

Standards Based Personal Knowledge Graphs

Omes Felix Baltes

Bachelor's Thesis – 15. July, 2022
Ruhr University Bochum
Faculty of Computer Science

Supervisor: Prof. Maribel Acosta
Advisor: Prof. Ajsa Fischer

© 2022
Omes Felix Baltes
ALL RIGHTS RESERVED

Abstract

Knowledge graphs have been leveraged in enterprise and open data for more than a decade, but for personal use, they are just emerging. Most of the tools for personal knowledge graphs are based on proprietary tech and data formats. This Paper explores how data-centric PKGs can be modeled using Semantic Web Standards and how to create usability focused Interfaces for them.

Contents

1	Introduction	1
1.1	Overview	1
1.2	Vision (for a PKG Ecosystem)	2
1.3	Approach	3
1.4	Outline (merge with approach?)	3
1.5	unsorted	3
2	Preliminaries	5
2.1	Related Concepts and Technologies	5
2.2	Related Work	5
2.3	unsorted	6
3	Modeling PKGs with SWS	8
3.1	The Structural Framework	8
3.2	Requirements of a PKG Graph Model	8
3.3	Types and Syntax	9
3.4	The different Data Layers	9
3.5	CRUD operations and effects	9
3.6	Advanced Semantics with OWL	9
3.7	unsorted	9
4	Semantic Markdown with RDF	11
4.1	Markdown Outliner in RDF	11
4.2	Advanced Markdown Flavors	12
4.3	A Semantic Markdown extension	12
4.4	Advanced Semantics (Owl etc)	12
5	PKG Ecosystem	13
5.1	Requirements for a PKG App Ecosystem	13
5.2	Analysis of the Software Landscape	13
5.3	Applying the Model	14
5.4	Abstracting away Technical Details of Semantic Web Technologies	14
5.5	Testing Usability	14
5.6	unsorted	14

6	Discussion	16
6.1	How will the tool be used by real users?	16
6.2	The Missing Global Ontology standard for Personal Knowledge	16
6.3	Why Graphs for personal Knowledge Bases?	16
7	Conclusions	17
7.1	SOLID Collaboration	17
7.2	Research Questions	17
	Glossary	18
	Acronyms	19
	References	20

List of Figures

5.1 short cap 14

List of Tables

4.1 a table 11

1 Introduction

Personal Knowledge Graphs (PKGs) are an uncharted research area [1], with unclear and ambiguous Definitions. PKGs HTML PKG

1.1 Overview

Current Tools for Personal Knowledge Management are not as powerful as needed to tackle the increasing problem of (personal) **Information Overload**. The Term PKG has recently been used by emerging Note-Taking Tools, some of them calling themselves “Tools for Thought” (TFTs). Although these tools advertise being able to manage a “second brain” or PKG, they have some crippling limitations, which make them unfit candidates for maintaining a lifelong PKG [Reference other PKG Book Chapter?]:

1. **based on proprietary Technology and Formats.** This prevents interoperability, integrating external knowledge, data ownership... and causes vendor lock-in, data silos...
2. **lack of expressivity (semantics).** They do not support formal structure or semantics...
3. **lack of support for mature query languages.** consequence of proprietary technology. only databases can support query languages, but TFTs are mostly based on text files, which they process to roll their own proprietary non-standard search, indexing and database systems.

In conclusion, there are no (relevant) Tools for creating PKGs based on open or standardised Technologies and Formats.

The Semantic Web Community is building standards for semantic Knowledge Graphs. Theoretically one could use Semantic Web Technologies and Tools to create PKGs; **Unfortunately however, the tools based on Semantic Web Standards (SWS) are notoriously hard to understand and use.** Expecting non-technical users to manage their personal knowledge with them is unrealistic.

...

There is potential here, just like bringing the Computer from Industry and Academia into personal homes in 19xx transformed the world by resulting in the internet, social media and smartphones; Bringing structured knowledge and databases, whose power is

well established in Industry and Academia, into the personal domain could usher a new age of Knowledge proliferation and collaboration.

With the need for PKG Tools based on open standards established, in this Paper I/We propose a Model, Structural Framework and Prototype to enable creating PKGs based on basic graph theory and SWS like RDF, OWL and SPARQL.

Furthermore the Domains of PKGs and Knowledge Management is lacking standardization, and is plagued by ambiguity of fundamental Definitions or interchangeably used Terms with different Meanings, e.g., Data, Information, Knowledge, Knowledge Base, Knowledge Graph, etc.

