needs of architecture, the Architecture Description Record is used and for documenting the final proposal, C4 and UML notation are used. Finally, the Section 6 presents the conclusions, and Section 7, outlines future works.

ICAIW 2025: Workshops at the 8th International Conference on Applied Informatics 2025, October 8–11, 2025, Ben Guerir, Morocco

*Corresponding author.

lstraccia@frba.utn.edu.ar (L. Straccia); bbaruzzini@frba.utn.edu.ar (B. Baruzzini); balegrerocha@frba.utn.edu.ar (B. Alegre); fpollo@frba.utn.edu.ar (M. F. Pollo-Cattaneo)

0000-0002-1183-7944 (L. Straccia); 0000-0003-4197-3880 (M. F. Pollo-Cattaneo)

3. Theoretical Framework

The basic concepts about knowledge and Knowledge Management are introduced in 3.1, and in the following sections are presented the concepts related to technologies: wikis 3.2 and software architecture 3.3.

3.1. Knowledge and Knowledge Management

Knowledge is introduced in the DIKW Hierarchy [12] whose base element is data; one level higher, information; later, knowledge; and at the top level, wisdom. Knowledge is the mixture of cognitive and contextualized beliefs, perspectives, judgments, methodologies, information, experiences and expectations made about an object, which are adapted and potentiated by the mind of an individual (knower) [13], a mixture of structured experiences, values and non-contextual information that provides a framework for evaluating new experiences and information [14]. The intellectual capabilities of human beings, cultural values, skills, experience, including mental models, can become a powerful tool to create value for the company [15]. Knowledge is the most significant resource of a company, making it essential to know how to acquire and manage it [16].

Knowledge management (KM) is a process through which organizations manage to discover, use, and maintain knowledge to align it with business strategies [17] and is based on a process of social influence of collaborative groups around the transfer of knowledge to take advantage of the experience and skills of talents and leaders [18].

3.2. Wiki

A wiki is a technological tool that enables collaboration among team members. The wiki concept was created in 1995 by Ward Cunningham and was originally associated with a content repository [19]. A wiki is "a virtual community, whose pages are edited directly from the browser, where the users themselves create, modify, correct, or delete content that is usually shared by any other user."[1] Wikis are therefore technologies for communication, documentation, and dissemination of knowledge, becoming spaces for the development of online content that allow users to create, edit, and distribute information [20]. A wiki allows knowledge to be shared freely among the workers of an organization, allowing the creation of a knowledge spiral with the accumulation of different contributions over time. A wiki used in companies "become a centralized way of exchanging ideas in work groups and a new model of knowledge management, facilitating the organization of everything that employees know and that needs to be stored and systematized so that knowledge is not lost" [21, 22].

Wikis have the following fundamental principles: anyone can modify, they use hypertext markup language, they are flexible (they do not have a predefined structure), and the content has "no time and no end", that is, it is timeless, it is never finished, and is being modified by new ideas. For Cunningham [19], the principles of a wiki are simplicity, integration, being an open system, and allowing exponential growth. Ebersbach [23] presents several characteristics that wikis should have.

Müller et al. [24] analyze the impact of wikis for KM, finding the following aspects: knowledge is free and sharable, content is evolving according to the dynamism of knowledge and its context, few barriers to knowledge use, no definition of knowledge management roles is necessary, the origin of content development is verifiable, and knowledge emerges. Annahi [25] suggests the following issues to be considered for the implementation of a corporate wiki: encouraging employees to participate (building critical mass), building culture (having a collaborative culture), ensuring updating (the wiki must be constantly updated to avoid being outdated), administration (there must be a manager responsible for the content, although all people have access and the possibility of participation) and basic research about the existing wiki software and its selection.

[1]Wikipedia https://es.wikipedia.org/wiki/Wikipedia:Portada

3.3. Software Architecture and Architecture Decisions

The software architecture of a system is the set of structures needed to reason about the system and comprises the software elements, the relationships between them, and the properties of both [26, 27]; it is the fundamental organization of a system, formed by its components, the relationships between them, the context in which they will be implemented and the principles that guide their design and evolution[2].

The term "software architecture" was proposed by Perry and Wolf [28] based on original ideas from Dijkstra, Wirth, De Remer, Kron, and Shaw, among others. Through understanding architectures, robust, scalable systems can be built that are agile enough to adapt to technological changes, address security and control issues [29], and present and resolve conflicts between technology quality attributes, including traditional notions of stability and control with the needs for speed and flexibility [30, 31].

An architecture decision (AD) is a software design choice that addresses a significant requirement. An architecture decision record (ADR) is a document that captures an important architectural decision made along with its context and consequences. An architecture decision record (ADR) is one of the most important deliverables of a solution architect, and a record should include consistent elements such as: problem statement with context, options considered, decision outcome, and include important tradeoffs made with this decision, and record the confidence level of the decision [32, 33]. Google proposes use ADR when there are two or more engineering options and you want to document your thoughts and reasons behind the selection and must include: authors and the team, context and problem you want to solve, functional and non-functional requirements you want to address, potential critical user journey (CUJ) the decision impacts, overview of the key options and the decision and reasons behind the accepted choice [34]. Project members should create an ADR for every architecturally significant decision that affects the software project or product, including the following [35, 36]: structure, non-functional, dependencies, interfaces, and construction techniques. The AWS Prescriptive Guidance says that "at a minimum, each ADR has to define the context of the decision, the decision itself, and the consequences of the decision for the project and its deliverables" [37]. There are various ADR templates proposed in the literature: Markdown Architectural Decision Records [38], Nygard ADR [39], and Y-Statement [40]. The Y-Statement template has the following elements: context, facing, decision, neglected alternatives, benefits, drawbacks, and other consequences; for space reasons, this article uses a partial notation of the Y-Statement.

