Knowledge Graph Management Platforms: A Feature Based Review

Prashant Kumar Sinha[1], Sourav Debnath[2], Kanu Chakraborty[3], Sagar Bhimrao Gajbe[1],
Shiva Shankar Mahato[4], Subhranshubhusan sahoo[1]
[1]Indian Statistical Institute, DRTC, Bangalore-560059,
Karnataka; University of Calcutta, Kolkata- 700073, West Bengal, India
[2]National Institute of Technology, Tiruchirappalli-620015, Tamil Nadu, India
[3]Indian Institute of Technology (BHU), Varanasi- 221005, Uttar Pradesh, India
[4]Indian Institute of Science Education and Research, Pune-411 008, Maharashtra, India
kchakraborty.lib@iitbhu.ac.in

ABSTRACT: **Purpose:** The purpose of this study is to identify and review the existing knowledge graph (KG) management platforms (KGMP) based on a comprehensive set of features to help KG managers, KG researchers, data scientists, scientific and commercial community of different domain for gauging the platforms for satisfactory coverage and usage.

Design/methodology/Approach –The study uses a systematic literature review approach to identify 24 KGMPs and then reviews the platforms, with the help of a set of 32 features prepared from existing literature and by studying the platforms. The set of features were segregated as basic features and core features to make the study granular.

Findings- The review revealed that most platforms were developed by corporate companies and very few were available as open-source platforms. The tools have emerged in the last five years, are domain independent, have both on-premise and cloud deployment facilities and are compatible with a variety of operating systems. They have extensive documentations; use cases in various domains with demo versions. The platforms allow creating KG from structured, unstructured and semi-structured data. NoSQL databases, especially the standard community accepted graph databases like Neo4J, GraphDB, AllegroGraph, are the preferred choice for data storage. The platforms use standard semantic technologies; for instance, for data representation uses RDF data model; for providing access to data SPARQL endpoint, GraphQL endpoint and for querying data languages like SPARQL, GraphQL. Platforms possess the facilities like metadata management, ontology management, linked data management that are integral for KG management. The platforms also provide KG search and visualization with data analytics facilities by third party integration softwares like KNIME, Tableau etc.

Originality/value: This paper carries out a basic level review of KGMPs employing a feature-based approach. Compared to the previous studies, present study not only covers more KGMPs but also provides more features that makes the study granular.

Keywords: Knowledge Graph Management Platform, Knowledge Graph Management, Knowledge Graph, Review, Features

Received: 15 August 2021, Revised 19 September 2021, Accepted 28 September 2021

DOI: 10.6025/stm/2021/3/115-137

1. Introduction

As the World Wide Web (WWW) gained popularity as an infrastructure for exchanging information, it led to the growth of knowledge and its utilization also became quite diverse. But the bottleneck of this infrastructure was the way it was formed. It was developed by connecting a series of Hyper Text Markup Language (HTML) pages from which contextual meaning could not be comprehended i.e., the representation of information could not be leveraged by computers (Glimm & Stuckenschmidt, 2016). That's where the idea of semantics came into the picture because it concentrates on the essence of the words, which exhibits a crucial role in the communication of information. So supposedly, if computers could comprehend the data or information available on a website, then many applications could be built around them. Even the transfer of data between computers, agents for automatic data processing could also occur swiftly (Glimm & Stuckenschmidt, 2016). Thus, Berners-Lee et al. (2001) defined Semantic Web (SW) as “a web of data that is well defined so that meanings could be exchanged which will allow the people and computers to work in parallel in cooperation”. This new branch of research led to advent of many new semantic technologies like ontology (“a formal and explicit specification of a shared conceptualization” (Studer et al., 1998)), linked data (LD) (“a structured data published on the web following a set of principles designed to promote the inter linking between the things (aka resources) and consequently between the various data sets on the web” (W3C, 2015)). These new technologies have revolutionized the areas of knowledge organization (Hjørland, 2008), knowledge representation (KR) (Davis et al., 1993) which has in turn helped in automating the information retrieval process. Thus, the semantic community has shifted its focus towards new technology like “Knowledge Graph” (KG).

The term is composed of two terms “Knowledge” (It can be termed as a derivative of information which represents the information in more precise and usable manner (Rowley, 2007)) and “Graph” (“It means a structure which tries to model a pairwise relations between objects or entities (West, 1996)). There have been many definitions of KG but two of the earliest definitions of KG are “a mathematical structure with vertices as knowledge units connected by edges that represent the prerequisite relation” (Marchi & Miguel, 1974) and “a way of structuring and representing text encoding scientific knowledge” (Bakker, 1987). There have been 24 more definitions of KG from 1988 to 2019 summarized in “A Common-Sense View of Knowledge Graphs” by Bergman (2019) and Ehrlinger & Wöß (2016). These definitions were given by range of authorities like companies (Google (Singhal, 2012); Semantic Web Company (Blumauer, 2014)), researchers ((De Vries, 1989); (Paulheim, 2017); (Ehrlinger & Wöß, 2016)), journal (Journal of Web Semantics) and university (Columbia University, 2019). These many definitions of KG still indicate the point Natasha Noy made in 2016 while reviewing Paulheim’s(2017) work that “more precise definition of KG was hard to find at that point in time”. Sheth et al., (2019) describes KG with a very basic definition as “a structured knowledge in a graphical representation”. The various applications of information processing and management in which KG plays an important role are like enriched semantic applications that improves browsing, personalization, recommendation process, improved data integration mechanisms for various data types and sources, updating the Machine Learning (ML) and natural language mechanisms to enhance the functionalities of chatbot and voice assistants (Sheth et al., 2019).

As KG and ontologies have many similarities; both of them consist of things/nodes/concepts and relationships between them, both can have the same type of visual representation, both of them may use the same data model like Resource Description Framework (RDF) etc. So, people may tend to confuse ontologies with KG (Hedden, 2016). But KG is much more than just ontologies. KG consists of more than one domain ontologies, or may comprise ontology and another vocabulary/knowledge organization system like thesauri. Even an upper ontology or foundation ontology may also act as a data model for a KG (Ehrlinger & Wöß, 2016). KG can be categorized as Open KG (http://datacommons.org/), KBpedia (http://www.kbpedia.org/), Proprietary KG (Knowledge Vault (Dong et al., 2014), Yahoo!’s Knowledge Graph (Torzec, 2014) (Fensel et al., 2020)). It can also be categorized as generic KG and domain specific KG. KG can be constructed using various pathways like manual curation (Cyc (Lenat, 1995); Wikidata (Vrandeèiæ & Krötzsch, 2014)), creation from (semi) structured sources (DBpedia (Lehmann et al., 2013); YAGO (Suchanek et al., 2007)), creation from unstructured sources (NELL (Carlson et al., 2010); WebIsA (Seitner et al., 2016) and (Heist et al., 2020)). Paulheim (2017) describes *KG construction* as a group of operations on one or more than one sources to create a KG, whereas *KG refinement* is considered to be improvement of an existing KG, by increasing the derived knowledge or locating and removing the wrong knowledge pieces etc. Fensel et al., (2020) proposed a process model to build a KG consisting of following steps namely knowledge creation (deals with the process of acquisition, development, structuring and evaluating the KG creation), knowledge hosting (deals proper KR with semantically enriched data), knowledge curation (deals with assessment, cleaning and enrichment of KG), and knowledge deployment (deals with publishing the KG). Thus, a KG life cycle includes processes like data extraction, data storage, knowledge creation, knowledge modification, knowledge updation, knowledge evaluation, querying, visualization, integration, application development etc. (Ruqian et al., 2020); (Fensel et al., 2020). The above discussion clarifies that overall knowledge graph management (KGM) is a tedious, time consuming, complex process and requires expertise. Thus, there is a requirement of user friendly, easily understandable and accessible, knowledge graph management platforms (KGMPs) which encapsulates most of the KG life cycle processes to offer the user a great experience. In this regard, the scientific community has been quite aware and sincere efforts have been made to develop KGMPs. But, the platforms required for performing KGM activities and building specific applications on them are still quite limited (Hasse et al., 2019).

The objective of this work is to identify the existing KGMPs and review them based on 32 features which were divided into groups as basic features (e.g. documentation, software type, domain coverage, creator etc.) and core features (e.g. semantic framework, ontology management, data access infrastructure, search platform etc.). Hence, this paper will act as a single point source for the KG developers, user and research community to know what features these KGMPs possess so that users can use this work as a stepping stone to decide upon which KGMP to choose for their work. The work will also help the developers of these KGMPs to compare their platforms with other existing platforms and modify or enhance their platforms. The major contributions of this work are:

–Identifies a comprehensive list of existing KGMPs.

–Provides a set of features that facilitate to elucidate the KGMPs concisely.

–Provides a review of the existing KGMPs based on the features.

–Description and summarization of the available KGMPs at one place.

Here the work has been organized in the following manner: Section 2 “Knowledge graph management” elucidates the basics of KGM. Section 3 “State of the art” discusses the similar works where features were used to compare KGs, or to study a platform related to semantic technologies like LD visualization tools and Knowledge Management (KM). Section 4 “Methodology” provides a step by step workflow used to perform the study. Section 5 “Review of KGMPs” discusses the features and reviews of listed KGMPs based on selected features. Section 6 is “Findings”; summarizes the findings about the KGMPs. Section 7 is “Implications”, provides significant directions of usage of this work. Section 8 “Conclusion”, concludes the work.

2. Knowledge Graph Management (KGM)

A knowledge graph management (KGM) is an umbrella term that can be described for the efficient management of KG that constitutes the process of knowledge acquisition, knowledge fusion, knowledge structuring, knowledge storage, KG development, KG curation, KG publication, KG usage and KG evaluation. Bellomarini et al., (2018) preaches that many corporate companies build KG and have a lot of data to play with analytics. Following equation was given by them to summarize knowledge graph management systems (KGMS).

