World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:9, No:6, 2015

Building a Hierarchical, Granular Knowledge Cube

Alexander Denzler, Marcel Wehrle, Andreas Meier

***Abstract*—** A knowledge base stores facts and rules about the world that applications can use for the purpose of reasoning. By applying the concept of granular computing to a knowledge base, several advantages emerge. These can be harnessed by applications to improve their capabilities and performance. In this paper, the concept behind such a construct, called a granular knowledge cube, is defined, and its intended use as an instrument that manages to cope with different data types and detect knowledge domains is elaborated. Furthermore, the underlying architecture, consisting of the three layers of the storing, representing, and structuring of knowledge, is described. Finally, benefits as well as challenges of deploying it are listed alongside application types that could profit from having such an enhanced knowledge base.

***Keywords*—** granular computing, granular knowledge, hierarchical structuring, knowledge bases.

I. Introduction

KNOWLEDGE is based on multiple accumulated pieces of information that are assimilated, structured, and then stored in the human brain [8]. Minsky [12] adds to this that any small fragment of information that is not connected to a large knowledge structure is meaningless. The reason for this lies in the way in which humans learn. According to Ausubel et al. [4] the most important factor in learning is what the learner already knows because already acquired knowledge is stored and structured. and this can facilitate the process of assimilating new information. This is achieved by relying on reference points that indicate how and where to allocate new information. Through this, humans manage to learn faster and more efficiently.

Furthermore, several studies suggest that the knowledge structure should consist of multiple hierarchical levels [6], [13], [26], [27]. Through this, human beings manage to better cope with large volumes of information, thus improving their efficiency in reasoning and in the processing of information.

The concept behind granular computing follows a similar approach, emphasizing the creation of a hierarchical structure consisting of multiple levels. Within each level, granules can be found that contain information drawn together by factors such as indistinguishability, functionality similarity, or proximity [28].

Multiple levels are utilized to express different degrees of granulation, which can be differentiated as rough, middle, or fine. While granules of fine granulation are located at the very bottom, due to their maximum degree of detail, rough ones are situated at the very top, being the most summarized. Granules of middle granulation are placed in between and are the only ones that can both be divided and aggregated further.

This research paper aim is to explain how the concept behind granular computing can be applied to a knowledge base in order to build a so-called granular knowledge cube that serves as an enhanced knowledge base for applications.

For this purpose, the research paper has been structured as follows: In Section II, the definition of a granular knowledge cube is given, followed by an elaboration of its intended use in Section III. The underlying architecture is described in Section IV alongside an example that illustrates the resulting outcome in Section V. Furthermore, benefits and challenges are stated in Section VI in addition to an overview of application fields of the granular knowledge cube in Section VII. The conclusion is presented in Section VIII.

II. Defining a Granular Knowledge Cube

A granular knowledge cube relies on graphs, which consist of concepts $c_1, c_2, \ldots, c_n$ and relationships $r_1, r_2, \ldots, r_n$ among concepts, as a means to represent information. Concepts are assigned to granules $G_1, G_2, \ldots, G_n$ with granules being structured in a hierarchical way consisting of multiple levels $l_1, l_2, \ldots, l_n$. Furthermore, all existing granular dependencies $d_1, d_2, \ldots, d_n$ are indicated. In order to illustrate the mentioned notations, an abstract example of a granular knowledge cube is presented in Fig. 1.