4. Discussion and ADRs

4.1. General Model for KMS Based on Wikis

A wiki is relevant for interaction with knowledge management users [10, 11], but the analysis of a complete KMS requires a broader structure. Architectures such as the one proposed by Kerchsberg [41] discuss wikis or knowledge portals as just one component within an entire system and include other components organized through layered architectures. Following the ideas of the C4 model [42], the first level of an architecture model is the definition of the system context. The context diagram is a high-level overview of how users interact with the system and allows for the high-level definition of internal and external systems; this model defines the people involved and the systems at a high level. Then, the first decisions appear and they are documented through these ADRs:

ADR 1: In the context of the general model for KMS, facing the need to define users related to the system, we decided that the actors involved are knowledge workers and knowledge managers (based on [2]) **ADR 2:** In the context of the general model for KMS facing the need to keep Wikis as main component for users we decided identify three systems in particular: the wiki itself (the initial object of analysis in this document), the external systems with which the wiki will be integrated, and a specific system (which at the end of this document will be called Wiki+) that will integrate all the functions that allow knowledge to be exploited to the maximum and will incorporate all integration responsibilities.

[2]IEEE Std. 1471 https://standards.ieee.org

4.2. WIKI+ Architecture and Layers

Component-Based Software Engineering (CBSE) approach is based on the production of various pieces of software assembled in an integrated software system. CBSE enables software reuse, simplified testing, simplified system maintenance, and higher component quality [43]. The first decisions about the architecture are presented in the following ADRs:

ADR 3: In the context of WIKI+ Architecture, facing the need to define the fundamental principles of its architecture, we decided to use a Component-Based Software Engineering (CBSE) approach.

Some authors present proposals regarding the layers of architecture for knowledge management systems. Kerchsberg [41] presents a technological architecture for knowledge management based on the 3-Layer Model: Presentation, Knowledge Management, and Data. In the presentation layer, knowledge workers communicate and share knowledge through the knowledge portal. The KM layer is where the activities are followed to acquire, refine, store, distribute, and present knowledge; this layer provides information to the data layer and in turn feeds on it.

Borghoff et al. [44] propose a four-component structure: knowledge repositories and stores; knowledge worker communities; knowledge flow; knowledge mapping. Tiwana [45] presents a seven-layer architecture: interface, access and authentication, collaborative intelligence and filtering, application, transport, middleware, and legacy integration, and repositories [46]. Rance [47] presents a four-layer architecture: data, information integration, knowledge processing, and presentation.

There is a consensus about the use of a presentation layer, a data layer, and intermediate components, but the scope of these intermediate components is not clear, even in the differentiation between those specifically associated with knowledge management, others associated with information management, and taxonomies or semantics. Lawton [48] proposes the following layers in its Ovum model: Knowledge Management, Taxonomy, Knowledge Maps, and Process and Information Management. Tiwana uses two layers: Collaborative Filtering and Intelligence and Application. Finally, Kerschberg incorporates everything in a single layer. Tiwana innovates with the proposal of an access and authentication system.

ADR 4: In the context of WIKI+ Architecture, facing the need to define the fundamental principles of its architecture, we decided to use a Layer Architecture Pattern. **ADR 5:** In the context of WIKI+ Architecture, facing the need to define their layers, we decided to consider the following layers: presentation, access and authorization, knowledge management, taxonomy and knowledge maps, information management, and repository (based on previous proposals [3]).

4.3. Presentation Layer

Presentation components allow a user to interact with the Wiki knowledge. The Visual Wiki model proposed by Hirsch [49] is appropriate for the definition of presentation components. Hirsch presents a proposal for knowledge access with special emphasis on visualization (which Kerchsberg includes in the presentation layer) which “based on mapping rules, resource objects are translated into visual objects as meaningful representations, offering easy and comprehensive access to the subject matter presented” [50] and introduces the concept “Visual Wiki” defined as a “combination or integration of two representations (a textual and a visual one) of the same underlying body of knowledge” with passive and active visualizations. Passive visualizations use existing aspects or features of a system, such as tags or page connections in a wiki, by visualizing them; examples are tag clouds, mashups using wikis and Google Maps or graphs. The resulting visualizations are passive in the sense that they are created automatically. On the other hand, active visualizations are realized as applications that allow the user to create diagrams and integrate them. For Hirsch, "the major problem of existing approaches to realizing the concept of a Visual Wiki seems to be a lack of integration between the visual and the textual representations. In other words, the approaches are either too active or too passive".

Hirsch presents an architecture model with four components: visualization, mapping, text, and concept. The visualization and text components correspond to the distinct types of knowledge representations mentioned above, while the mapping component favors the integration between both types. Regarding data visualization, [3] states that to accomplish this task, techniques are used to present information graphically, such as bar charts, pie charts, time series, or any graphical representation that meets the goal of enabling clear and efficient communication. [51] propose knowledge graphs that organize data from multiple sources, capture information about entities of interest in each domain or task (like people, places, or events), and forge connections between them. Finally, based on Hirsch, the component called concept, which "underlies all of the others, consists of two parameters: purpose and content of the Visual Wiki". Regarding external visualization tools, for example, Martin [52] proposes D3JS, Chart.js, and Google Charts to obtain a better view of the evolution of the use of articles in wikis.