1.2 Vision (for a PKG Ecosystem)

Instead of haggling about definitions, motivation, etc, I want to highlight what future possibilities developing a mature PKG ecosystem can hold.

Due to the ambiguity of the Terms and the technological Expertise needed to navigate the field, The Vision of PKGs is hard to grasp for people that are not specialists in a Domain closely related to Information/ Computer Science. That's why here, rather than trying to come up with detailed definitions or features of the PKG Ecosystem, I want to highlight some of the possibilities this technology can enable. I think this will help grasp why this Technology is important and highlight the paradigm shift that this Technology will enable. This is a prerequisite to embracing it's development and enthusiasm.

- **Data centralisation, ownership, access rights, two way binding**. Imagine after changing your address you just have to update the adress in your PKG, share the Link to that Information, and all external Agents will always retrieve the most up to date address, Instead of having to update your adress across dozens of data silo locations. See access rights all in one place, and revoke at any time. - **Free Association of Knowledge**. Imagine you are visiting a Wedding in Spain with some wonderful people and are dancing to a Song. In a PKG you can freely associate / link all of this information, just like your brain does. You then have multiple associative retrieval paths to remember the song, because it is connected to the people, the wedding and the Spain trip. You can also freely associate any information with the song, for example make notes, which is not possible on vendor locked data, that is coupled with the UI, like Spotify. - **Interoperability/ portability**. Data centric applications means you are not bound by I/O changing or Applications / UIs changing.

1.3 Approach

Create a Knowledge Management Tool that

- is standards based
- easy to use
- creates a Personal Knowledge Graph

Important in this endeavor is to keep in mind what the PKG data model and PKG Apps need to be capable of. all the memex II features + what about new I/O and technologies. You need to look into the future, and that can only be achieved by using a data centric approach, that can be interacted with by any applications.

by standards based we mean leveraging Semantic Web Standards. By easy to use we mean the tool should have a good User Interface (UI) and User Experience (UX) and additionally should not require any technical expertise (Semantic Web, Knowledge Management or otherwise). For our understanding of a PKG refer to the appendix or [[relevant citation]]. The Format for saving the PKG should be founded in SWS, while work on standardising basic PKG features needs urgent work.

1.4 Outline (merge with approach?)

In chapter 1 we will propose a model for representing the structured Knowledge of a PKG with RDF and OWL; Chapter 2 will explore an extension to Markdown, to enable freely mixing structured and unstructured Knowledge. Then in chapter 3 we will analyse what a PKG Tool needs to be capable of and create a prototypical application to show how our model can be used to abstract away technical expertise of SWS by focusing on Usability and UI

1.5 unsorted

A PKG is a lifelong second brain; it includes your memories, knowledge, thoughts. Vendor Lock-in, Data Silos, Proprietary Formats are out of the question. Privacy, Data Ownership, Control and Interoperability are necessities not up for discussion. while limitations 1-3 can be solved by creating a model for standards based pkgs, limitation 4-5 need to be addressed on the application level.

currently pkg tools are the main thing, but going forward data needs to be the focus, not tools. Like there are .png files, and many different tools can work with .png files. Imagine

data centralisation allows freely associating your knowledge

unfortunately the innate complexity of Knowledge management and related terms means, that people outside the field think of it like some kind of phantastical IA device that's not realistic or comprehensible.

I just want to give some quick examples of the benefits a data centric approach to personal knowledge management could bring and how this breaks a lot of status quos in how we use software and how we have come to accept a lot of it without questioning it

1. change adress in one place. benefits also you can always see who has access to your information and revoke it at any time. unlike with data silos of external vendors, where you don't always have control of how to delete and update your data. 2. wedding trip with favorite music and persons all interlinked 3. commenting and linking freely unrelated to external software 4. porting your media librarys freely between services, interacting with people on other platforms than yourself

while these ideas where facing unsurmountable hurdles in the 1950 century (history knowledge graph) by now all of these capabilities and more are possible.