KGMS = KBMS + Analytics + Big data (Bellomarini et al., 2018)

Knowledge graph management systems (KGMS), Knowledge base management system (KBMS)

Bellomarini et al., (2018) also put forth the main requirements of a full-fledged KGMS: a) formalized language system for reasoning (with features like simple syntax, high expressive power, numeric computation and aggregations, probabilistic

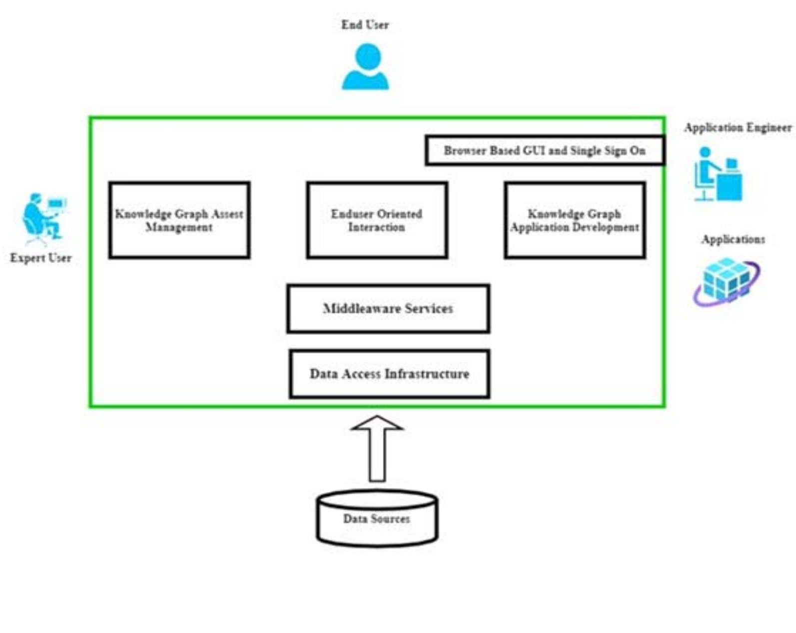

Figure 1. A basic architecture of KGMP (Hasse et al., 2019)

reasoning, ontological reasoning, low complexity, rule repository, rule management, ontology editor, and dynamic orchestration); b) Accessing and Handling Big Data (with features like data, database, data warehouse access, ontology based data access, multi-query support, data cleaning, exchange and integration, Web Data Extraction, interaction, and internet of things); c) Embedding Procedural and Third-Party Code (Procedural Code, Third-Party Packages for ML, Text Mining, Natural Language Processing (NLP), Data Analytics (DA), and Data Visualization) to build KG and manage them. A software suite or platform which supports these KGM activities or processes is termed as knowledge graph management platforms (KGMPs). KG is not only acting as a repository of corporate organization's master data i.e. schema ontology and static knowledge reference point but also a place to link various legacy data sources like relational databases (Hasse et al., 2019). Different KGMPs are composed of different components based on the functionality they need to support, but a few basic components remain the same. A basic architecture of a KGMP platform is given in Figure 1 which was built to facilitate various functionalities for separate groups of users. The KGMP works on a graph database which stores the KG. The interaction is done by the data access infrastructure of the KGMP using standard SPARQL 1.1 queries, which makes the system independent from a specific database provider. On top of the data access infrastructure, the platform service layer is implemented at the KGMP backend which supports a variety of functionalities for communication with KGs (Hasse et al., 2019).

3. State of the art work

The research community has reviewed and compared various existing platforms/software like ontology developing tools, LD visualization tools, KM tools, etc. based on features. So that an overall holistic study of these platforms in a fine-grained way could be done to facilitate the user in choosing an appropriate platform for their own work. Kapoor & Sharma (2010) reviewed the freely available ontology building tools namely Protégé 3.4, Apollo, IsaViz and SWOOP based on various features like tools' architecture, interoperability, inference services, usability, versioning and collaborative work support. The study observed that a lot of the users are hesitant in migration from one tool to another even the functionalities are more, because of the inability of inter-changing their ontologies. Similarly, Padmavathi & Krishnamurthy (2017) reviewed ontology developing tools like Ontolingua, WebOnto, WebODE, Protege-2000, OntoEdit and OilEd based on various parameters listed as developers, availability, SW architecture, ontology storage, import from languages, export from languages, ontology libraries etc. Desimoni & Po (2020) prepared a list of 77 LD visualization tools and investigated which of them were working and as a result reduced the count to 29. These tools were then separated into five categories: facet browsers, LD browsers, multi visualization tools, graph-based visualization tools and ontology visualization systems. A detailed description of all 29 working tools were provided and compared based on various features like statistical information, visualization types, entry point, query capabilities, type of visualizations, and export types. 14 tools were also tested on a big LD dataset. This revealed the efficacy of these tools as well. The study is one of the most comprehensive works in the LD visualization tool study and can easily be used by the LD researchers for choosing the best tool for their purpose. Tang et al., (2010) compared five architectural KM tools based on an evaluation framework defined by a set of 10 criteria in order to draw a detailed perspective of the concentration area of architectural KM support, their strength, weaknesses, and agreement to the present architectural description standard. Similarly, Ngai & Chan (2005) used analytical hierarchy processing methods to show how to analyse and compare KM tools in the software market to select the most appropriate tools. The method used here for comparing these tools is a well-accepted, quantitative tool used for multi-criteria decision making and has been used by many investigators of the field.

Researchers have attempted to study the available KG, based on various features in order to present a holistic view of them. Zhao et al., (2018) studied some of the popular KGs like DBpedia, Google Vault, YAGO, and NELL to extract a core architectural system with basic components for building KG. The work covered; various approaches of KG development like top down and bottom up approaches; storage schemas of KG: RDF-based and graph database; and also four aspects of knowledge extraction with their concepts, related approaches, and development tools. The work can act as a starting point for the researchers working in the area of KG. In a similar study Heist et al., (2020) presented a work where publicly available KG like Voldemort, CaLiGraph, Wikidata etc. were compared against four basic features namely contents, size, coverage, and overlap to provide detailed insights. The work also discusses the varied approaches of KG creation: manual curation, creation from (semi) structured sources, and creation from unstructured sources. The work basically provides an inner depth about the publicly available KGs and analyzes the integration of such KGs . Ji et al., (2020) presented a comprehensive survey on KG based on four scopes: KG embedding with a full-scale systematic review from embedding space, scoring metrics, encoding models, embedding with external information, and training strategies; knowledge acquisition of entity discovery, relation extraction, and graph completion; perspectives of embedded learning, relational path inference and logical rule reasoning; temporal KG representation learning and completion; real-world knowledge-aware applications on natural language understanding, recommendation systems, question answering and other miscellaneous applications. The work also provides a simple classification of research on KG which can come in handy for researchers to determine their area of interest. Chen et al., (2020) provided a review on the various methods and techniques that are being used for KG completion, a topic that is related to KG construction by predicting the missing entities or relationships in KG and mining unknown facts.

Bilal (2021) presented an extensive survey on various domain-based KGs like healthcare, education, information and communication technologies, science and engineering, finance, society and politics, and travel based on parameters like KG usage, KG construction algorithm(s), resources used to feed the KG, presence of KG embedding techniques, evaluation approach used. This work revealed various limitations like the quality of data being used for KG development, lack of discussion about KG construction algorithms, inability to attain interoperability of information because of insufficient incorporation of semantic expansion/broadening techniques, rare inclusion to KG embedding techniques in KG construction approaches, lack of KG evaluations through proper metrics, lack of KG construction methodologies for a domain-specific KG to automate the process etc.

In case of KGMPs, there are significantly very few attempts being made to review them because research on KG technologies is still at the budding stages. But KG in enterprises is being considered as the backbone of the data integration process where big data processing could be achieved through semantic technologies (Galkin, et al., 2016). Thus, Galkin et al., (2017) gave an assessment framework for enterprise KG and then compared 10 popular enterprises KG. Some of the dimensions of this consist of KG curation, exploration, search, and security. That work was restricted in comparing enterprise KG systems; thus, the scope was limited but this can be considered as the first state-of-the-art studies in terms of KG systems. McCusker et al., (2018) elucidated a framework that could be reused for not only creating a KG, but also helps in knowledge curation, interaction, and inference. Bellomarini et al., (2019) proposed KG as the reference technology for the enterprise artificial intelligence (AI) context, and a small comparison out of this work of a few of the existing KGMPs which had diverse capabilities was presented by Georg (Gottlob, 2019). Though the work compared a number of KGMPs, the features used were very less that do not provide comprehensive information about the platforms. But, in the present work, authors attempt to review a comprehensive number of existing KGMPs based on an extensive group of features to act as a stepping stone to help the research community select the best platform for their use.

4. Methodology

Before reviewing the KGMPs, it was important for authors to use scientific methods for identifying them. In order to identify the KGMPs, authors followed the methodology mentioned by Desimoni & Po (2020); Sinha & Dutta (2020). A detailed survey of various scientific papers, search engines, was performed with the objective of identifying a variety of KGMPs that could be utilized by the users of KGs. It is to be noted that KG technologies is a blooming research area so there may not be many KGMPs platforms available but here authors tried to prepare as exhaustively as possible for the benefits of the community. A list of 24 KGMPs was identified and shortlisted. A step by step methodology (Figure 2) for the identification of tools and its review has been mentioned below.

Step 1: Query formulation
The process of identifying KGMPs tools started with the formulation of queries. The terms such as “Knowledge graph,” “Management,” “Platform,” “Software,” “Tools,” “Knowledge graph management,” “Knowledge graph management platform,” were applied on various databases and search engines to retrieve the publications on KGMPs or to get a tool directly.

Step 2: Selection of sources
The domain of KG research is fairly new and immature where a lot of efforts are being made. The type of platform choice for this study was also very specific so the sources used for the identification of tools were not restricted to specific databases but search engines were also used to obtain the platforms. The selected databases are Scopus (https://www.scopus.com/home.uri), Science Direct (https://www.sciencedirect.com/), IEEE (https://ieeexplore.ieee.org), Web of Science and GoogleScholar (https://scholar.google.co.in/). The selected search engines like Google, Yahoo, and Microsoft Bing were searched through.

Step 3: Search
The queries that were formulated in the first step were put on the database and search engines in different combinations like, “Knowledge graph” AND “Management” OR “Knowledge graph management platform” OR “Knowledge graph management” AND “Software” to obtain the scientific papers discussing the tools or to locate the tools available on the web. To remove the redundant results that appeared while querying the databases were removed using the Excel workbook.

Step 4: Applied Inclusion and Exclusion Criteria

This step actually was one of the most important parts of the methodology that helped to prepare a list of KGMPs. A number of inclusion and exclusion criteria were formulated as provided below. These criterions were used to select the papers/ tools for complete reading and analysis.

Inclusion criteria (IC)
–Papers published in journals, conferences and as book chapters which dealt with KGMPs were selected.

–Papers dealing with KGMPs which were domain specific were also included.

–KGMPs on which scientific papers were not written but were available through websites for study like Linked Data Hub (https://atomgraph.com/) and had enough extensive documentation that allowed them to be selected for the study.

-The study is restricted till the year 2020. Platforms published or coming into existence after 2020 were not included in the study.

Exclusion criteria (EC)
–Papers not published in English.

–Platforms like HD-KGMP: A Health Domain Knowledge Graph Management Platform, Oracle Apex-Knowledge Graph Management Platform with oracle apex, Interest Taxonomy at Pinterest, Open data platform were not considered for the study because they had very little information about the platform itself and for some of them even websites were not working as well.