ADR 6: In the context of Presentation Layer for WIKI+ Architecture, facing the need to define their components, we decided to consider these components: visualization, mapping, text, and concept.

ADR 7: In the context of Presentation Layer for WIKI+ Architecture, facing the need to define their interface with Wikis, we decided to consider the concept component as the interface component with the Wiki.

ADR 8: In the context of Presentation Layer for WIKI+ Architecture, facing the need to define their integration with external tools, we decided to consider that the Visualization component must be integrated with external visualization tools.

4.4. Knowledge Management Layers

In the layered architecture presented, there are three fundamental layers associated with knowledge management: knowledge management, taxonomies and ontologies, and Information management. This subsection analyzes possible scopes to be included in these layers.

Wahl [50] presents a web semantic tool based on ontologies, Semantic MediaWiki (an extension of MediaWiki), and data import and transfer tools, and Gorton [53] proposes an architecture called GS3 for knowledge management for Collaborative Geologic Sequestration Modeling. Although it presents some aspects associated with the semantic web, data analysis, and content management, the "particular interest for GS3 is the semantic extensions that enable a range of dynamic views or 'mashups' of the content". It also presents a pipeline for data extraction and exploitation. For Gorton, the scientific data pipeline component is a data ingest pipeline that accepts published documents (PDF files being the dominant format) and analyzes the content against a dictionary of important terms. This pipeline provides an automated knowledge extraction capability, with the results available for searching, browsing, and inclusion in page mashups in the wiki.

ADR 9: In the context of Knowledge Management layers for WIKI+ Architecture facing the need to define their components, we decided to consider a pipeline for data extraction and exploitation, and it is included in the Information Management Layer.

Gavekar [54] presents technological components for a semantic web, integrating it with ontologies, knowledge management tools, and web services, but does not present further details about these components. Loia [55] also presents aspects associated with ontologies and their relationship with a KMS. Meanwhile, [56] presents a Semantic Wiki KnowWE, which provides the possibility to define and maintain ontologies, and [57] presents a semantic extension of MediaWiki.

Martin [52] proposes pattern mining on the information produced in the wiki; following Saxena et al. [58] presents techniques as association rules, cluster analysis, characterization and comparison, sequential pattern analysis, and trend analysis. Other authors, such as Astudillo, Torres, and Sevilla, present the relevance of Semantic Wiki [59, 60, 61]. A Semantic Wiki "has the same basic characteristics of a traditional Wiki but incorporates meta information into the markup language to give it meaning; this meta information is understandable by a machine" [60]; this concept is adapted from the "semantic web", a term proposed by Tim Berners-Lee.

Knowledge management requires knowledge structuring activities associated with web semantics as well. As stated by the creator and reference of the semantic web, [51], for the semantic web to function, computers must have access to structured collections of information and sets of inference rules that they can use to conduct automated reasoning. This semantics may include indexing, use of ontologies (explicit and formal specification of a concept within a given domain of interest [61]; [5] describes concepts related to ontologies), and taxonomies. A semantic wiki can be seen as a wiki including an associated ontology, i.e., an operational representation model of domain knowledge [60].

The literature presents ontologies as a specific case among the relevant knowledge management techniques, which have been included in this article as a layer of architecture. Ontologies, tools, and concepts are discussed in [5]. Wang [62] analyzed 1275 ontologies on the Web and found that most of them were in OWL, the most widely used for ontology implementation too [63], and that consists of three languages with increasing expressivity: OWL Lite, OWL DL, and OWL Full. Finally, Gao et al. [64] presented ontology-based knowledge systems. Rozo Rodriguez [65] analyzes different ontologies for the field of software quality control, such as STOWS, OntoTest, TaaS, TOM, ROoST, and the importance of building ontologies on specific domains can be observed.

ADR 10: In the context of Knowledge Management Layer for WIKI+ Architecture facing the need to define their components, we decided to include components for each of the following tasks: association rules, cluster analysis, characterization and comparison, sequential pattern analysis, trend analysis, and indexing.

ADR 11: In the context of the Knowledge Management Layer for WIKI+ Architecture, facing the need to integrate results produced for each component proposed in previous ADR, we decided to use a Plugin (also known as Microkernel) Architecture Pattern.

A Plugin Architecture Pattern is a structure with a Core System and different Plugins (insertable modules). The Core System contains all the minimum elements to ensure the functioning of the architecture, and the Plugins are add-on components associated with the Core that have a specific purpose.

ADR 12: In the context of Knowledge Management layers for WIKI+ Architecture facing the need to define their components, we decided to include relationships with ontology tools in the Taxonomy and Knowledge Maps layer.

4.5. Integrations

This subsection explores different possible integrations for wikis based on an analysis of existing documentation. Our team has conducted a detailed analysis of some wikis in recent years. For these tools, their integrations were analyzed to identify categories through open coding, where "all the segments of the material obtained for analysis are reviewed and it generates, by constant comparison, initial categories of meaning. It thus eliminates redundancy and develops evidence for the categories (raises the level of abstraction)" [66]. The categories found are:

1. Data analytics tools: they allow visualizing usage patterns, identifying relevant or obsolete content, and measuring contributions, e.g., Google Analytics, Grafana, Prometheus, Piwik, Segment.
2. Knowledge classification and processing tools: they facilitate the debugging, filtering, and restructuring of knowledge according to its usefulness. They allow us to classify semantically, identify redundancies, and extract tacit knowledge by means of NLP or text mining techniques.
3. Document Management Systems (DMS): complement wikis with rigorous version control, document approval flows, and regulatory compliance, e.g., Google Drive.
4. Automation and integration platforms: allow task automation and synchronization with other platforms, e.g,. API, Zapier, Webhooks.
5. Communication and notification tools: ensure that the knowledge shared in the wiki remains active and is disseminated to users in real time, e.g., Slack.
6. Project and task management tools: allow traceability between documented knowledge and derived actions, e.g., Jira, Trello, Asana, Redmine, Zoho Projects.
7. Development and Infrastructure tools: allow the integration of tools for software construction, version control, deployment, and other software development activities, e.g., Github, Bitbucket, Gitlab, Mercurial, Docker.