Problem: Except for specialists, almost no one is aware of Semantic Web

- how can these visions be merged with the needs of today by basing Knowledge Management on a Graph Framework, how this graph framework can be implemented with Semantic Web Technologies and an evaluation of PKG Tools and their feature requirements to solve this complex topic of creating digital "second brains" - **Information Overload.** Ever increasing amount of Knowledge (papers, bookmarks, media libraries, mail, calendar, tasks, projects), but human brain capacity for storing and processing information is the same as thousands of years ago. - personal data is spread out in external data silos

- A noticeable phenomenon in the history we have sketched is the never-ending growth of data and knowledge, in both size and diversity. [A noticeable phenomenon in the history we have sketched is the never-ending growth of data and knowledge, in both size and diversity.](<https://www.notion.so/A-noticeable-phenomenon-in-the-history-we-have-sketched-is-the-never-ending-growth-of-data-and-knowl-d844784676ec4cd8bd3c861b3c43b798>)

pkg is more like a .mp3 or .png file, than it is like photoshop

2 Preliminaries

2.1 Related Concepts and Technologies

- Personal Information Management - Memex / Tools for Thought - Knowledge Bases -
Augmenting Human Intellect - Data Privacy and Ownership - Semantic Web Standards
— RDF, OWL, SPARQL - Ontologies and Open Vocabularies - In-Memory Triple Store
— RDF-JS spec

2.2 Related Work

Related Work

Personal Information Management

Personal Knowledge Bases (merge with above)?

Semantic Web usability

****How would a memex2 based on RDF work?****

- In the cloud, with multi-device sync - stored in RDF - can integrate external Data
Sources - just the ones the user is interested in - structured information in the graph can
be queried - collaboration is possible - this needs access rights → handled by the server.
based on RDF metadata? - how would you manage write tokens / keys in RDF? thats
stupid actually - - reference tools that mention the need of - privacy, - collaboration -
read - comment - write - - access control,

PIM Tools

- Notion, Roam, etc.

2.3 unsorted

do pkgs need to be spiderwebs, with the person in the center, or are graph datasets enough?

This is relevant when thinking about merging PKGs or collaboration on PKGs

Say you have a library and “Dieter” likes certain movies, while “Anna” likes other movies. if there is no connection to the liked movies, how would the merged KG represent which person likes what?

- **Spider Web PKG?** While PKGs are by some Authors suggested to be Data “about” a Person and its Relationship to Entities (that are not necessarily public) in his Personal Domain, PKGs can also include a Persons knowledge about personal or public Entities, or just Knowledge Management of Information and Media Domains relevant to the user.
- Workshop on PKG Introduction

The concept of **personal knowledge graphs (PKGs)** has been around for a while, in recognition of the need to represent structured information about entities that are personally related to a user. However, several open questions remain: (**comment**: I would argue the need is **information overload**, and pkgs are about information and knowledge RELEVANT to it’s user)

- **Definition** — The notion of a personal knowledge graph has been established loosely, as a resource of structured information about entities personally related to its user. This definition needs crystallization: What is personal knowledge and how is it represented in a PKG? What differentiates a PKG from general KGs, how are they related? How can PKGs benefit from information stored in general KGs and how is the benefit realised? How is work on PKG related to work in areas such as commonsense KGs and entity/event-centric understanding?
- **PKG construction/population** — What are the potential data sources (textual, visual, geolocation, etc.)? How to mitigate the ‘dual use’ of automated technology for PKG construction/population, since it can be used to extract and exploit personal knowledge about others?
- **PKG utilities** — What novel application scenarios would PKGs enable and what role does/can novel techniques such as semantic technologies and knowledge modeling play in this respect? How do PKG compare to existing solutions to these applications?
- **Practical realization** — Where would PKGs be stored and how would these interact with a range of external services, while considering access control as well as privacy concerns of users?
- knowledge graphs concern public / global entities - global entities / persons
- global knowledge - problems - rules out many entities people interact with in private domain - not just information regarding yourself, but also non-public knowledge you have

While early imaginers of PKBs and PKGs were limited by technological hurdles, these hurdles have all but disappeared today.

- Semantic Web Standards - RDF - RDFS - RDF* - OWL - SPARQL

general KGs are about consensus schema, interlinking and inference. PKGs are about centralisation, PIM, collaboration.....