Step 5: Selecting and downloading of papers and documentations
Keeping the above inclusion and exclusion criteria in consideration the databases were searched through to retrieve and identify the publications. The abstract and title of the publications were also read carefully to realise the relevant papers that spoke about KGMPs and accordingly, their full text was obtained. The KGMPs that were available through their websites were visited and their documentations were downloaded as well. Thus, 24 KGMPs were shortlisted.

Step 6: Complete reading of the papers and documentations
The downloaded papers and documentations were read completely. The reading process of these scientific communications and documentations allowed us to conduct the review.

Step 7: Review of KGMPs
Since the primary objective of this work was to review the KGMPs based on various features, the features were extracted by analysing existing literature and studying the shortlisted KGMPs. The features were segregated in two groups and necessary data were accumulated. This whole step has been detailed in section 5. Authors enlisted the KGMPs selected for the study in Table I.

SNo.	KGMPs	Brief Summary
1	Metaphactory (Haase et al., 2019)	A creative FAIR data KGMP that facilitates to convert data into real knowledge to accelerate technological change by developing adaptable applications for organization, institutions etc. by researchers and developers.
2	Unified Workbench for Knowledge Graph Management (UWKGM) (Ichise et al., 2018)	A platform for developing, managing and using KG, to create and expand systems. It also supports efficient language refining competencies for artificial intelligent systems.
3	Heaven Ape (HAPE) (Ruqian et al., 2020)	A big KGMP prototype facilitating the development, management, and application of comprehensive scale KG.
4	News Hunter Architecture (NHA) (Berven et al., 2020)	A prototype that aggregates data sources, utilizes KG, NLP and ML to develop into an adaptable framework for news journalism to assist journalists.
5	TypeDB (Earlier: Grakn: The Knowledge Graph) (https://grakn.ai/)	Created to scale with organizational data accompanied with a work base which is a visual platform that allows overseeing the whole process of development to management of KG.
6	VADALOG (Bellomarini et al., 2020)	This KMGP takes advantage of the Warded Datalog language as the primary language for KR and reasoning.

7	Wikibase (https://wikiba.se/)	It is a free and flexible knowledge base that takes advantage of Wikidata and acts as an ideal KGMP to develop the data library that interworks with the SW via open standards.
8	MONOLITH (Lepore et al., 2019)	A KGMP,amalgamation of an ontology-based data management (OBDM) platform and an enterprise KGIDE.
9	Spindra (Sun et al., 2019)	A KGMP that provides a geographic KG storage, indexing module that enhances the primary usage of a graph database system (e.g., Neo4j) to appropriately archive location facts and relationships as vertices and edges to allow effective handling of geographic KG.
10	Linked Data Hub (LDH) (https://atomgraph.com/)	KGMP that helps to manage various domain-specific RDF datasets. It facilitates faceted search to easily locate and scrutinize the information according to requirements.
11	GNOSS (https://www.gnoss.com/en/knowledge-graph)	KGMP to integrate, query, visualize and present the enterprise data and processes in a more intuitive way.
12	Blue Brain Nexus (BBN) (https://bluebrainnexus.io/)	A framework that facilitates to model and better utilize data by using KG. It was developed out of the requirement of data management in the simulation neuroscience area, but now has been created to suit alternative use cases.
13	Graph BRAIN (Ferilli & Redavid, 2020)	A generic system that facilitates the development and collaborative population of KG and develops innovative applications.
14	Ontotext (https://www.ontotext.com/products/ontotext-plat form/)	A KGMP to create, manage KG from a variety of datasets and create innovative products.
15	MAANA (https://www.maana.io/)	A KGMP to structure organization data and human knowledge into digital knowledge to help making swift decisions across the full value chain of an Oil and Gas Companies.
16	Anzo (https://docs.cambridgesemantics.com/anzo/v5.1/userdoc/features.htm)	An open overlay KGMP that facilitates users to amalgamate KG against the concealed data silos without unsettling existing processes.
17	OntoWiki (http://ontowiki.net/)	A free and open-source SW solution developed to act as a knowledge acquisition system.
18	Stardog Studio (https://www.stardog.com/platform/)	A tool for data modellers, creators, and managers to develop and efficient management of data fabric.
19	Franz's Entity-Event Knowledge Graph(FEEKG) (https://allegrograph.com/)	KGMP that takes advantage of novel Entity-Event Model and natively amalgamates domain ontologies and metadata. It supports the development of innovative data management solutions.
20	Top Braid Enterprise Data Governance™ (TBEDG™) (https://www.topquadrant.com/products/topbraid-enterprise-data-governance/)	A smart and innovative solution to meaningfully traverse through enterprise metadata silos by cataloguing and linking any type of data; securing and conserving meaning; helping the commercial importance of data.

21	Siren Platform (https://siren.io/platform-overview/)	An adaptable, inquisitive AI platform that takes advantage of semantic data modelling techniques to enforces insights from the data.
22	Poolparty Semantic Suite Graph Editor (PPSSGE) (https://www.poolparty.biz/resources/automating-knowledge-graph-managemen)	An all-round platform that allows enterprises to perform tasks like KM, DA and content organisation. The frame-work is based on W3C recommended standards.
23	Semaphore (https://www.smartlogic.com/semaphore)	A web-based, data model management solution that helps users to develop and manage KG to gain hidden insights about the data.
24	Graphite Knowledge Organization System (GKOS) (https://www.synaptica.com/synaptica-graphite/)	A potent platform for swiftly developing and handling KOS with help of a GUI. It is based on LD and SW standards and utilizes RDF concept modelling.

Table 1. Summary of shortlisted KGMPs

Step 8: Findings about KGMPs

The data accumulation allowed us to derive a few facts about the studied platforms. These facts about the KGMPs can act as a guide for the users to understand and have a bird's eye view of the KGMPs while selecting the platforms for their projects. This finding about the platform has been detailed in section 6.

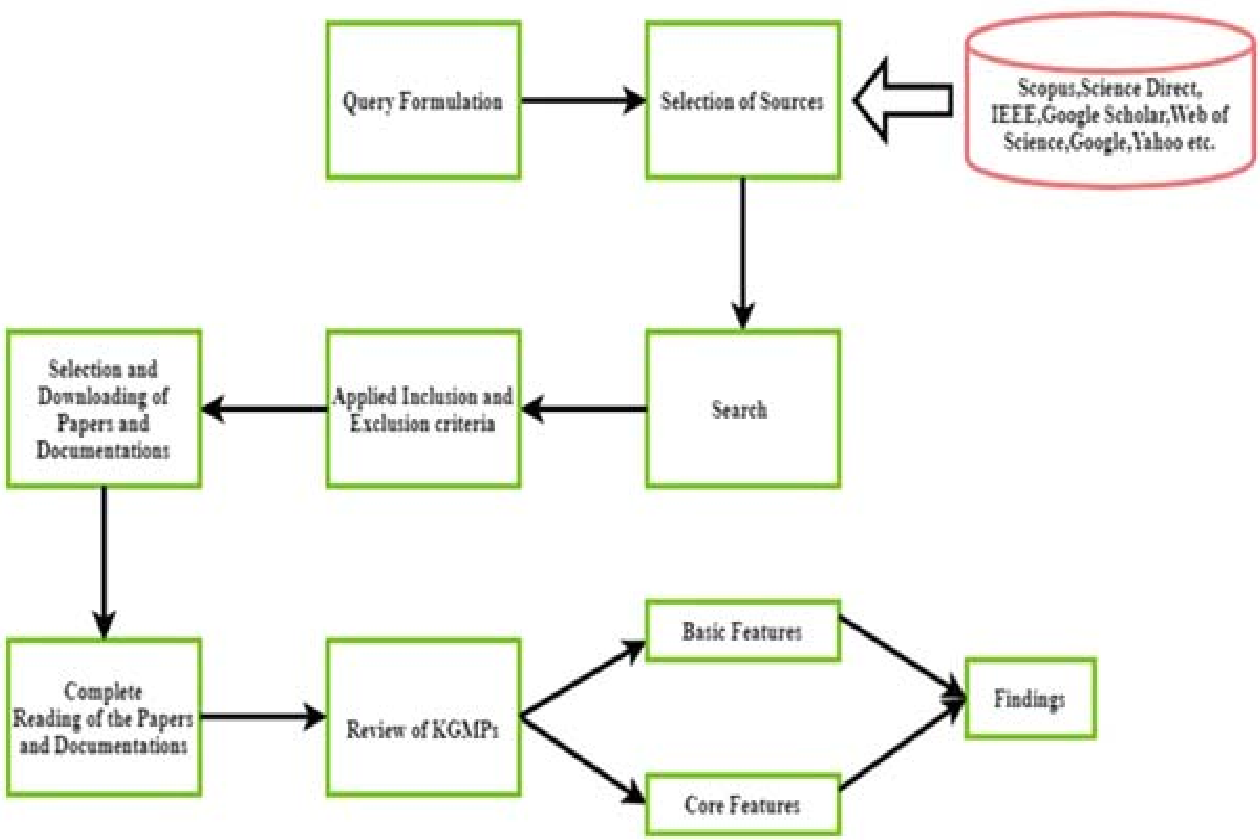

Figure 2. Methodology to review KGMPs

5. Review of Knowledge Graph Management Platforms (KGMPs)

In order to review the KGMPs, a feature-based approach was employed. Few features were extracted by analysing existing literatures like (Gajbe et al., 2021); (Galkin et al., 2017); (Desimoni & Po, 2020); (Kapoor & Sharma, 2010); (Padmavathi & Krishnamurthy, 2017); whereas a few criteria were also identified by studying the KGMPs. From a general point of view, these are specialized platforms producing specific products called KG. A list of 32 features was identified for reviewing the KGMPs so that a user may be able to decide the right for their project. These features were segregated in two sets called the basic features and core features. Definitions of the set of features and each of the criteria used to review the KGMPs are given below. The data collected against the list of features has also been provided in Tables II and III.

5.1 Basic Features

Features that exhibit introductory information about the KGMPs which a user generally desires to know about a KGMP. These features are not directly related to development or management of the product that a platform offers to produce (i.e., KG here) but it highlights the overall composition, feasibility, availability, usability, usage, advancement of the KGMPs in general that acts as a compass for its selection for a particular project. Below authors provide definitions of 15 basic features and in Table II data about the platforms against these basic features have also been provided.

• **Creator:** Refers to the developer of the platform.

• **First Release (FR)**: Refers to the year of release of the platform for the first time. For a few platforms like HAPE, NHA, MONOLITH, GraphBRAIN specifically their first release year was not available because these are prototype platforms. But scientific papers related to them were published. Hence for these few platforms their paper publication year was taken as their first-year release values.