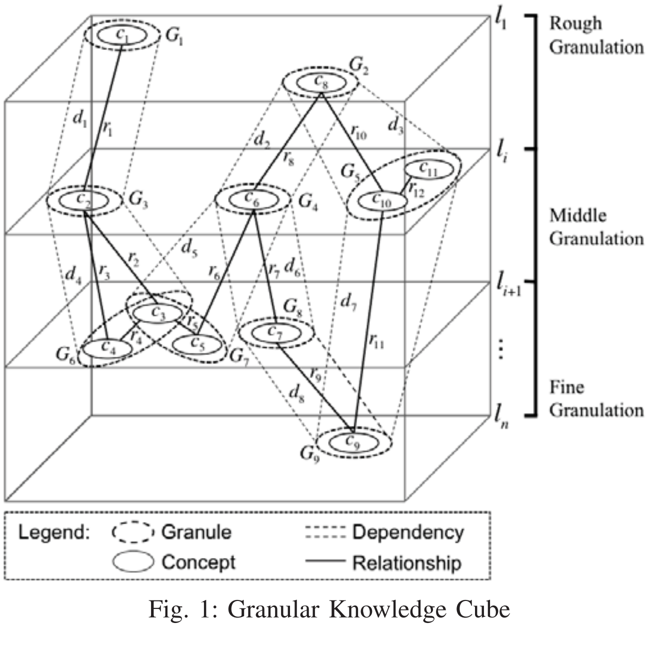


Fig. 1: Granular Knowledge Cube

A. Denzler is with the Information Systems Research Group at the University of Fribourg, Boulevard de Perolles 90, 1700 Fribourg, Switzerland (Email: alexander.denzler@unifr.ch).

M. Wehrle is with the Information Systems Research Group at the University of Fribourg, Boulevard de Perolles 90, 1700 Fribourg, Switzerland (Email: marcel.wehrle@unifr.ch).

A. Meier is with the Information Systems Research Group at the University of Fribourg, Boulevard de Perolles 90, 1700 Fribourg, Switzerland (Email: andreas.meier@unifr.ch).

The used notations are defined as follows:

A. Concepts

Concepts are the smallest entities in a granular knowledge cube. They stand for singleton information units that are either extracted automatically from content, using sophisticated data-mining techniques, or supplied manually by humans.

B. Relationships

Relationships between concepts are present intragranularly as well as intergranularly and are hierarchical. They are illustrated through the use of either undirected or directed graphs. The type of graph depends on whether simple connections are to be drawn, in which case undirected, unlabeled graphs should be used or semantic relationships, prompting the use of directed, labeled or unlabeled graphs.

C. Granules

On the same hierarchical level, granules have similarities with clusters, as their primary purpose is to group concepts together that share same or closely related properties. Granules are permitted to overlap with other granules in order to ensure that concepts can belong to two or more granules at once. A granule can belong to just one hierarchical level at a time. In addition, it is mandatory that all concepts be placed into granules and that a granule has at least one concept in order to be allowed to exist.

D. Hierarchical Levels

The hierarchical structure consists of multiple levels, with one top and bottom level and an undefined number of middle levels in between. However, each level needs to hold at least one granule inside in order to exist. The number of layers is influenced by the underlying data and algorithm used for building the hierarchical structure. Furthermore, each level represents a different degree of granularity.

E. Granular Dependencies

Granular dependencies are used to indicate and assess the degree of relatedness between granules located in different levels, regardless of the number of levels in between. A possible approach to determining the degree of relatedness between two granules is evaluating the number of relations that are shared intergranularly between concepts.

III. Intended Use

The primary use of a granular knowledge cube is to serve as a centralized or distributed knowledge base. For this purpose, information has to be stored and made available in a way that allows applications to make accurate and fast decisions. In order to achieve this, basic as well as enhanced functionalities are provided that can be used to improve applications efficiency and potency.

One reason for picking a granular knowledge cube is due to its granular structure, which ensures that concepts are drawn together and allocated to topics, based on a predefined set of parameters. This allows knowledge domains to be identified, which stretch vertically as well as horizontally in the cube. Knowledge about such domains can yield several advantages for applications that want to know more about a concept surroundings and relative location. Furthermore, the structure can reveal interesting regularities as well as irregularities.

Another reason for choosing a granular knowledge cube is due to its ability to handle different data types. Natively, it can cope with structured, semi-structured, and unstructured data. This is ensured through knowledge representation methods that are initially used to extract concepts from any source and type of data.