3 Modeling PKGs with SWS

3.1 The Structural Framework

- the fundamental data structure limits what you can store and express with the model — we want to express everything from atomic concepts (vocabulary) to large topics (Projects, Science) - limitations of Trees/Documents - single parent Nodes - Freely associating Nodes is not possible - benefits of Databases - association - Transclusion (embedding) - structure - Querying - Benefits of Graph - adaptable schema - Query evaluation - Benefits of SWS - semantics - open vocabularies - integrating external Graphs - limitations of SWS / RDF - RDF* - performance

3.2 Requirements of a PKG Graph Model

- **Structured and Unstructured Knowledge.** Both academia (memex, augmenting human intellect) and industry (notion, roam) arrived at the conclusions that a personal knowledge base should be capable of representing structured information (databases, knowledge graphs), as well as unstructured Information (hypertext, markdown) [cite memex 60 years later etc] - structured Knowledge — as close to SWS as possible - unstructured Knowledge — Markdown Notes - **bi-directional links?** automatic generation of backlinks. (investigate owl property types; inverseProp etc.) - **Node exist quantor.** Nodes need to be atomic entities. For RDF each Node has a Class Triple (fallback Class Resource) - **Node IRI.** Every Node needs exactly one IRI - **Relationships?** In RDF Nodes with ‘rdfs:type rdfs:Property’ - **Arcs?** Called Triples in RDF (instances of Relationships) - **Unique human readable label.** every Node needs a **unique?** human readable label (2 sources of uniqueness, IRI and Label) - unique, because for the user it’s the only visible “adress” of the entity? - IRIs are not human readable - pkgs are supposed to be operated by non-technical users.

3.3 Types and Syntax

- Types - Node (BlankNode?) - Relationship - Literal (String or Datatype) - Casing
- Nodes are 'PascalCase' - Relationships are 'camelCase' - Classes are SCREAMING SNAKE

3.4 The different Data Layers

1. static external Schema (Ontologies) 2. adaptive personal Schema 3. Structured Data and Notes - extra Note Data Layer? 4. temporary Data Layer (inferred from Notes, external Data, etc) 5. Metadata for display properties (order, sorting, filters, alignment)

3.5 CRUD operations and effects

3.6 Advanced Semantics with OWL

3.7 unsorted

the data storage aspect is the most fundamental and limiting factor when designing a PKB Tool, because it defines what you can express and store.

Text, Trees and Graphs

trees are obv. a subset of graphs, however thats like saying a masterfully crafted sculpture is a subset of clay. specification creates unique structural properties, and a graph lacks the hierarchy and probably more importantly the chronology that a Book / Tree provides. This chronology provides the user with a sense of location and navigation, that is urgently missing from graphs.

Data model, structural framework, architecture

examples of standardised RDF formats; iCal, Contacts, Musicbrianz, etc

PKM putting everything into hierarchy makes free association impossible. This leads to copying of Information, with some of it outdated.

anyone who has seen a TOC knows text is a ordered tree. It's time we went from "Text" Information Media like Articles and Books, to Graph based Media, like Databases and Knowledge Graphs

how to model PKG as a data resource for Apps

- schema needed per data domain - media - health - contacts - calendar - resources -
...

Thesis: RDF / graph naturally lends itself to PKG TFT functionality, because it actually is a graph. So things like bi-links etc get easier to implement

- Notes

converting unstructured → structured knowledge must be frictionless

- It is worth mentioning that 1. lots of people are not satisfied with SWS and even the creators acknowledge Problems in design (better rdf repo) and 2. standards can change and be extended (RDF*) thus the basis for the model is graph theory, not SWS theory. For now the selected graph model will be RDF

4 Semantic Markdown with RDF

- differentiate semantic nodes vs text nodes - semantic nodes are entities with types - text nodes are unstructured knowledge that exists as nodes in the graph, but are also arranged in hierarchy attached on a semantic entity

cell1	cell2	cell3
cell4	cell5	cell6
cell7	cell8	cell9

Table 4.1: a table

4.1 Markdown Outliner in RDF

It is possible to display Text stored in a Graph like it was plain text or markdown.

There is however the question of how to handle metadata about display order and style.

results in:

Lady Gaga

rdfs:type→ [Person](<https://www.notion.so/Person-dece8f28272f428789d1a0cd875982d4>)
, [Artist](<https://www.notion.so/Artist-62b70a2518e447528a7165ed48022d22>) , [Musician](<https://www.notion.so/Musician-cd6095c9ba374616ad2ca44e06981729>)

foaf:givenName→ “Stefani Joanne Angelina”

foaf:familyName→ ”Germanotta”

[Lady Gaga got a lot of attention for wearing unorthodox dresses during events and performances](<https://www.notion.so/Lady-Gaga-got-a-lot-of-attention-for-wearing-unorthodox-dresses-during-events-and-performances-9bd88b658e5a44fd8054b8e670cd5ec3>)

[example for exotic dress nr. 1](<https://www.notion.so/example-for-exotic-dress-nr-1-b4b1125f9afc4c9e86c5a49e1a12d95c>)

[example for exotic dress nr. 2](https://www.notion.so/example-for-exotic-dress-nr-2-a9a21d52e922480d8c40d68cd9770cd4)

[example for exotic dress nr. 3](https://www.notion.so/example-for-exotic-dress-nr-3-02c3074cac824313b1fdb957f7c9ca38)

Note that all of the unstructured text notes are still their own entities, just designated with a `rdf:type` of `j:Note`.