Some other tools were found that can be various information systems (CRM software, customer service platforms, for example) or other wikis (Confluence, for example). In addition, some wikis provide integration with LDAP (Lightweight Directory Access Protocol) and Active Directory, and Semantic MediaWiki (an extension of MediaWiki that allows semantic structuring of content in a wiki. It allows adding metadata and relationships to wiki pages to facilitate analysis and search).

The tools defined in the different categories presented must be included in an architecture that allows the scope and impact of wikis to be expanded. However, the relationship with Project and task management tools, and Development and Infrastructure apps will not be included since they correspond to a specific type of software for certain organizations or industries; each industry and each organization will have other integrations associated with its own activity. With respect to communication and notification tools, we believe it is appropriate to separate these two functions: communication and notification. The first function requires tools that capture conversations and information from users and that can produce data, information, and knowledge for the wiki; meanwhile, the second requires tools for sharing and disseminating the knowledge through the organization [2]. Then wikis must integrate with different types of tools, and it is necessary to define which layer corresponds to each of the proposed integrations.

ADR 13: In the context of KMS facing the need to define their integration with tools, we decided to define the following tool types: Data analytics tools, Knowledge classification and processing, Document Management Systems, Automation Platforms, Communication with Source Data, and Notification for Sharing and Dissemination.

ADR 14: In the context of KMS facing the need to define how layers must be integrated with each tool type, we decided that the data analytics tools must be related to the Presentation Layer.

ADR 15: In the context of KMS facing the need to define how layers must be integrated with each tool type, we decided that the Document Management Systems and Communication with Source Data tools must be related to the Repository Layer.

ADR 16: In the context of KMS facing the need to define how layers must be integrated with each tool type, we decided that the Knowledge Classification and Processing, Automation Platforms, Notification and Sharing, and Dissemination tools must be related to the Knowledge Management Layer.

5. Result

Based on the decisions presented in the previous sections and documented by the different ADRs, this section presents the resulting model.

5.1. General model

Based on ADRs 1 and 2, and following the first level proposed by the C4 model, called Context Level, the Figure 1 shows the general model for KMS based on wikis from a Software Architecture Perspective.

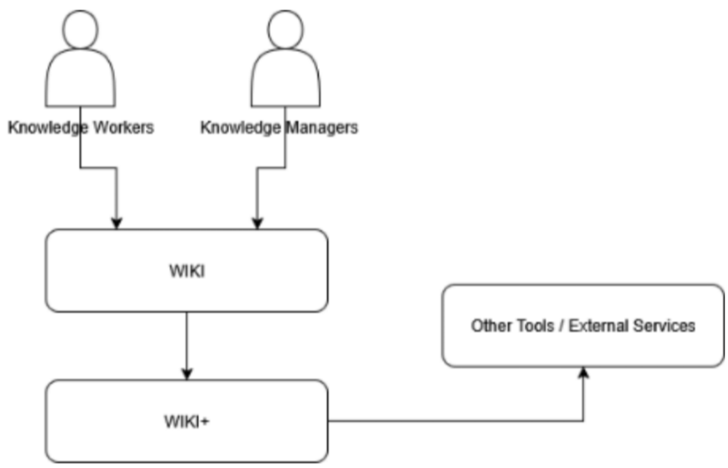

Figure 1: KMS – Level Context.

5.2. WIKI+ Architecture Model

Based on ADRs 4 and 5, the Figure 2 shows the layer proposal for WIKI+.

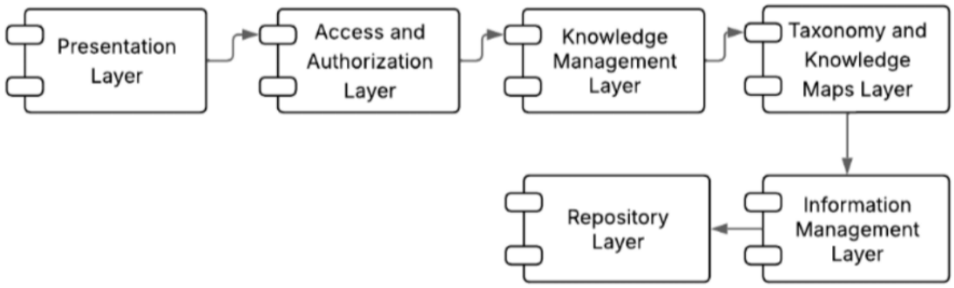

Figure 2: WIKI+ Architecture Model.

The components for the presentation layer, knowledge management layer, taxonomy and knowledge maps layer, and repository layer are detailed. The remaining layers are not detailed and are expected to be defined in a future version of the architecture, as detailed in section 7.