IV. Architecture of a Granular Knowledge Cube

The architecture of a granular knowledge cube consists of three layers. It includes a layer responsible for storing, one for representing, and another for structuring knowledge. They are structured in ascending order, with the storage layer being the first, representation the second, and structuring the third layer. The order derives from the fact that initially, a database that is capable of storing the represented knowledge needs to be chosen. The represented knowledge itself then serves as a preliminary step upon which any kind of structuring can be performed.

Within each of the three layers, different methods and tools can be deployed to fulfill the particular layer purpose. This openness and flexibility is vital, considering the fact that the cube should be capable of coping with different constraints and environments.

The cube can be built using a manual, semi-manual, or fully automated approach, which has a direct impact on the choice of methods in the representation and structuring of layers. However, only the use of a semi-manual and fully automated approach is viable, which will be justified later on. The semi-manual approach requires an external source to specify how information is interrelated, while in the fully automated approach, algorithms perform this task autonomously. Fig. 2 shows both approaches and the impact they have on the portfolio of methods that can be used.

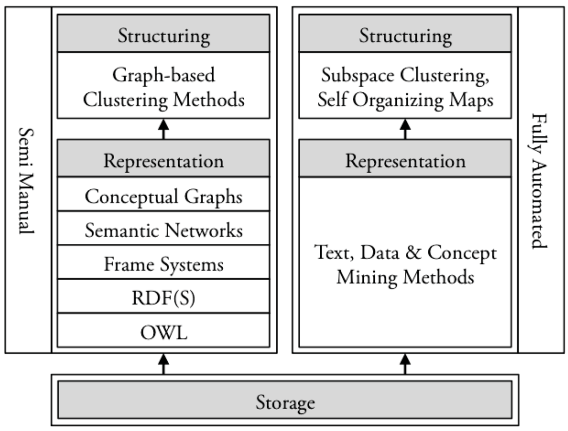


Fig. 2: Semi Manual & Fully Automated Approach

A specific elaboration of methods and tools that can be used within each of the three layers will be given in the following subsections, with storing of knowledge in A, representation of knowledge in B, and structuring of knowledge in C.

A. Storing of Knowledge

In the first layer, a database solution needs to be chosen that allows concepts and relationships to be stored. Both are best stored as a graph, which means that a database type needs to be selected that natively supports the storing and querying of graphs.

For this purpose, graph databases should be considered a viable candidate, as they have been developed exclusively for storing graphs. Furthermore, they provide the necessary features and functionalities to perform operations commonly used in the domain of graph theory.

After identifying graph databases as the most suitable type of database for storing graphs, the next steps consist of choosing a tool that can be deployed. For this task, a wide range of different candidates that differ in certain characteristics can be chosen.

In some cases, a graph database is built on top of another non-relational data model, while in others, it is a single, standalone solution. Another difference derives from the purpose and environment for which the graph database has been developed. While Web-based solutions aim to maintain low latency times for queries, others focus on handling large graphs by scaling horizontally. Still others have been developed and are optimized in a way that allows algorithms to be processed as quickly as possible by storing the entire graph in memory [20].

In addition, different numbers and types of features and functionalities are available throughout the existing tools. This has been evaluated and published in a study by Angles and Gutierrez [3] along with an empirical comparison of graph databases by Jouili and Vansteenberghe [9].

Therefore, in order to select the best-fitting tool, it is initially necessary to assess requirements and constraints that a graph database tool has to fulfill.

B. Representation of Knowledge

The second layer's purpose is to build a representation of knowledge, consisting of a set of concepts $c_1, c_2, \ldots, c_n$ and meaningful relationships $r_1, r_2, \ldots, r_n$ among them, that has been abstracted $a_1, a_2, \ldots, a_n$ from information entities $e_1, e_2, \ldots, e_n$. The described notation is shown in Fig. 3.