4.2 Advanced Markdown Flavors

Markdown although not standardised, is embraced by the Web Community and continuously extended. There are several Flavors (Github, CommonMark, etc.). Approximately the following expressivity levels of Markdown have developed:

- **Basic Markdown.** Includes mostly Text formatting: - Headings, **Bold**, *Italic*, Strikethrough, Quotes, 'Code' - Lists - Images and Links - **Extended Markdown.** Includes advanced Formatting options: - Tables - Heading ID's (in-document navigation) - Syntax highlighted Code Block - Footnotes - Todos - Emoji - Highlighting - Sub and Superscript - Table of Content - Callouts - Comments - Captions - **"Hypertext" Markdown.** Recently Note-Taking Tools have adopted shared extended Syntax, even enabling toolwide Hyperlinks: - '[' automatically gets converted to links - '(' mention - " embeds - '\$\$' LaTeX code - ' Highlight Text

4.3 A Semantic Markdown extension

This takes the approach of Hypertext markdown and adds semantic relationships to it that can be used inside the PKG. These semantic relationships are then inferred into one of the PKG Datasets graph layers.

- inferred links from markdown nodes: - mentions - embeds - links - A proposal for semantic markdown could look like this: - '[' link `rdfs:seeAlso` - '(' mention - " embed - 'i' Relationship - 'j' inverse Relationship - '::' - '@@' - '&&' - ';;' - '

4.4 Advanced Semantics (Owl etc)

These need some kind of Syntax... - inverse - Felix —parent*i* —child*j* Omes - functional : x eindeutig auf y - company —foundingYear== 1960 - inverseFunctional : y eindeutig auf x - Marion ==bioMotherOf— Omes - transitive - ancestor - symmetric - obama :hasspouse Michelle - antisymmetric - :parent - reflexive - knows - irreflexive - married

5 PKG Ecosystem

- A PKG Tool criteria must work with a graph or database and allow unrestricted linking of entities - otherwise it's just KM Tool

The prototype is about UI and UX, not PKG Ecosystem feature show-down!

5.1 Requirements for a PKG App Ecosystem

- memex 60 years later ****must have**** functionality - and everything from yourself

5.2 Analysis of the Software Landscape

- include current State of PKM - web app, to reduce UX friction of install, login etc
- reference to other PKB Tools - what basic features are needed? - bi-links - sets - overview of - in/out arcs - in/out mentions - what are advanced features the model is capable of and could be explored? - even though a prototype was implemented on SWS compatible Graph structure, that was more to showcase how easily implemented these features are on a real graph structure, and to show hard to understand SWS Concepts can be abstracted away.

Include all algorithms from code? how things are inserted into triple store etc?

- required structure of Data: - define what a Node is (Class Node?) - define what a Title is (Literal Label?)