5.3. Presentation Layer

The Presentation layer components are presented in Figure 3, which shows the internal and external components based on ADRs 6, 7, 8, 13, and 14.

The visualization and text components correspond to the distinct types of knowledge representations, while the mapping component favors the integration between both types. The component called concept capture, purpose, and content of the Visual Wiki. The visualization component is responsible for integrating visualization and data analytics tools.

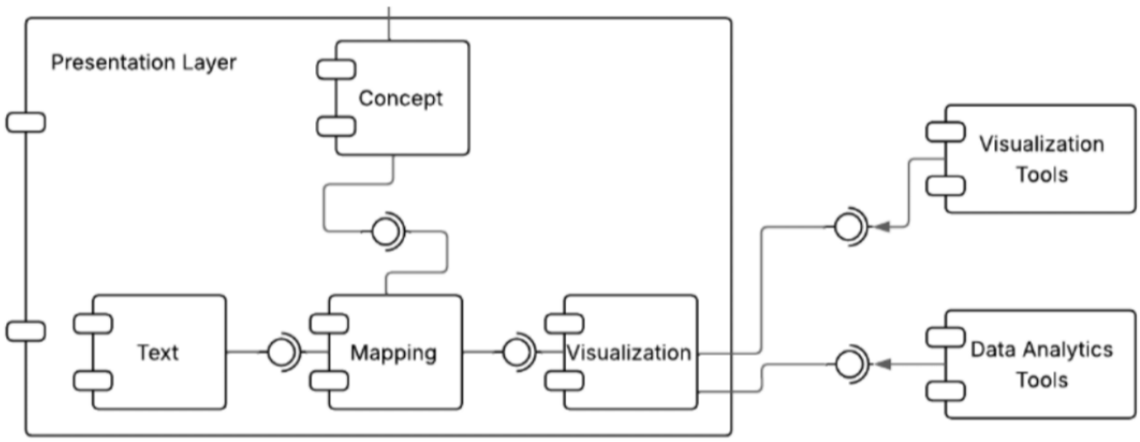

Figure 3: Presentation Layer Components.

5.4. Knowledge Management Layer

Based on ADRs 3, 10, 11, 13, and 16, the Figure 4 shows the internal and external components related to the Knowledge Management layer components. These components are related with some tasks and they are integrated through a Plugin Architecture Pattern.

5.5. Taxonomy and Knowledge Maps Layer

Based on ADR 12, the Figure 5 shows the internal and external components related to the Taxonomy and Knowledge Maps layer components. Although taxonomy and knowledge map tools were not included

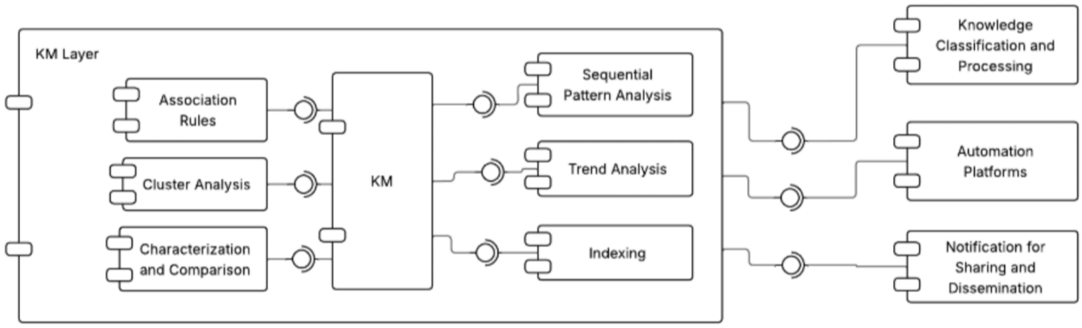

Figure 4: Knowledge Management Layer Components.

in the analysis in the previous section, it is understood from the name of the layer that they should be included. For this component, it is considered that most tasks are performed by external ad-hoc components.

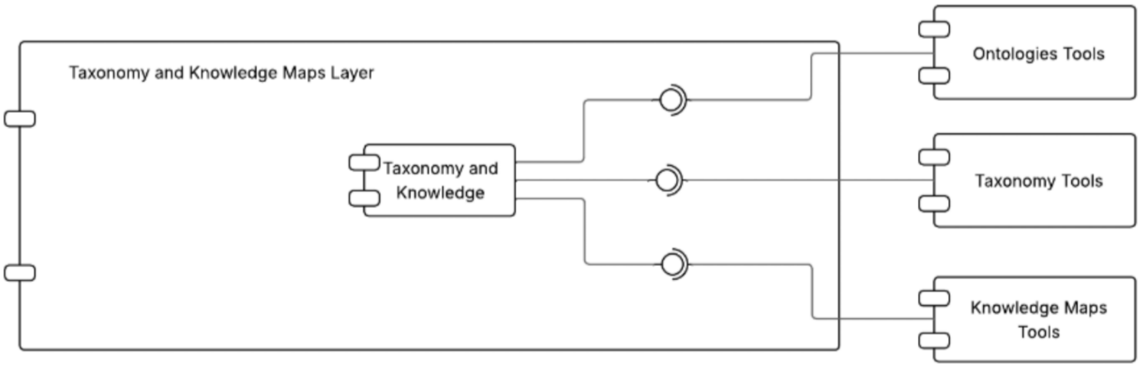

Figure 5: Taxonomy and Knowledge Maps Layer Components.