• **Latest Release (LR)**: Refers to the year of release of the platform's latest version that is restricted to 31th May 2021.

• **Latest Version (LV)**: Refers to the most recent version of the platform that is restricted to 31th May 2021.

• **Software Type (ST):** It refers to the nature of the software based on its availability for use to the user. This makes the software either open source software (OSS), free software (FS) or subscription-based software (SBS).

• **License**: A license or licence is an official permission or permit to do, use, or own something. It demonstrates the agreement under which the platforms can be registered.

• **Platform Deployment (PD):** Refers to the process of deploying a platform on a server or device. The PD can be performed using different methods for instance on premise (a kind of PD method where the software is deployed and operated on a user's in-house server and computing framework); on cloud (a kind of PD method where the software is deployed and operated outside the user's in-house server and computing framework where the installing model are of various types like software as a service (SaaS), platform as a service (PaaS) and/or infrastructure as a service (IaaS)) (Grabis, 2020).

• **Operating System Compatibility (OSC):** It refers to the flexibility of a platform in accordance with its ability to be installed on an operating system for use.

• **Container Orchestration Systems (COS):** Refers to condensation software codes and its dependencies so that a platform can run appropriately and consistently on any digital framework. It leverages the developers to develop and deploy solutions swiftly and safely, for instance docker, kubernetes (K8s) (*Containerization*, 2019).

• **Language:** It refers to all the different kinds of software languages that have been used in developing the various components of the platform. This includes the markup languages (like HTML), scripting languages (like Python, PHP etc.) or Compiler languages (like Java) etc.

• **Single Sign On (SSO):** Refers to the feature that facilitates the user to authenticate various applications by signing in using a single identification.

• **User Interface (UI):** It refers to the interface through which the user interacts with the platform. It can be a browser based graphic user interface (BBGUI) or native GUI (NGUI) (Local System).

• **Domain Coverage (DC)**: Refers to the scope of the platform in terms of domain coverage based on various datasets that can be managed through the use of the platform.

• **Tool URI (Uniform resource Identifier) /Live Demo:** Refers to the location of the platform on the web where the tool could be obtained or explored by the user.

• **Documentation:** Refers to the document or the scientific article or white papers that explain the platforms by providing detailed information about its overall architecture, functionalities and usage.

KGMPs	Basic Features														
	Creator	**Software Type**	**License**	**First Release**	**Latest Release**	**Latest Version**	**Platform Deployment**	**Operating System Compatibility**	**Container Orchestration Systems**	**Language**	**Single Sign On**	**User Interface**	**Domain Coverage**	**URI to the tool / Live Demo**	**Documentation**
Metaphactory	Metaphacts	FS /SBS (14 days trial and one-year trial for academia)	Commercial	2015	2020	v3.6	on-premise and cloud based	Linux Windows MAC	Docker	HTML5 SPARQL.js	Yes	BBGUI	Domain Independent	Yes (https://wikidata.metaphacts.com/resource/app:Start)	Yes (https://metaphacts.com/get-started,Scientific paper available)
UWKGM	Ichise Lab	OSS (https://github.com/ichise-lab/uwkgm)	BSD 3-Clause License	2020	2021	v 0.1.0.18	on-premise and cloud based	Ubuntu (Linux)MAC	Docker K8s	Python Java	Yes	NGUI	Domain Independent	Yes (https://uwkgm.aire.aist.go.jp/login)	Yes (https://sites.google.com/view/uwkgmplatform/home, scientific paper available)
HAPE	Chinese Academy of Sciences, National Key Research and Development Program, National Natural Science Foundation (China)	NA	NA	2020 Prototype (not yet released)	NA	NA	NA	Knowledge graph operating system (KGOS)	NA	JavaScript Python	NA	BBGUI	Domain Independent	NA	Scientific paper available
NHA	Wolftech and Intelligent Information System University of Burgen	SBS	Commercial	2020 Prototype (not yet released)	NA	NA	NA	NA	NA	Python, C#, HTML,AngularJS, CSS, Sketch, Marvel	Yes	BBGUI	Domain Independent (But only news oriented)	NA	Scientific paper available
TypeDB	Vaticle (earlier Grakn Labs)	OSS (https://github.com/vaticle)	AGPL v3.0,Commercial	2016	2021	v 2.0.2	on-premise and cloud based	Linux Windows MAC	NA	Java, Python, Node JS, Julia, Ruby, Rust,C++	Yes	BBGUI	Domain Independent	Yes (https://grakn.ai/)	Yes (https://docs.vaticle.com/docs/general/quickstart)
VADALOG	University of Oxford	NA	NA	2019 Prototype (not yet released)	NA	NA	NA	NA	NA	Datalog	NA	BBGUI , NGUI	Domain Independent	NA	Scientific paper available
Wikibase	Wikimedia Deutschland	OSS (https://github.com/wikimedia/Wikibase.git)	GNU General Public License	2012	2021	v 1.37.0	on-premise and cloud based	Cross-platform	Docker	PHP, JavaScript	Yes	BBGUI	Domain Independent	Yes (https://wikiba.se/)	Yes (https://doc.wikimedia.org/Wikibase/master/php/)
MONOLITH	OBDA Systems SRL and Sapienza University of Rome	SBS	Commercial	2019	NA	NA	NA	Linux Windows	NA	ReactJS	Yes	BBGUI	Domain Independent	Yes (http://www.monolith.obdasystems.com/)	Yes, (http://www.monolith.obdasystems.com/monolith-user-manual/, Scientific paper available)

Spindra	Arizona State University	NA	NA	2019 Prototype (not yet released)	NA	NA	NA	NA	NA	NA	NA	BBGUI	Spatial	NA	Scientific paper available
LDH	Martynas Jusevičius and Džiugas Tornau (Atomgraph)	OSS (https://linkeddatahub.com/linkeddatahub/docs/linkeddatahub-cloud/)	Apache License and Commercial	2020	2021	v2.0	on-premise and cloud based	Linux, Windows	Docker	Java	Yes	BBGUI	Domain Independent	Yes (https://linkeddatahub.com/linkeddatahub/demo)	Yes (https://linkeddatahub.com/linkeddatahub/docs/get-started/)
GNOSS	RIAM Intelearning Lab	SBS	Commercial	2010	NA	NA	on-premise and cloud based(SaaS, PaaS)	Linux Windows MAC	NA	NA	Yes	BBGUI	Domain Independent	Yes https://sherlock.gnoss.ai/en/Account/Login?ReturnUrl=/en/Demo	White papers available (https://www.gnoss.com/en/library)
BBN	École polytechnique fédérale de Lausanne, European Union's Horizon 2020 Framework Programme for Research and Innovation	OSS https://bluebrainnexus.io/docs/index.html	Apache License 2.0	2019	2020	v1.5	on-premise and cloud based	Linux, Windows MAC	Docker	JavaScript, ReactJS	Yes	BBGUI	Domain Independent	Yes (https://sandbox.bluebrainnexus.io/web/)	Yes (https://bluebrainnexus.io/docs/delta/index.html)
GraphBRAIN	University of Bari, Artificial Intelligence SRL Bari	NA	NA	2020 Prototype (not yet released)	NA	NA	NA	Linux, Windows MAC	NA	NA	Yes	BBGUI	Domain Independent	Yes (http://193.204.187.73:8088/GraphBRAIN/)	Yes (http://193.204.187.73:8088/GraphBRAIN/faces/GraphBRAIN_users_guide_v5.4.3_1.pdf, scientific paper available)
Ontotext	Atanas, Sirma Group.	SBS	Commercial	2000	2021	v3.4	on-premise and cloud based	Linux Windows MAC	Docker K8s	JavaScript, Java	Yes	BBGUI	Domain Independent	Yes https://www.ontotext.com/knowledge-hub/demoservices/	Yes https://platform.ontotext.com/3.4/release-notes.html
MAANA	MAANA Inc.	SBS	Commercial	2012	NA	NA	cloud based	NA	Docker	HTML5 jQuery	NA	BBGUI	Oil and Gas Company	Yes (https://www.maana.io/)	White papers Available (https://www.maana.io/resources/)
Anzo	Cambridge Semantics Inc.	SBS	Commercial	2007	2021	v5.2	on-premise and cloud based	Linux Windows MAC	K8s	HTML, CSS, JavaScript.	Yes	BBGUI	Domain Independent	Yes (https://www.cambridgesemantics.com/request-a-demo/)	Yes (https://docs.cambridgesemantics.com/anzo/relnotes/anzo-releases.htm)

OntoWiki	Agile Knowledge Engineering and Semantic Web (AKSW) research group at the University of Leipzig	OSS (https://github.com/AKSW/OntoWiki)	GNU General Public License Version 2 (GPLv2)	June 3,2016	October 4, 2016	v 1.0.0	on-premise and cloud based	Linux Windows MAC	NA	PHP, CSS, JavaScript	Yes	BBGUI	Domain Independent	Yes (http://master.demo.ontowiki.eu/)	Yes (http://docs.ontowiki.net/Install-Ontowiki.html)
Stardog Studio	STARDOG	SBS	Commercial	2011	2021	v7.6.3	on-premise and cloud based	Linux Windows MAC	Docker	Java	Yes	BBGUI	Domain Independent	Yes (https://stardog.studio/#/)	Yes (https://docs.stardog.com)
FEEKG	Franz Inc.	FS/SBS (https://allegrograph.com/downloads/)	Commercial	2004	2020	v 7.0.0	on-premise and cloud based	Linux Windows MAC	Docker	C#, Clojure, Java, Lisp, Perl, Python, Ruby, Scala,	Yes	BBGUI	Domain Independent	Yes (https://franz.com/agraph/downloads/?ui=new)	Yes (https://franz.com/agraph/downloads/?ui=new)
TBEDG™	TopQuadrant, Inc.	FS/SBS	Free and Commercial	2011	2021	v7.0	on-premise and cloud based	Linux Windows MAC	NA	Java	Yes	BBGUI	Domain Independent	Yes (https://www.topquadrant.com/get-started-data-governance/)	Yes (https://www.topquadrant.com/knowledge-assets/product-documentation/)
Siren Platform	Siren	FS (https://siren.io/downloads/)	Private license can be occupied, Siren License	2018	2021	v11.1.2	on-premise and cloud based	Linux Windows MAC	Docker	Java	Yes	BBGUI	Domain Independent	Yes (https://public.demo.siren.io/kibi/app/kibana)	Yes (https://docs.siren.io/index)