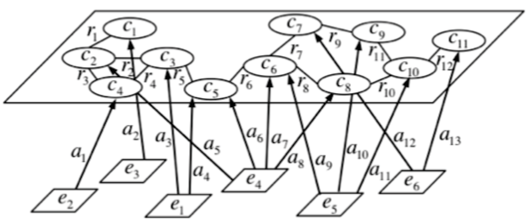


Fig. 3: Representation of Knowledge

The resulting representation of knowledge in Fig. 3 is displayed in a two-dimensional map. However, no restrictions are imposed with regard to the degree of dimensionality that a knowledge representation can have. This is essential due to manual approaches tending toward low-dimensional maps for this purpose, while automated ones favor multi-dimensional vector spaces.

In order to build a knowledge representation, it is necessary to first gather all relevant information entities. An information entity resembles a construct that contains raw information, which can be in an un-structured, semi-structured, or structured state. An example of an entity would be a post, a website page, or an article.

In a second step, all relevant concepts have to be located and then extracted from each entity. Because an entity can contain several concepts at once, it is possible that several abstractions are present per entity.

The third and final step concerns itself with the interrelation of concepts and therewith the construction of the knowledge representation. For this purpose, a method needs to be selected that provides the tools required to draw meaningful relationships between a set of concepts.

A semi-manual approach offers more possibilities for this task, compared to a fully automated approach, but requires a human being to decide which concepts to extract from information entities and how to interrelate and define the meaning of the relationships. The most commonly used representation methods for the semi-manual approach belong to either the group of formalisms or the group of Semantic Web languages.

While formalisms are graphic notations for representing knowledge in patterns of interconnected nodes and arcs, Semantic Web languages are about representing vocabulary of a particular domain or subject with relationships between concepts through the use of metadata [18].

Representatives from the domain of formalism methods include conceptual graphs [21], frame systems [12] and semantic networks [16]. Conceptual graphs focus on a logic-oriented approach, while semantic networks are built around the concept of semantic memory models [17] which is a non-logic-based approach [19]. Frame systems were developed as an alternative to semantic networks but with a clearer focus on being a logic-oriented approach.

While knowledge representation formalisms are primarily used in smaller and closed environments, their use is simpler to standardize and promote. Semantic Web languages, on the other hand, rely on metadata as a means to help machines to understand the meaning of Web-based data, hence the term Semantic Web. Some of the commonly used methods from this domain include standards proposed by the W3C, such as the Resource Description Framework Schema (RDFS) [25] as well as the prominent Web Ontology Language (OWL) [24]. The difference between them is that RDFS serves as a method for defining the structure of data, while OWL is used to describe semantic relations, hence why OWL's expressiveness is significantly higher, especially with regard to making logical expressions.

What all of these methods have in common is that the possibility of expressing taxonomies is provided, which is of vital importance for the creation of hierarchical structures in the next layer. Further, it is possible to express logical rules, which applications can use for reasoning purposes. However, this approach has its drawbacks, which are primarily related to requiring a human being to build the representation.

The advantage of using a fully automated approach is that it manages to build representations autonomously. Various ways exist to achieve this, depending on the chosen modeling method, as described in [7], [15], [22], [23]. Because a concept can be either words or phrases, it is possible to apply the following steps when using a vector space approach.

1) Extraction of concepts: First, a text-processing method that is capable of identifying and extracting concepts needs to be chosen. For this task, it is possible to use commonly used text, data, or concept-mining methods. The filtering out of stop words is performed either before or after the extraction of concepts. In order to reduce every word to its basic form, lemmatization methods can be used. Then, a set of natural language processing tools, such as named-entity detection or part-of-speech tagging, can be performed.

2) Rendering of concepts into a vector: After successfully processing every information entity, it is necessary to span a dimension for each entity, which results in the creation of a multi-dimensional vector space. However, if too many dimensions are used, the curse of dimensionality can occur. This can be avoided by applying either extraction or selection methods [11], as they manage to reduce the degree of dimensionality.