5.6. Information Management Layer

Based on ADR 9, the Figure 6 shows the internal and external components related to the Information Management layer components. Based in Wahl, we use a pipeline for data extraction and exploitation. The IM Component can integrate these layers with others.

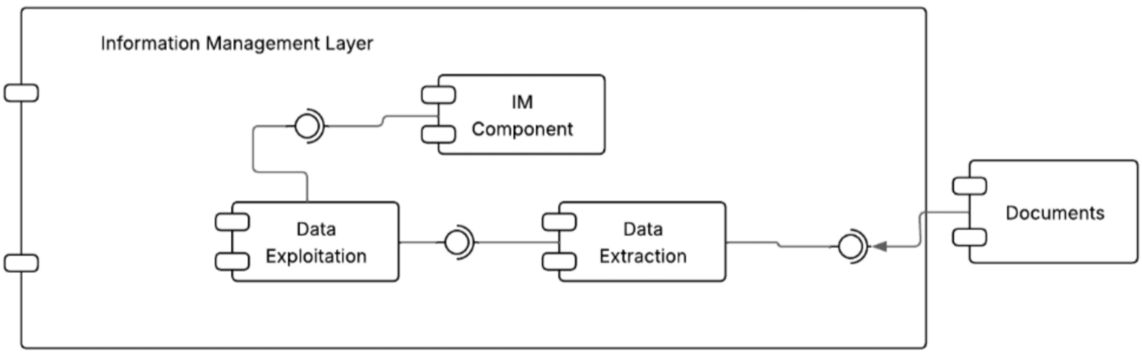

Figure 6: Information Management Layer Components.

5.7. Repository Layer

Based on ADRs 13 and 15, the Figure 7 shows the internal and external components related to the Repository layer components. For this layer, repositories are considered external, and the KMS has a component to integrate them.

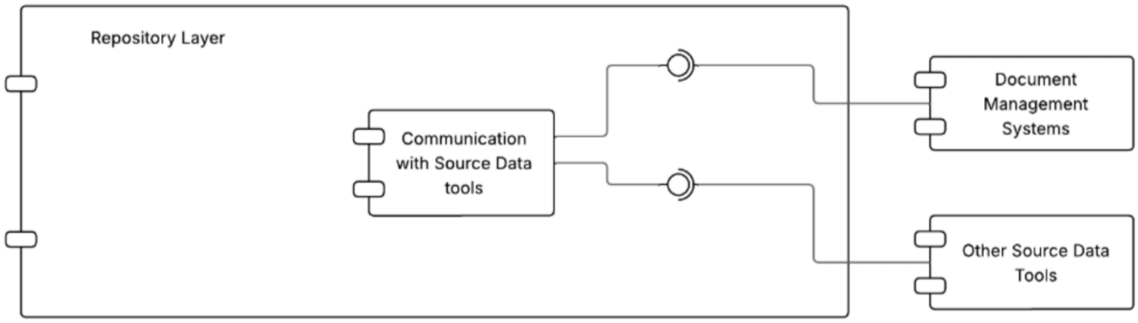

Figure 7: Repository Layer Components.

6. Conclusions

This paper proposes a Knowledge Management System (KMS) based on Wiki, considering the wikis as the main component from the user's perspective. This article provides a more comprehensive view of the use of wikis as part of a broader knowledge management system and recognizes the importance of integration with different tools. The KMS incorporates other components and relationships with external tools, using a Software Architecture Perspective and using Architecture Description Record (ADR), a C4 Model, and UML notation.

The KMS users are knowledge workers and knowledge managers, and it has three high-level components: a wiki, a new component called WIKI+ for this paper, and external tools. The architecture proposal follows the Component-Based Software Engineering (CBSE) approach and uses a Layer Architecture Pattern, considering the following layers: presentation, access and authorization, knowledge management, taxonomy and knowledge maps, information management, and repository. The Presentation Layer considers these subcomponents: visualization, mapping, text, and concept, and the Concept component has an interface with the Wiki, and the Visualization component is related to external tools, including visualization and data analytics tools. Knowledge Management is based on a Plugin Architecture Pattern with plugins for these tasks: association rules, cluster analysis, characterization and comparison, sequential pattern analysis, trend analysis, and indexing. In addition, this layer is integrated with external tools for Knowledge Classification and Processing, Automation Platforms, and Notification for Sharing and Dissemination. The Taxonomy and Knowledge Maps layer is related to ontologies (and external tools for ontologies), taxonomies, and knowledge maps. Finally, the Repository Layer has a component for Communication with Source Data, which is the interface with external tools, including Document Management Systems.

7. Future Works

This is the first version of a Knowledge Management System based on Wiki Architecture. In future analyses, we expect to further develop the analysis of the Access and Authorization Layer, considering concepts from information systems security. We also expect to expand on the details of the different components and interfaces of the layers presented, especially the Information Management Layer. Finally, the analysis of external tools for each of the defined categories should be expanded, identifying selection criteria and selecting different tools according to the different types of organizations.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] L. Straccia, M. Pollo-Cattaneo, A. Maulini, Visões da gestão do conhecimento para a construção de um modelo integral, in: Proceeding in Congresso Brasileiro de Gestão do Conhecimento, São Paulo, Brasil, 2022, pp. 19–22.

[2] L. Straccia, M. F. Pollo-Cattaneo, A. Maulini, Knowledge management model: A process view, in: International Conference on Computational Science and Its Applications, Springer, 2023, pp. 599–616.

[3] L. Straccia, M. F. P. Cattaneo, A. Maulini, Knowledge management technologies for a n-layered architecture., in: ICAI Workshops, 2021, pp. 117–130.