HAPE	RDF	RDF, Concept taxonomy, Snapshot, topic and topic pattern, Possible world	RDF, XML, N-Triples, Turtle, RDFa, JSON-LD,	PostgreSQL 10.1, Neo4J 3.4.4	SPARQL	SPARQL endpoint Virtuoso	RESTful API	NA	Elastic Search 6.6.2.	Yes	Yes	Yes	Yes	Yes	Yes	NA	DBpedia (Just prototype)
NHA	Text	JSON OWL	NA	BrightstarDB	a .NET component for C#-native, SPARQL	SPARQL endpoint Virtuoso	RESTful API	Microsoft Entity Framework along with Language Integrated Query	Elastic Search	Yes	NA	Yes	Yes	Yes	NA	NA	News articles of different domains (Just prototype)
TypeDB	CSV JSON XML OWL	TypeDB schema ER model, HyperGraph	NA	RocksDB	TypeQL (Graql)	NA	RESTful API gRPC	TypeDB +TypeQL	NA	Yes	Yes	Yes	Yes	Yes	Yes	No	Financial services, robotics, life sciences, drug discovery, ML, text mining
VADALOG	CSV, JSON -, XML ,OWL, Text	RDF	NA	RDMS (SQL),Graph databases	SPARQL, Cypher, GraphQL, Datalog	Ontological and recursive reasoning based on Datalog	RESTful API	NA	NA	Yes	NA	Yes	Yes	Yes	Yes	Yes	NA

Abbreviations Used: OSS: Open Source Software, **UWKGM:** Unified Workbench for Knowledge Graph Management, **HAPE:** Heaven Ape, **NHA:** News Hunter

Architecture, **LDH:** Linked Data Hub, **BBN:** Blue Brain Nexus , **FEEKG:** Franz's Entity-Event Knowledge Graph, **TBEDG™ :** Top Braid Enterprise Data Governance™ ,

PPSSGE: Poolparty Semantic Suite Graph Editor, **GKOS:** Graphite Knowledge Organization System, **SBS:** Subscription based software, **FS:** Free Software, **BBGUI:**

Browser Based Graphic User Interface, **NGUI**: Native Graphic User Interface, **HTML:** Hypertext Mark-up Language, **K8s**: Kubernetes , **URI:** Uniform resource Identifier,

SPARQL: Simple Protocol and RDF Query Language or SPARQL Protocol and RDF Query Language, **NA**: Not Applicable,

PHP: Hypertext Preprocessor, **CSS:** Cascading Style Sheets

5.2 Core Features

Features that exhibit granular information about the functionalities of KGMPs. These features are directly related to development, storing, querying, handling, management and application of the product that platform offers to produce (i.e., KG in here). Core features along with the basic features provide an excellent combination for the selection of KGMP for a particular project. Below authors provide definitions of the 17 core features of KGMPs and in Table III the data about the platforms against these core features have also been provided.

- **Data Import Format (DIF):** It refers to the type of data formats that are allowed to be inserted in the platform for KG creation. For instance, text, Comma Separated Value (CSV) etc.
- **Dataset Transformation Formats (DTF):** It refers to the transformed data formats which are used for KG creation. For instance, RDF.
- **Data Export Formats (DEF):** The data formats in which the data are allowed to be downloaded once KG is created. For instance, RDF, JavaScript Object Notation (JSON) etc.
- **Database (DB):** The data storage space that stores the data for the platform. For instance, My Structured Query Language (MySQL), Network Exploration and Optimization 4 Java (Neo4J) etc.
- **Database Query Language (DQL):** It refers to the language that leverages users to interact with the databases so that the data can be utilized by the platform. For instance, Simple Protocol and RDF Query Language or SPARQL Protocol and RDF Query Language (SPARQL), Graph Query Language (GraphQL).
- **Data Access Infrastructure (DAI):** The infrastructure that allows access to the data stored in a database. For instance, SPARQL endpoint, GraphQL endpoint.
- **Application Program Interfaces by Architectural Styles (APIAS):** It refers to the platform architectural style that elucidates a group of protocols that can be leveraged for developing web services. For instance, Representational State Transfer (REST) API, GraphQL API.
- **Semantic Framework (SF):** It refers to the backend framework that allows to store,query, and analyse RDF data. For instance, Resource Description Framework 4 Java (RDF4J), Apache Jena, Mulgara.
- **Search Platform (SP)**: It is termed as the backend software system that is used for providing the search facilities in KG. For instance, Elasticsearch, Apache Solr etc.
- **Metadata Management (MM):** Feature leveraged for creating and customising metadata for KG development.
- **Thesaurus and Taxonomy Management (TTM):** Feature used for creating and customising thesaurus for KG development.
- **Ontology Management (OM):** Facility that allows development and customization of ontology for KG development.
- **Linked Data Management (LDM):** Facility leveraging the development and customization of LD for KG development.
- **Knowledge Graph Search (KGS):** Facilities of search of entities and relationships in the developed KG.
- **Knowledge Graph Visualization (KGV):** Facilities of visualization of KG through a graphic tool.
- **Data Analytics (DA):** It refers to the feature that is provided by the KGMPs to analyse the data and have insights about them.
- **Use Case Domains:** Refers to the successful example of domains where the platforms have been used for KG manage

Platforms	Core Features																
	Data Import Format	**Dataset Transformation Formats**	**Data Export Formats**	**Database**	**Database Query Language**	**Data Access Infrastructure**	**Application Program Interfaces by Architectural Styles**	**Semantic Framework**	**Search Platform**	**Metadata Management**	**Thesaurus and Taxonomy Management**	**Ontology Management**	**Linked Data Management**	**Knowledge Graph Search**	**Knowledge Graph Visualization**	**Data Analytics**	**Use Case Domains**
Metaphactory	RDF, CSV, Text	RDF	RDF	StardogDB, GraphDB, Amazon Neptune, Blazegraph DB, AllegroGraph, MarkLogic, Oracle Spatial and Graph, OpenLink Virtuoso,	SPARQL 1.1 R2RML, GeoSPARQL (Domain specific)	SPARQL endpoint	RESTful API	RDF4J	Graph Scope	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Engineering, manufacturing, cultural heritage,life science and pharma
UWKGM	Text	RDF	RDF	MySQL, MongoDB	SPARQL	SPARQL endpoint Virtuoso	RESTful API	NA	NA	Yes	NA	Yes	Yes	Yes	Yes	NA	Life science, Japanese COVID-19 case study, DBpedia
HAPE	RDF	RDF, Concept taxonomy, Snapshot, topic and topic pattern, Possible world	RDF, XML, N-Triples, Turtle, RDFa, JSON-LD,	PostgreSQL 10.1, Neo4J 3.4.4	SPARQL	SPARQL endpoint Virtuoso	RESTful API	NA	Elastic Search 6.6.2.	Yes	Yes	Yes	Yes	Yes	Yes	NA	DBpedia (Just prototype)
NHA	Text	JSON OWL	NA	BrightstarDB	a .NET component for C#-native, SPARQL	SPARQL endpoint Virtuoso	RESTful API	Microsoft Entity Framework along with Language Integrated Query	Elastic Search	Yes	NA	Yes	Yes	Yes	NA	NA	News articles of different domains (Just prototype)
TypeDB	CSV JSON XML OWL	TypeDB schema ER model, HyperGraph	NA	RocksDB	TypeQL (Graql)	NA	RESTful API gRPC	Type DB +TypeQL	NA	Yes	Yes	Yes	Yes	Yes	Yes	No	Financial services, robotics, life sciences, drug discovery, ML, text mining
VADALOG	CSV, JSON, XML, OWL, Text	RDF	NA	RDMS (SQL),Graph databases	SPARQL, Cypher, GraphQL, Datalog	Ontological and recursive reasoning based on Datalog	RESTful API	NA	NA	Yes	NA	Yes	Yes	Yes	Yes	Yes	NA

Wikibase	RDF	RDF	JSON, RDF, XML, N3, YAML	MariaDB	Wikidata Query Service(an implementation of SPARQL server based on Blazegraph engine)	SPARQL endpoint	RESTful API	Media Wiki	Elastic Search 6.5.4	Yes	NA	Yes	Yes	Yes	Yes	NA	Life sciences, Library science, Museology
MONOLITH	RDF	Tabular or RDF	Tabular or RDF	RDMS (through API making the query by SPARQL)	SPARQL	SPARQL endpoint	RESTful API	Apache Jena	NA	Yes	NA	Yes	Yes	Yes	Yes	Yes	No specific use but books ontology and superheroes ontology available for trial
Spindra	Tabular or RDF	RDF	RDF	Neo4J and Titan	Cypher or Gremlin	NA	RESTful API	Spindra	NA	NA	NA	NA	NA	Yes	Yes	NA	Finding museums at a single geolocation (Just prototype)
LDH	CSV RDF	RDF	JSON-LD	NA	SPARQL	LD templates	RESTful API	Apache Jena	Nginx	Yes	NA	Yes	Yes	Yes	Yes	NA	Physics, media and publishing, library science
GNOSS	JSON CSV, RSS, HTML, XML RDF, image, video, text,	OWL and RDF	RDF	RDF databases	SPARQL	Resource API, User API, Community API, SPARQL API, SPARQL endpoint	RESTful API	GNOSS framework	GNOSS Semantic Search engine	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Banking, education, health care, tourism, industry
BBN	RDF, JSON-LD	RDF	NA	Cassandra, BlazeGraphDB	SPARQL	SPARQL endpoint	RESTful API	Forge python framework	Elastic Search	Yes	Yes	Yes	Yes	Yes	Yes	NA	Neuroscience
GraphBRAIN	OWL RDF	RDF	XML, OWL, RDF	Neo4J	Cypher	NA	RESTful API	Graph BRAIN	NA	Yes	NA	Yes	Yes	Yes	NA	NA	Just prototype
Ontotext	Text RDF CSV	RDF	RDF	GraphDB	GraphQL R2RML	GraphQL SPARQL endpoint	RESTful API	RDF4J	Elastic Search	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Healthcare,life sciences, financial services, media and publishing industry
MAANA	Text RDF CSV	Knowledge Models	NA	RDBMS	GraphQL	GraphQL endpoint	RESTful API,	NA	Elastic Search	Yes	NA	Yes	Yes	Yes	Yes	Yes	Oil and gas industry and other industries
Anzo	RDF, text	RDF	CSV, JSON	AnzoGraphDB	SPARQL	SPARQL endpoint, Anzo Command Line Interface (CLI)	RESTful API, Graph Data Interface	Knockout JavaScript framework	Elastic Search 7.10.2 .	Yes	NA	Yes	Yes	Yes	Yes	Yes	Healthcare, life sciences, financial services, manufacturing, government, retail
OntoWiki	RDF	RDF	RDF XML	MySQL DB, Virtuoso	SPARQL	SPARQL endpoint	OntoWiki API	Zend Frame work	Elastic Search	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Library science,Distributed Semantic Social Network