3) Interconnection of concepts: Because data become sparser in a high-dimensional space, the interrelation of concepts also becomes more challenging. This means that commonly used techniques, such as distance and proximity measurement, need to be used with caution, as their use does not always yield correct results [2].

These are the steps that a fully automated approach needs to perform in order to build a knowledge representation fully autonomously.

C. Structuring of Knowledge

The last layer is responsible for structuring the represented knowledge in order to build the granular knowledge cube. For this purpose, the following two tasks need to be accomplished:

1) Establishing of a hierarchical structure that consists of multiple levels
2) Building of overlapping granules that permit concepts to belong to two or more granules at the same time

However, granular computing itself is solely a theoretical approach that describes how knowledge should be structured and displayed. It does not suggest or provide any particular algorithm or method for this purpose. As a consequence, it is necessary to find an approach that is capable of fulfilling all of the above tasks.

When choosing a semi-manual approach, the construction of a hierarchical structure can be derived from taxonomical expressions present in the generated knowledge representation. However, the construction of overlapping granules requires the use of an algorithm and cannot be done manually due to the complexity. This is also the reason why a fully manual approach is not viable. Because granules and clusters share similarities, the authors have reviewed several different algorithms that can be deployed for this purpose from the domain of cluster analysis. The most viable candidate has been found to be clique percolation [14] which belongs to the group of graph-based clustering methods.

The fully automated approach, unlike the semi-manual one, relies on an algorithm that is capable of accomplishing both tasks. In addition, the algorithm needs to be able to cope with data that have a medium to high degree of dimensionality. Based on these requirements, a review of clustering methods from the domain of cluster analysis has been performed by the authors in order to find the most suitable algorithm(s).

These requirements have significantly limited the number of potential clustering methods and, with this directly, the number of algorithm(s) that could be deployed. The most viable ones have been found to belong to the group of subspace-clustering methods and self-organizing maps, as they manage to fulfill all of the imposed requirements.

Subspace-clustering methods were developed, particularly to cope with high-dimensional datasets. As such, their primary aim is to detect arbitrarily shaped and positioned clusters that are embedded in various subsets of dimensions. Their use in granular computing is considered a novel approach. Subspace clustering is highly efficient, and the computational complexity is moderate. Overlapping granules are permitted and adaptive learning is present through self-adjusting grid sizes used to measure the density of concepts in an area in order to detect possible granules. The authors consider the hierarchical subspace clustering algorithm (HiSC) [1] that is particularly suited for granular computing, as it allows hierarchical structures to be established while requiring only very little external input to initiate the computation process.

Self-organizing maps per se are not a clustering method but belong to the group of data visualization methods. Nonetheless, in granular computing, they are commonly used as an approach for building and detecting hierarchical, granular structures. In addition, they support the existence of overlapping granules and have a natively built-in mechanism for adaptive learning. The authors particularly consider hierarchical self-organizing maps (HSOM)) [10] and growing hierarchical self-organizing maps (GHSOM) [5] as potent algorithms. Both are hybrids, consisting of self-organizing maps and a part, responsible for applying the hierarchical structure.

Algorithms from both domains can be deployed to build the granular knowledge cube. However, their use should be made dependent of the present degree of dimensionality. In case of a high-dimensional dataset, subspace-clustering methods should be selected over self-organizing maps, which are more suited for datasets with a medium degree of dimensionality.

However, both algorithms rely on a set of indicators that can be used during the hierarchical structuring process in order to determine a concept degree of granulation. The authors suggest for this purpose the following non-concluding list of indicators:

1) Connectivity: The degree of interconnectivity of concepts is a possible indicator of a concept degree of granularity. The more connections are established with other concepts, the more likely it is in an aggregated state. In addition, the degree of nodes can be assessed, which takes into account the number of inbound and outbound connections, if directed graphs are used.