[4] L. Straccia, M. F. Pollo-Cattáneo, M. Giorda, M. G. Bongiorno, A. Maulini, Architecture on knowledge management systems: Its presence in the academic literature, in: International Conference on Applied Informatics, Springer, 2022, pp. 411–423.

[5] L. Straccia, M. F. Pollo-Cattaneo, Knowledge management: Innovation, technology and ontologies, 2024.

[6] S. S. Alkhir, Understanding the model driven architecture, published in "Methods & Tools" October (2003).

[7] C.-H. Yang, M.-Y. Wu, C.-M. Lin, D.-L. Yang, Implementation of wiki-based knowledge management systems for small research groups, in: 2008 Eighth International Conference on Intelligent Systems Design and Applications, volume 2, IEEE, 2008, pp. 346–349.

[8] L. Straccia, A. Maulini, M. G. Bongiorno, M. Giorda, M. F. P. Cattaneo, Knowledge representation and technologies in the latin american academic literature., in: ICAI Workshops, 2022, pp. 271–287.

[9] A. Chua, Knowledge management system architecture: a bridge between km consultants and technologists, International journal of information management 24 (2004) 87–98.

[10] Á. L. M. Cerdán, Tecnologías de información y gestión del conocimiento: integración en un sistema, Economía industrial (2005) 107–116.

[11] D. Pérez, M. Dressler, Tecnologías de la información para la gestión del conocimiento, Pérez, Daniel; Dressler, Matthias." Tecnologías de la información para la gestión del conocimiento". Intangible Capital, enero-marzo de 2007, vol. 3, núm. 15, p. 31-59. (2007).

[12] R. L. Ackoff, From data to wisdom, Journal of applied systems analysis 16 (1989) 3–9.

[13] M. T. R. Díaz, J. J. G. Millán, Gestión del conocimiento y capital intelectual, a través de modelos universitarios, Económicas Cuc 34 (2013) 85–116.

[14] T. H. Davenport, Knowledge management and the broader firm: strategy, advantage, and performance, Knowledge management handbook 2 (1999) 1–2.

[15] E. Ropain, L. Amaya, E. Arroyo, Factores de la gestión del conocimiento enel desarrollo de software, 3. GESTIÓN DEL RIESGO TECNOLÓGICO EN EL USO DE LA ENERGÍA FOTOVOLTAICA EN VENEZUELA....... 4 (2014) 520.

[16] P. F. Drucker, The post-capitalist world, The Public Interest 109 (1992) 89.

[17] E. B. Campos, Dirección del conocimiento y aprendizaje: creación, distribución y medición de intangibles, in: Perspectivas sobre dirección del conocimiento y capital intelectual, 2000, pp. 36–45.

[18] A. Sánchez-Sánchez, O. Valés-Ambrosio, C. García-Lirios, M. Amemiya-Ramirez, Confiabilidad y validez de un instrumento que mide la gestión del conocimiento, Espacios en blanco. Serie indagaciones 30 (2020) 1–10.

[19] B. Leuf, W. Cunningham, The Wiki way: quick collaboration on the Web, Addison-Wesley Longman Publishing Co., Inc., 2001.

[20] L. B. Angarita, Librería de activos para la gestión del conocimiento sobre procesos software: PAL-Wiki, Ph.D. thesis, Universidad Carlos III de Madrid, 2010.

[21] L. HARARI, R. SGANZERLA, Web 2.0–a era da participação e colaboração coletiva. 2011, Blog Roberto Sganzerla. Disponível em:. Acesso em 6 (2011).
[22] P. C. Rodrigues Borges, R. M. Drumond Rhaddour, A arquitetura da informação em plataformas colaborativas como suporte para a gestão da inteligência coletiva nas organizações, Biblios (2017) 62–72.
[23] A. Ebersbach, M. Glaser, R. Heigl, A. Warta, Wiki: web collaboration, Springer, 2006.
[24] C. Müller, P. Dibbern, Selbstorganisiertes wissensmanagement in unternehmen auf basis der wiki-technologie-ein anwendungsfall, HMD-Praxis der Wirtschaftsinformatik 252 (2006) 45–54.
[25] C. ANNAHI, ótimas dicas para um wiki de sucesso, Portal gestão de pessoas e tecnologia da informação (2015).
[26] F. Bachmann, L. Bass, D. Garlan, J. Ivers, R. Little, P. Merson, R. Nord, J. Stafford, Documenting Software Architectures: Views and Beyond, Addison-Wesley Professional, 2011.
[27] H. Florez, M. Sánchez, J. Villalobos, G. Vega, Coevolution assistance for enterprise architecture models, in: Proceedings of the 6th International Workshop on Models and Evolution, 2012, pp. 27–32.
[28] D. E. Perry, A. L. Wolf, Foundations for the study of software architecture, ACM SIGSOFT Software engineering notes 17 (1992) 40–52.
[29] O. M. Albada, N. Salman, Software architecture in practice: Challenges and opportunities in the age of digital transformation: Literature review, 2024.
[30] M. Rebouças, R. O. Santos, G. Pinto, F. Castor, How does contributors' involvement influence the build status of an open-source software project?, in: 2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR), IEEE, 2017, pp. 475–478.
[31] H. Florez, M. Sánchez, J. Villalobos, iarchimate: a tool for managing imperfection in enterprise models, in: 2014 IEEE 18th international enterprise distributed object computing conference workshops and demonstrations, IEEE, 2014, pp. 201–210.
[32] Learn Microsoft, Architecture decision record, 2024.
[33] H. Florez, M. Sánchez, J. Villalobos, A catalog of automated analysis methods for enterprise models, SpringerPlus 5 (2016) 406.
[34] Google Cloud, Architecture decision records overvie, 2024.
[35] Amazon Web Services, Adr process, 2022.
[36] M. Richards, N. Ford, Fundamentals of Software Architecture: A Modern Engineering Approach, " O'Reilly Media, Inc.", 2025.
[37] D. Kunce, D. Goby, Using architectural decision records to streamline technical decision-making for a software development project, 2022.
[38] O. Kopp, A. Armbruster, O. Zimmermann, Markdown architectural decision records: Format and tool support., in: ZEUS, 2018, pp. 55–62.
[39] M. Nygard, Documenting architecture decisions, NFJS One 187 (2011).
[40] O. Zimmermann, Y-statements, 2020.
[41] L. Kerschberg, D. J. Weishar, Conceptual models and architectures for advanced information systems, Applied Intelligence 13 (2000) 149–164.
[42] S. Brown, The c4 model for software architecture, Updated August 1 (2018).
[43] J. Casal Terreros, . MSDN Microsoft, Desarrollo de software basado en componentes, 2020.
[44] U. M. Borghoff, R. Pareschi, Information technology for knowledge management., J. Univers. Comput. Sci. 3 (1997) 835–842.
[45] A. Tiwana, The knowledge management toolkit: practical techniques for building a knowledge management system, Prentice hall PTR, 2000.
[46] L. Halawi, R. McCarthy, J. Aronson, Success stories in knowledge management systems, Issues in Information Systems 18 (2017) 64.
[47] S. Rance, It service management, 2011.
[48] G. Lawton, Knowledge management: Ready for prime time?, Computer 34 (2001) 12–14.
[49] C. Hirsch, J. Hosking, J. Grundy, T. Chaffe, D. MacDonald, Y. Halytskyy, The visual wiki: A new metaphor for knowledge access and management, in: 2009 42nd Hawaii International Conference