Stardog Studio	RDF JSON CSV Text	RDF	RDF	SQL, NoSQL, Graph DBMS, RDF store	GraphQL SPARQL	SPARQL endpoint	Sesame Rest API, HTTP API	Spring framework	Apache Lucene	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Life sciences, financial, manufacturing
FEEKG	JSON JSON-LD RDF OWL	Entity – Event Data Model	RDF	Document store info Graph DBMS RDF store & Spatial DBMS	SPARQL, Prolog	FedShard ™	RESTful API, HTTP API	RDF4J and Jena interfaces	Apache Solr and MongoDB	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Healthcare, defense and intelligence, financial, banking, pharmaceutical, manufacturing
TBEDG™	XML CSV JSON RDFS OWL	RDF	RDF	Databases used for storing	SPARQL, Rules Engine, GraphQL	SPARQL GraphQL endpoint	RESTful API	Eclipse platform and the Jena API	TopBraid explorer	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Life science, financial services, engineering and design, education,health care, oil and gas
Siren Platform	CSV JSON RDFS OWL	RDF	ANB graph file	Neo4J,Stardog DB, Spark, Oracle SQL, Postgres SQL, Dremioprestp(virtual data warehouse)	ElasticSearch query language	NA	RESTful API	NA	Elastic Search	NA	NA	Yes	Yes	Yes	Yes	Yes	Life sciences, Law, Cyber security, telecommunications, fraud and threat analysis
PPSSGE	RDF, CSV, text	RDF	CSV, Excel, RDF	GraphDB,Stardog, MarkLogic	SPARQL	SPARQL endpoint	RESTful API	Apache Jena	Apache Solr, Elastic search	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Construction and engineering, government organizations Healthcare and pharma, banking and insurance, media and publishing, retail and e-commerce
Semaphore	CSV, OWL, text	RDF	NA	AllegroGraph, StarDog, Neo4J, MarkLogic, Jena, BlazeGraphDB, Amazon Neptune	SPARQL	SPARQL endpoint	RESTful API , Custom API	Apache Jena	Apache Solr	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Energy and utilities, government and intelligence Healthcare, life sciences, financial services and insurance, media and publishing, retail and industry
GKOS	Excel CSV, RDF, XML, JSON, TriG, Turtle	RDF	CSV RDF XML	GraphDB	SPARQL GraphQL	SPARQL GraphQL endpoint	RESTful API	RDF4J API	Apache Solr, Elastic Search	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Defense & Intelligence, Medical & Scientific, e-Commerce & Retail, Media & Publishing, Corporate & Legal, Library & Academic

Abbreviations Used: UWKGM : Unified Workbench for Knowledge Graph Management, **HAPE:** Heaven Ape, **NHA:** News Hunter Architecture, **LDH:** Linked Data Hub, **BBN:** Blue Brain Nexus, **FEEKG:** Franz's Entity-Event Knowledge Graph, **TBEDG™:** Top Braid Enterprise Data Governance™, **PPSSGE**: Poolparty Semantic Suite Graph Editor, **GKOS:** Graphite Knowledge Organization System, **RDF:** Resource Description Framework, **JSON:** JavaScript Object Notation, **JSON-LD:** JavaScript Object Notation-Linked Data, **CSV:** Comma Separated Value, **ER**: Entity Relationship, **OWL:** Ontology Web Language, **RDFS**: Resource Description Framework Schema, **XML**: eXtensibleMarkup Language, **API**: Application Programming Interface, **MySQL**: My Structured Query Language, **SPARQL:** Simple Protocol and RDF Query Language or SPARQL Protocol and RDF Query Language, **GraphQL:** Graph Query Language, **RDF4J:** Resource Description Framework 4 Java, **REST**: Representational State Transfer, **RDBMS:** Relational database management system, **DBMS:** Database Management System **NA:** Not Applicable

6 Findings

The review of KGMPs relied on two sets of features: basic and core features (discussed in previous section). For the developers, managers and potential users of KG, the review of these platforms can come in real handy as they will be able to decide which tool to choose for their project and also determine the limitations or lack of facilities that a tool may possess. Data were collected against the discussed features and it had descriptive, date, categorical, and not available (NA) data values. Descriptive data values are the factual information; date data express the timeline; categorical data values are 'Yes' that means known or available features and 'No' that means no such feature is available; and 'NA' means authors could not find information related to the particular feature (Gajbe et al., 2021). Below authors describe the findings about these tools based on the collected data.

6.1 Findings about KGMPs through Basic Features

6.1.1 Platform Origin and Current Status

Basic features describe the general information about the KGMPs. The first and foremost information is the developer of the platform. It was observed, most platforms (15 KGMPs) were developed solely by corporate companies as these technologies are heavily used in companies to manage huge amounts of data. Besides, there are few platforms like MONOLITH, GraphBRAIN, and NHA developed in company and academia partnership whereas the rest six KGMPs were developed by academic organizations solely like VADALOG is developed by University of Oxford; Spindra was developed by University of Arizona etc. Usually, developers leverage the source code to modify the existing or develop a new platform whereas users desire the platforms that are available freely. Considering the KGMPs origin it's quite evident that industry is more involved in the development of platforms and academia is lagging behind. The reason behind this may be companies have huge amount of data which can be mined to reveal insights (Sinha *et al.* 2021). There is more demand for KGMPs in the industry whereas academia is slowly realizing the power of such platforms.

The data about the FR, LR and LV gives a fair idea about development of KGMPs as KGM is the upcoming technology for data processing. But a few platforms like Ontotext, FEEKG have been under development from the early 2000s and now they have become advanced throughout the years. Few of the platforms like Metaphactory, Siren Platform, BBN, and GKOS were released recently in the last few years but have gone through a lot of transformation to become sophisticated and released new and advanced versions of the platform. A lot of platforms have released their latest version very recently in 2021 like Anzo, GKOS, and Siren Platform etc. One peculiar disadvantage of evolving technology is, its currency becomes obsolete very rapidly. This is also happening with such KGMPs. A lot of the KGMPs are coming up with newer versions of the platforms quite rapidly, so the challenge for the stakeholders of such platforms will always be keeping the system at bay with the upcoming versions of the platforms. In such cases sticking to a single platform becomes quite essential and the mode of availability of the platform is critical.

6.1.2 Platform Type and Licenses

It was observed that six KGMPs i.e., OntoWiki, BBN, Wikibase, TypeDB, LDH, UWKGM were OSS. Ten KGMPs were SBS; three KGMPs were available as FS and SBS both; one KGMPs were available as FS and the rest four KGMPs were prototypes so their ST was not available. Since most of the platforms were developed by companies, it was natural that most of the platforms had commercial license whereas a few platforms that were available as OSS like BBN has Apache License 2.0, UWKGM has BSD 3-Clause License etc. If platforms are available as OSS then it's up-gradation does not require cost. But if it's not available as OSS or FS then it's quite difficult to have the latest version always as it will incur cost. Most of the KGMPs have commercial licenses which may be costly for an implementer. It was quite evident that in the last five to six years more and more platforms have emerged on the forefront as the requirement, understanding, and utilization of the KG has increased.

6.1.3 Platform Deployment, Compatibility and Development Languages

Most of the platforms i.e., 17 KGMPs can be deployed both on local and cloud infrastructure. It was also observed that most of the platforms can be deployed on a variety of operating systems like Windows, Linux, and MAC. HAPE is a unique prototype platform, which has a client side browser operating on KG's knowledge base, while its server side has HAPE's OS termed as KGOS. Containerization is a concept that has come into existence in the last decade, through which various applications can be run on isolated user space (Pahl, 2015). It was observed nine KGMPs used Docker for containerization whereas one platform Anzo uses K8s and Ontotext, UWKGM uses both Docker and K8s. Software development and presentation also depends upon different types of computer languages. The most preferred languages used for development of KGMPs were found to be Java, Python, JavaScript, PHP and HTML.

6.1.4 Platform Usability, Domain Coverage and Documentation

20 KGMPs provide SSO facility for authorization which increases security of KGMPs, improves the usability of the platforms and reduces the overall cost. Among the 24 KGMPs under the study, 22 had BBGUI facility whereas one has both BBGUI, NGUI and one has NGUI facility. The facility of using the platform through a working demonstration site is specifically crucial for developers and users of platforms, as it facilitates to get acclimatization with software and evaluate its capabilities. It was observed that among the 24 KGMPs in the study, 12 KGMPs provided demonstration facilities and URI to these demonstration sites are also made available in Table II. These KGMPs can be compared in their efficacy based on publicly available KG like YAGO, NELL.

Except one platform, Spindra, all the platforms are domain independent allowing to manage KG from different domains. Spindra is a specialized platform for spatial domains. A user or even a developer of any software requires the detailed information of the software which is generally provided through documentation. It was observed that each of the 24 KGMPs provided documentations but they were provided in different flavours. Four platforms HAPE, NHA, MONOLITH, and GraphBRAIN were available as prototype platforms and provided scientific papers explaining their features. Among these MONOLITH, GraphBRAIN provided manual documents as well. For platforms like GNOSS, Semaphore, and MAANA, only white papers could be located online. Rest of the 17 KGMPs had detailed documentations available online. Metaphactory had both scientific paper and manual documents. For each of the platforms, URI of document location has also been provided in Table II.

6.2. Findings about KGMPs through Core Features

6.2.1 Data Ingestion, Transformation and Production

Core features explain the granular insights about the KGMPs which are really helpful to understand the full potentials. The first and foremost information for KG development and management is the kind of data sources on which the platforms are functional. KG development can take place from structured, semi-structured or unstructured sources. It was observed that the platforms under the study were very diverse in nature and though most the platforms allowed structured data format like OWL, RDF, a lot of tools like Metaphactory, Semaphore, PPSSGE etc. allowed ingestion of textual data as well as CSV, XML. Tools like UWKGM and NHA only allow ingestion of textual data. Thus, KG could be created by both bottom-up and top-down approaches based upon the data ingestion (Qiao et al., 2016). Top-down approach leverages the structured sources for developing KG whereas bottom-up leverages the semi-structured data like XML or unstructured data like text data (Wang & Hou, 2018). The bottom-up approach requires usage of ML models for knowledge extraction, relationship extraction which are readily available in platforms like MAANA, Ontotext, GNOSS etc. The data is mostly transformed into RDF format for development and management of KG as it's the de facto standard followed by semantic community for KR. There were few platforms that transformed the data into similar formats like FEEKG transformed data into Entity – Event Data Model, TypeDB transformed data into TypeDB schema ER model, HyperGraph and MAANA transformed into knowledge models. HAPE is a prototype KGMP, which is still under development and has different point of view knowledge structures. Hence it described five different knowledge structures as triplet pattern (similar to RDF); concept taxonomy; snapshot; topic and topic pattern; possible world to make knowledge operations such as inference, search, mining, prediction etc. more accurate on big KGs generated from big data. The facility of downloading the data makes the KGMPs more versatile in use. Similar to DTF, in DEF the most preferred format is RDF. Besides RDF, XML CSV, and JSON are the most allowed formats in which data can be downloaded from the KGMPs.