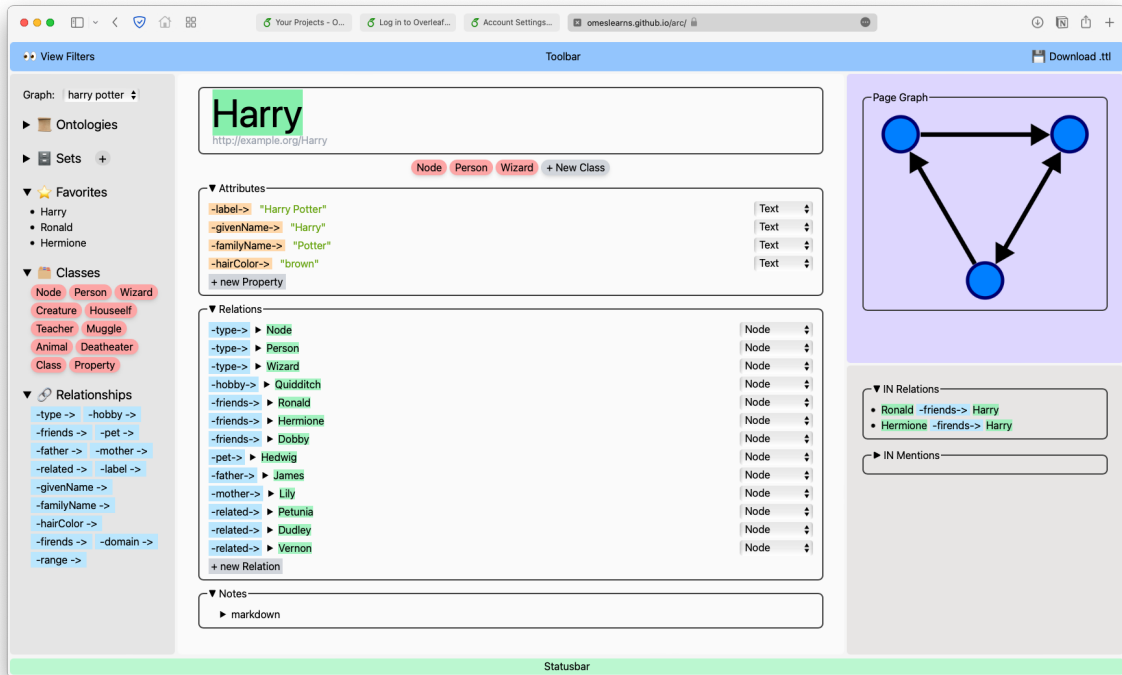


Figure 5.1: long caption

5.3 Applying the Model

5.4 Abstracting away Technical Details of Semantic Web Technologies

5.5 Testing Usability

5.6 unsorted

current note taking tools only save “snapshots” of files in themselves, while we need the actual files (audio, video, pdf, text etc) to be workable and openable in any program. pkg needs to be a file system / cloud drive

Knowledge Graph Summarization: store only what’s relevant to the user

Premise Pkg stored in a cloud. Like social media posts or cloud drives, set visibility in the graph

integration of external knowledge by clicking add node / triple on external RDF data

****Abstract 0.1****

Semantic Web Technologies have been slow to adopt / gain traction outside of the academic areas. While there are relevant use cases in the industry, they are living a niche dasein and are a tough sell. the lack of education and accessible tools for SWT means these technologies don't get the spread necessary to really be usable and enable a semantic Web experience. This leads to SWT being sparsely used on the internet, with the user-base being mostly academics, and sparsely in the industry.

This paper presents the implementation of a tool that tackles the problem of making a Tool (Browser Extension) that makes SWT approachable to the layman internet user. The hope is, that such a tool can drive adoption through implementation in Browsers, creating awareness of the technology and it's benefits for regular people and the industry.

To be able to reach laymans, the software needs to have exceptionally easy UX and availability.

This means:

- no installation process - no required configuration / setup - no prior knowledge about the technologies and terms involved

To achieve these goals, the optimal case would be for Semantic Web Technologies to be integrated into browsers as a feature. Just like users are able to interact with websites now, in the future they should be able to browse and enrich / extend the semantic web.

Semantic Web Technologies, should be embedded into browsers, the same as features like having tabs, creating bookmarks, password managers or having a history visualization.

we will explore the matured state of these ideas due to evolution through time, as well as through the removal of technical hurdles.

We are building a web app, for usability sake, but also because semantic web is about WEB

When ****loading** RDF into a human readable editor, you get a load of machine relevant meta crap******

- Knowledge Graph Summarization: store only what's relevant to the user

6 Discussion

- people have been struggling to develop better mediums than text, for mainly x reasons: - people learn to read and write in school, they don't learn information science or databases -

6.1 How will the tool be used by real users?

6.2 The Missing Global Ontology standard for Personal Knowledge

6.3 Why Graphs for personal Knowledge Bases?

comparison of Text Documents / PDFs / Books to the Graph structure (hierarchy, chronology, sense of orientation / navigation)

- Use Cases of KG - Open - Enterprise - Personal - Collaborative PKGs?

7 Conclusions

7.1 SOLID Collaboration

7.2 Research Questions

- How can PKGs be stored in the cloud - Access rights - Knowledge graph summarization

Glossary

HTML Hypertext Markup Language 1

Acronyms

PKG Personal Knowledge Graph 1

References

- [1] K. Balog and T. Kenter, “Personal knowledge graphs: A research agenda,” *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*, 2019.