on System Sciences, IEEE, 2009, pp. 1–10.
[50] H. Wahl, A. Mense, Visual semantic client a visualization tool for semantic content, in: 2010 2nd International Conference on Education Technology and Computer, volume 2, IEEE, 2010, pp. V2–19.
[51] The Alan Turing Institute, Knowledge graph, 2024.
[52] M. Jonathan, T. Diego, Análisis de patrones en la evolución de wikis, in: Humanidades Digitales: Construcciones locales en contextos globales, Asociación Argentina de Humanidades Digitales, 2016.
[53] I. Gorton, G. D. Black, K. L. Schuchardt, C. Sivaramakrishnan, S. K. Wurstner, P. Hui, Gs3: A knowledge management architecture for collaborative geologic sequestration modeling, in: 2010 43rd Hawaii International Conference on System Sciences, IEEE, 2010, pp. 1–10.
[54] V. Gavekar, M. A. Kumbhar, A. D. Kumbhar, Semantic web for e-governance using wiki technology, in: 2010 3rd International Conference on Emerging Trends in Engineering and Technology, IEEE, 2010, pp. 543–548.
[55] V. Loia, G. Fenza, C. De Maio, S. Salerno, Hybrid methodologies to foster ontology-based knowledge management platform, in: 2013 IEEE Symposium on Intelligent Agents (IA), IEEE, 2013, pp. 36–43.
[56] J. Baumeister, J. Reutelshoefer, F. Puppe, Knowwe: a semantic wiki for knowledge engineering, Applied Intelligence 35 (2011) 323–344.
[57] J. Davies, M. Grobelnik, D. Mladenić, Semantic knowledge management: Integrating ontology management, knowledge discovery, and human language technologies, volume 352, Springer, 2009.
[58] K. Saxena, D. Rajpoot, A way to understand various patterns of data mining techniques for selected domains, arXiv preprint arXiv:0911.0781 (2009).
[59] H. Astudillo, V. Codocedo, G. Canals, D. Torres, A. Diaz, A. Napoli, A. Gomes, M. G. Pimentel, Combining knowledge discovery, ontologies, annotations, and semantic wikis, 2009.
[60] D. Torres, Wiki semántica usando Folksonomies, Ph.D. thesis, Universidad Nacional de La Plata, 2009.
[61] G. A. Sevilla, S. G. Zapata, E. Torres, Uso de wiki en elicitación distribuida de requisitos de software: un experimento controlado, in: XX Congreso Argentino de Ciencias de la Computación (Buenos Aires, 2014), 2014.
[62] T. D. Wang, B. Parsia, J. Hendler, A survey of the web ontology landscape, in: International Semantic Web Conference, Springer, 2006, pp. 682–694.
[63] A. C. Khadir, H. Aliane, A. Guessoum, Ontology learning: Grand tour and challenges, Computer Science Review 39 (2021) 100339.
[64] T. Gao, Y. Chai, Y. Liu, A review of knowledge management about theoretical conception and designing approaches, International Journal of Crowd Science 2 (2018) 42–51.
[65] M. R. Rodríguez, I. Casanovas, M. F. P. Cattaneo, Ontología para transferir conocimiento en la etapa de pruebas de software, Revista Gestión de las Personas y Tecnología 13 (2020) 24–24.
[66] J. Corbin, A. Strauss, Basics of qualitative research: Techniques and procedures for developing grounded theory, Sage publications, 2014.