2) Relevance: The more relevant a topic is in a given dataset, the more likely its concepts are promoted to higher-ranked levels in the cube.

3) Actuality: If time is of significance, it can also be used as an indicator to determine a concept's location.

By applying one of the proposed algorithms, the granular, hierarchical structure can be established from a previously generated knowledge representation.

V. Building a Granular Knowledge Cube

In the following section, the process of building a granular knowledge cube is demonstrated with an example, which goes step by step through each of the three layers. Through this, the authors want to improve the described architecture understandability. A fully automated approach is chosen to perform the construction of the cube. The example consists of six different posts, which have been extracted from a blogging platform.

The first step consists of choosing a database that is best suited for storing the phrases as well as graphs that will be created while building the granular knowledge cube. For this purpose, the authors select a hybrid solution that consists of a graph database and document store. Such a hybrid solution is best suited for storing text and graphs efficiently, which is the assessed requirement.

After successfully storing all phrases in the database, the next step consists of building a knowledge representation. For this task, it is first necessary to extract all relevant concepts from the phrases, which are considered to be information entities. This is achieved by applying sophisticated concept-mining methods that are able to identify all relevant concepts. The resulting outcome of this procedure is shown in Fig. 4.

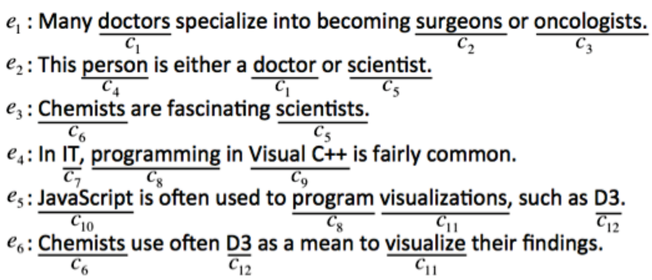


Fig. 4: Concept Extraction

After identifying all relevant concepts from the information entities, it is necessary to position them in a vector space. This is done by first spanning a vector for each entity and then using the vectors to mark the presence of concepts. Through this, it is possible to determine the precise location of each concept in the multi-dimensional vector space. The resulting outcome is illustrated in Fig. 5.

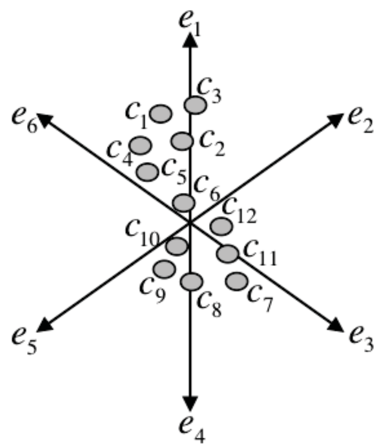


Fig. 5: Concepts in Vector Space

From the obtained multi-dimensional vector space, it is now possible to draw interrelations between concepts. This is done by establishing connections between concepts that have short distances in between. This procedure yields a fully automated knowledge representation.

In a final step, structure is applied to the built knowledge representation. This is achieved by first allocating all concepts to their corresponding levels in a hierarchical structure, taking the previously mentioned indicators into account, and then by placing them into one or several granules. For this task, one of the mentioned algorithms can be used to yield the result illustrated in Fig. 6.