6.2.2 Data Storage, Access and Query

Database is a storage area where the data is stored. Most KGMPs use NOT ONLY SQL (NoSQL) databases as it has several advantages - reading and writing data quickly; supporting mass storage; easy to expand; low cost infrastructure. These NoSQL databases are of different types like key-value database (like Flare), column-oriented database (like Cassandra), document database (like MongoDB), graph databases (like Neo4J) (Han et al., 2011). Among these NoSQL databases the graph databases are top notch for handling complex, semi-structured, and densely connected data. It's swift in terms of queries and gives a response in milliseconds. These KGMPs use mostly graph database technology. A lot of KGMPs specifically mention the specific graph database that they use like Anzo uses AnzoGraphDB, Ontotext uses GraphDB etc. There are KGMPs like Metaphactory, Siren platforms compatible with many graph databases like StardogDB, GraphDB, Amazon Neptune, Blazegraph DB, AllegroGraph. Few of the platforms like UWKGM takes the text data, uses document and relational databases like MySQL, MongoDB whereas platforms like Stardog Studio uses both the relational databases and NoSQL databases. So, in terms of storage facilities KGMPs are found to be quite versatile.

Since most of the platforms use graph databases it was observed that for querying the data SPARQL, GraphQL and Cypher were the most preferred query language. There are few platforms which use few other query languages like TypeDB platform uses its own query language TypeQL; VADALOG uses Datalog along with SPARQL, GraphQL, Cypher and FEEKG uses SPARQL, Prolog. Since some of the platforms like Metaphactory, Ontotext also use relational databases for storing the data, these data are mapped in a RDF data model using a mapping language called R2RML. Such mappings provide the ability to view existing relational data in the RDF data model, expressed in a structure and target vocabulary of the mapping author's choice. Processors could, for example, offer a virtual SPARQL endpoint over the mapped relational data, or generate RDF dumps, or offer a LD interface (Das et al., 2012). To access the data sets preferred infrastructure was the SPARQL endpoints. Few platforms like HAPE, NHA and UWKGM used very specific endpoints Virtuoso. Besides SPARQL, GraphQL was also employed by platforms like MAANA and platforms like TBEDG™, GKOS, and Ontotext employed both SPARQL, GraphQL endpoints for providing access to the data.

Search facilities are generally provided with the help of specialized search engines. Here data reveal that the most preferred search engine was Elasticsearch. It allows a distributed, multi-tenant-capable full text search engine with an HTTP web interface and schema free JSON documents. It also allows a real-time GET request, which makes it suitable for NoSQL databases. Most of the KGMPs are built upon graph databases which are NoSQL databases thus making it a suitable choice. Besides Elasticsearch, Apache Solr was also another popular search engine used by platforms like GKOS, PPSSGE, Semaphore. A few platforms use other specific search engines like Metaphactory uses GraphScope; GNOSS uses GNOSS semantic search engine. SF leverages API to extract data from and write to RDF graphs. The most used SF are Apache Jena (https://jena.apache.org/), RDF4J. Apache Jena is an open source Java framework for SW and LD applications. It allows RDF and SPARQL support, an ontology API and reasoning support as well as triple stores. Similarly Eclipse RDF4J (formerly OpenRDF Sesame) is also an open-source framework for storing, querying, and analysing RDF data (RDF4J, 2021).

6.2.3 Vocabulary Management and Knowledge Graph Facilities

Metadata management is a key feature that supports organizations to increase productivity, compliance and scalability. Scalability is very much beneficial for KG as it describes the expansion of existing content, its structure and its evolution timeline which requires the definition to be revisited. This helps KG to evolve with time (Roszkiewicz, 2010). By connecting and organizing metadata as a KG, an overall better picture of the entire data ecosystem could also be seen. The feature of Metadata management was found in 22 KGMPs out of the 24 under this study. Thesaurus and taxonomy management and ontology management are features that help to build the data models on top of the data that are compliant to the organization regulations, SW standards and enhance facilities like navigation, search etc. The most important part of this facility is building data models which comply with SW standards allowing it to be used, reused, repurpose etc. for the objective at hand. The ontology management facility was found in 23 KGMPs whereas thesaurus and taxonomy management facility was found in 11 KGMPs. The publishing and consumption of ontologies and URI-based LD adhering to the RDF data model is seen as a promising strategy to increase coherence and facilitate information flows between various enterprise information systems (Galkin, Auer, & Scerri, 2016). These facilities of linked data management are an integral part of KGMPs and 23 of the platforms had. Once a KG is developed, the two facilities it should provide the user are search and visualization. These facilities allow one to understand the holistic view of the KG. The knowledge graph search was found to be available in all the KGMPs and knowledge graph visualization was available in 22 of the KGMPs.

6.2.4 Application Development Architecture, Data Analytics and Use Case

APIAS allows for developing new applications on top of the data that are managed by the platform or even new features for the platform (Moesif, 2019). The data about the KGMPs revealed that most of them use the RESTful architecture. It has simpler architecture, requires less expertise developing the communication model, and leverages the client's server side to store information which is relevant for the client. It utilizes smaller messaging formats, similar to existing other web infrastructure in terms of their architecture (Shafiq, 2020).

More often data scientists complain that they face difficulties to find the actual context from data that are provided to them. But KG connects those actual data thus, it can be leveraged to tap into the links of datasets and provide a context to data. Analytics performed on such datasets will be more accurate and will fetch better results. 17 KGMPs have the facility of DA. A few platforms like Metaphactory, Anzo specifically mention the support of specialized software like Konstanz Information Miner (KNIME), Power BI, and Tableau for performing the data visualization, processing etc. KGM technologies have risen in the last decade and have been implemented in various domains like life sciences, banking, library sciences, education, healthcare, manufacturing, engineering, media and publishing industries; using these KGMPs showing its versatile usage.

7. Implications

The present study performs a general review of KGMPs and has several theoretical and practical implications. Based on the literature review it was seen none of the existing studies like (Galkin et al., 2017) considered 24 platforms and 32 features for reviewing the KGMPs. Thus, this work is unique in terms of its scope, comprehensiveness and intent in studies related to KGMPs. KG is basically a big network of entities, their characteristics and the semantic connections. The study can be especially useful for choosing the most feasible KGMPS to be utilized by the KG managers, KG researchers and data scientists working with KG for connecting the data residing in data silos and executing queries against them swiftly. Graph databases have been a significant technology in facilitating data integration of structured and unstructured data. But to fully utilize the capabilities of semantic technologies the KGMPs can come in handy as it allows amalgamation of the semantic data model, or ontology, with the data in the graph database. Employing a semantic data model on top of the integrated data produces a true semantically enriched enterprise KG (Ivanov, 2018). KGMPs discussed here have all these facilities making this work an essential study for the stakeholders of KG. KGMPs discussed here support semantic data integration, data transformation, and analytics on the KG. The facility reveals insights about the data residing at different departments within the organization or different geo-locations which results in the optimized use of data for the organizations. The Table II also provides direct links to the demo sites of the KGMPs which can be visited by the potential users to evaluate functionalities first hand as well. The platforms leverage the standards of SW recommended by W3C's like RDF, OWL, SKOS and SPARQL, showing that platforms follow the best practices for superlative results. The use of such standards also points out that these KGMPs allow the agencies to incorporate various standard and community accepted vocabularies with their internal processes, systems, etc., to automate the data management process. Thus, this work acts as a one stop shop for the implementers of KG technology in their organization to explore a variety of platforms at a single place concisely.

8. Conclusion and Future Works

KGMPs are crucial for semantic data management as KG connects the datasets and provides context to them. The current study attempts to review the existing KGMPs based on two sets of features - basic features and core features. The study revealed that a lot of KGMPs have evolved in the last decade and these platforms are quite extensively used by corporate industries to manage the data from different data silos. A few KGMPs are also available as OSS and even some of the commercial KGMPs have free editions as well. All the KGMPs have browser-based interfaces, extensive documentation and 50% of the platforms have free demonstration facilities as well. The KGMPs can be ingested with all the different types of data i.e. structured, semi-structured and unstructured and that can be transformed into RDF data models. To store the data mainly specific type of NoSQL database called as graph databases are preferred. Besides the RDF data model, other standard SW technologies like SPARQL or GraphQL for querying the data, SPARQL endpoint for providing access to the data, Apache Jena and RDF4J as the SF is being used. Most of the KGMPs have the facilities like MM, OM, and LDM which can be leveraged for easier KGM and these KGMPs have been employed various domains like library sciences, life sciences, engineering etc. Platforms like Metaphactory, Ontotext, PPSSGE, Semaphore, TBEDG™, FEEKG, GNOSS, and Stardog Studio are quite comprehensive in nature and can be utilized effectively. Features of this study can be used as metadata elements for describing KGMPs as well as can be extended to design a metadata schema for KGMPs. In future we would like to evaluate these KGMPs by assessing the KG produced by these platforms using the same dataset based on the existing KG evaluation techniques.

References

[1] Bakker, R. (1987). *Knowledge Graphs: Representation and Structuring of Scientific Knowledge*. University of Twente, Enschede.

[2] Bellomarini, L., Fakhoury, D., Gottlob, G., Sallinger, E. (2019). Knowledge graphs and enterprise AI: the promise of an enabling technology. *In: I*EEE 35th International Conference on Data Engineering (ICDE), p. 26–37.

[3] Bellomarini, L., Gottlob, G., Pieris, A., Sallinger, E. (2018). Swift logic for big data and knowledge graphs". *International Conference on Current Trends in Theory and Practice of Informatics*, Edizioni della Normale, Cham, p. 3–16.

[4] Bergman, M. K. (2019). *A Common-Sense View of Knowledge Graphs*. https://www.mkbergman.com/2244/a-common-sense-view-of-knowledge-graphs/

[5] Berners-Lee, T., Hendler, J. A., Lassila, O. (2001). The semantic web, *Scientific American*, 284 (5) 34-43.

[6] Bilal, A. (2021). Domain-specific knowledge graphs: a survey. *Journal of Network and Computer Applications*, *185* (5). 103076, https://doi.org/10.1016/j.jnca.2021.103076

[7] Blumauer, A. (2014). *From Taxonomies over Ontologies to Knowledge Graphs*. https://blog.semanticweb.at/2014/07/15/from-taxonomies-over-ontologiesto-knowledge-graphs

[8] Carlson, A., Betteridge, J., Wang, R. C., Hruschka Jr, E. R., Mitchell, T. M. (2010). Coupled semi-supervised learning for information extraction, *In*: 3rd ACM International Conference on Web Search and Data Mining, p. 101–110.

[9] Chen, Z., Wang, Y., Zhao, B., Cheng, J., Zhao, X., Duan, Z. (2020). Knowledge graph completion: a review". *IEEE*

Access, 8, p. 192435–192456.

[10] Columbia University. (2019). *Knowledge Graph Conference*. http://sps.columbia.edu/executive-education/knowledge-graph-conference

[11] Containerization. (2019). IBM Cloud Education. https://www.ibm.com/in-en/cloud/learn/containerization

[12] Das, S., Sundara, S., R., C. (2012). *R2RML: RDB to RDF Mapping Language*. https://www.w3.org/TR/r2rml/

[13] Davis, R., Shrobe, H., Szolovits, P. (1993). What is a knowledge representation? *AI Magazine*, 14 (1) 17–33.

[14] Desimoni, F., Po, L. (2020). Empirical evaluation of linked data visualization tools. *Future Generation Computer Systems*. 112, p. 258-282

[15] DeVries, P. H. (1989). *Representation of Science Texts in Knowledge Graphs*. Ph.D. thesis,

University of Groningen, Groningen, The Netherlands, 1989.

[16] Dong, X. L., Gabrilovich, E., Heitz, G., Horn, W., Lao, N., Murphy, K., Strohmann, T., Sun, S., Zhang, W. (2014). Knowledge vault: a web-scale approach to probabilistic knowledge Fusion. *20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14*, pp. 601–610.

[17] Ehrlinger, L., and Wöß, W. (2016). Towards a Definition of Knowledge Graphs. *SEMANTiCS*. (Posters, Demos, SuCCESS), 48 (1-4) 2.

[18] Fensel, D., ªimºek, U., Angele, K., Huaman, E., Kärle, E., Panasiuk, O., Wahler, A. (2020). *Knowledge Graphs*. Springer International Publishing.

[19] Gajbe, S. B., Tiwari, A., Gopalji, Singh, R. K. (2021). Evaluation and analysis of data management plan tools: a parametric approach. *Information Processing & Management*, 58 (3) 102480. https://doi.org/10.1016/j.ipm.2020.102480

[20] Galkin, M., Auer, S., Kim, H., Scerri, S. (2016). Integration strategies for enterprise knowledge graphs. *10th International Conference on Semantic Computing (ICSC)*, p. 242–245.

[21] Galkin, M., Auer, S., Scerri, S. (2016). Enterprise knowledge graphs: a backbone of linked enterprise data. *IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, p. 497–502.

[22] Galkin, M., Auer, S., Vidal, M. E., Scerri, S. (2017). Enterprise knowledge graphs: a semantic approach for knowledge management in the next generation of enterprise information systems. *In:* Proceedings of the 19th International Conference on Enterprise Information Systems(ICEIS), Vol. 2, p. 88–98.

[23] Glimm, B., Stuckenschmidt, H. (2016). 15 years of semantic web: an incomplete survey. *KI-KünstlicheIntelligenz*, 30 (2) 117–130.

[24] Gottlob, G. (2019). *Knowledge Graphs and Enterprise AI/: The Promise of an Enabling Technology*. http://conferences.cis.umac.mo/icde2019/wp-content/uploads/2019/06/icde-2019-keynote-georg-gottlob.pdf

[25] Grabis, J. (2020). On-Premise or Cloud Enterprise Application Deployment: Fit-Gap Perspective. *In*: F. J., Œ. M., B. A., H. S. (Eds.), *Enterprise Information Systems* (p. 378). Springer.

[26] Han, J., Haihong, J., Le, G., and Du, J. (2011). Survey on NoSQL database. *In:* 6th International Conference on Pervasive Computing and Applications, p. 363–366.

[27] Hasse, P., Herzig, D. M., Kozlov, A., Nikolov, A., Trame, J. (2019). Metaphactory: A platform for knowledge graph management. *Semantic Web*, 10 (6) 1109–1125.

[28] Hedden, H. (2016). The accidental taxonomist. *Information Today*.

[29] Heist, N., Hertling, S., Ringler, D., Paulheim, H. (2020). *Knowledge Graphs on the Web— an Overview*. https://arxiv.org/abs/2003.00719

[30] Hjørland, B. (2008). What is knowledge organization (KO)?" *KO Knowledge Organization*, 35 (2–3) 86–101.

[31] Ichise, R., Kertkeidkachorn, N., Zhao, L., Rupu, E. F. (2018). Unified workbench for knowledge graph management". *EKAW (Posters & Demos)*, 45–48.

[32] Ivanov, Y. (2018). *What is an Enterprise Knowledge Graph and Why Do I Want One?* https://enterprise-knowledge.com/

what-is-an-enterprise-knowledge-graph-and-why-do-i-want-one/

[33] Ji, S., Pan, S., Cambria, E., Marttinen, P., Yu, P. S. (2020). A survey on knowledge graphs: Representation, acquisition and applications. *IEEE Transactions on Neural Networks and Learning Systems,* https://arxiv.org/abs/2002.00388

[34] Kapoor, B., Sharma, S. (2010). A comparative study ontology building tools for semantic web applications". *International Journal of Web & Semantic Technology (IJWesT)*, 1 (3) 1–13.

[35] Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Bizer, C. (2013). DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semantic Web*, 6 (2) 167-195

[36] Lenat, D. B. (1995). CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38 (11) 33–38.

[37] Marchi, E., Miguel, O. (1974). On the structure of the teaching-learning interactive process. *International Journal of Game Theory*, 3 (2) 83–99.

[38] McCusker, J., Rashid, S. M., Agu, N., Bennett, K. P., McGuinness, D. L. (2018). Why is the knowledge graph framework in action. *International Semantic Web Conference (P&D/Industry/BlueSky)*.

[39] Moesif. (2019). *API*. Moesif. https://www.moesif.com/blog/api-guide/getting-started-with-apis/

[40] Ngai, E. W., and Chan, E. W. C. (2005). Evaluation of knowledge management tools using AHP. *Expert Systems with Applications*, 29 (4) 889–899.

[41] Padmavathi, T., Krishnamurthy, M. (2017). Semantic web tools and techniques for knowledge organization: an overview". *KO Knowledge Organization*, 44 (4) 273–290.

[42] Pahl, C. (2015). Containerization and the PAAS cloud. *IEEE Cloud Computing*, 2 (3) 24–31.

[43] Paulheim, H. (2017). Knowledge graph refinement: a survey of approaches and evaluation methods. *Semantic Web*, 8 (3) 489–508.

[44] Qiao, L., Yang, L., Hong, D., Yao, L., Zhiguang, Q. (2016). Knowledge graph construction techniques. *Journal of Computer Research and Development*, 53 (3) 582–600.

[45] RDF4J. (2021). *The RDF4J REST API*. https://rdf4j.org/documentation/reference/rest-api/

[46] Roszkiewicz, R. (2010). Enterprise metadata management: how consolidation simplifies control. *Journal of Digital Asset Management*, 6 (5) 291–297.

[47] Rowley, J. (2007). The wisdom hierarchy: representations of the DIKW hierarchy. *Journal of Information and Communication Science*, 33 (2) 163–180.

[48] Ruqian, L. U., Chaoqun, F. E. I., Chuanqing, W. A. N. G., Shunfeng, G. A. O., Han, Q. I. U., Zhang, S., Cungen, C. A. O. (2020). HAPE: A programmable big knowledge graph platform. *Information Sciences*, 509, 87–103.

[49] Seitner, J., Bizer, C., Eckert, K., Faralli, S., Meusel, R., Paulheim, H., Ponzetto, S. P. (2016). A large database of hypernymy relations extracted from the web, *In:* 10th International Conference on Language Resources and Evaluation (LREC 2016), 360–367.

[50] Shafiq, S. (2020). *Why Is REST API Architecture Gaining Popularity in the Digital Industry?* Blog. https://dzone.com/articles/why-is-rest-api-architecture-gaining-popularity-in

[51] Sheth, A., Padhee, S., Gyrard, A. (2019). Knowledge graphs and knowledge networks: the story in brief. *IEEE Internet Computing*, 23 (4) 67–75.

[52] Singhal, A. (2012). *Introducing the Knowledge Graph: Things, not Strings*. Blog.

[53] Sinha, P. K., Dutta, B. (2020). A systematic analysis of flood ontologies: a parametric approach. *KO Knowledge Organization*, 47 (2) 138–159.

[54] Sinha, P.K., Gajbe, S.B., Debnath, S., Sahoo, S., Chakraborty, K. Mahato, S.S. (2021) A review of data mining ontologies, *Data Technologies and Applications*, Vol. ahead-of-print No. ahead-of-print. https://doi.org/10.1108/DTA-04-2021-0106

[55] Studer, R., Benjamins, V. R., Fensel, D. (1998). Knowledge engineering: principles and methods, *Data and Knowledge Engineering*, 25 (1/2) 161–197.

[56] Suchanek, F. M., Kasneci, G., Weikum, G. (2007). YAGO: a core of semantic knowledge unifying wordnet and wikipedia. *16th International Conference on World Wide Web*, p. 697–706.

[57] Tang, A., Avgeriou, P., Jansen, A., Capilla, R., Babar, M. A. (2010). A comparative study of architecture knowledge management tools. *Journal of Systems and Software*, 83 (3) 352–370.

[58] Torzec, N. (2014). Yahoo knowledge graph: making knowledge reusable at yahoo, *In*: 10th Annual Semantic Technology & Business Conference. https://www.slideshare.net/NicolasTorzec/the-yahoo-knowledge-graph

[59] Vrandeèiæ, D., Krötzsch, M. (2014). "ikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57 No. 10, p. 78–85.

[60] W3C. (2015). *Linked data*. W3C. www.w3.org/standards/semanticweb/data

[61] Wang, Y., Hou, X. (2018). A method for constructing knowledge graph of disaster news based on address tree". *5th International Conference on Systems and Informatics (ICSAI)*, p. 305–310.

[62] West, D. B. (1996). *Introduction to graph theory* (Vol. 2). Prentice Hall.

[63] Zhao, Z., Han, S. K., So, I. M. (2018). Architecture of knowledge graph construction techniques. *International Journal of Pure and Applied Mathematics*, 118 (19) 1869-1883.