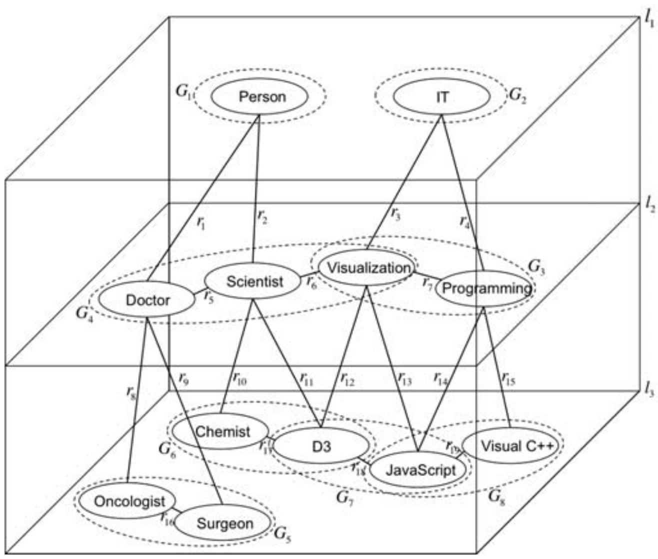


Fig. 6: Sample Granular Knowledge Cube

The final result is a granular knowledge cube, which in this case consists of eight granules, some of which overlap, 19 relationships in between concepts, and three levels. The taxonomical structure is clearly visible alongside intergranular as well as intragranular relationships.

VI. Benefits and Challenges

The benefits are that a granular knowledge cube can be applied in a wide range of different cases due to its open design and flexibility. Restrictions are imposed only in the dataset, which is capable of limiting the possibilities.

Furthermore, any type of data is being supported, be it unstructured, semi-structured or structured, which increases its applicability significantly.

The possibility of coping with uncertainty and vagueness is given due to the fact that granular computing relies on fuzzy logic. Having the possibility of allocating a concept to more than one granule allows for the expression of fuzziness.

Another benefit is that the original context of information entities remains preserved due to the established relationships between concepts, which is a similar approach as the one the human brain uses.

The hierarchical, graph-based design allows for the exploration of the stored knowledge by mapping connections and levels of the granules. Through this, humans can better understand large volumes of information, as they see how information is related and its context, which can lead to the discovery of unexpected findings.

Through granular computing, the tools needed to explore the cube are given, such as zooming in and out, which allow users to see the bigger picture more easily.

At the same time, challenges exist, which are in place due to granular computing failure to offer a clear set of algorithms and methods that can be used for it. Hence, for each layer of the architecture, it is necessary to first find a best-suited approach. This can be a difficult task in some cases.

Another challenge is the creation of a visualization that users can utilize intuitively and that manages to prevent the risk of showing too many concepts at once, as this limits the overall usability.

Last but not least, a granular knowledge cube depends to a high degree on the underlying dataset. The better its quality, the better the resulting output and, with it, the cube usefulness.

VII. Possible Applications

As an enhanced knowledge base, a granular knowledge cube fields of application are manifold, with a clear focus on being part of an artificial intelligence-centered solution. This derives from the intended use, which is not solely to store information but to structure it in such a way that it can be used as something on which to base decisions. This is vital for solving problems and answering questions.

It may also possibly be used as part of an expert system. These systems require a knowledge base in order to solve complex problems and provide solutions, similar to human being's system.

Another possibility is using it in an expert finder system. Unlike in an expert system, the intended output here is not finding a solution but rather making contact with an expert, who is seen as most-suited for giving an answer to a question. The knowledge base can provide vital information on where experts have their domain of knowledge and which ones should be recommended.

If a granular knowledge cube is fitted with a graphical user interface, it can further be used as a monitoring and managing instrument. Through this, administrators have the possibility of discovering stored information and making adjustments.

VIII. Conclusion

In this paper, the concept and architecture behind a granular knowledge cube have been described and, with it, the role as an enhanced knowledge base. Through this, several benefits as well as challenges were identified that occur upon deployment of this construct. Particularly, the ability to cope with different data types, such as unstructured, semi-structured, and structured has been stressed alongside the possibility of supply information on the location, relation, and surrounding knowledge domain of a concept in the cube.

Furthermore, methods and tools have been determined and differentiated that could potentially be used to fulfill each layer purpose. The reason for suggesting more than just one option per layer is derived from the fact that different constraints and environments require different methods and tools. This ensures that the granular knowledge cube can cope with different constraints and environments.

Our future work will aim to validate the proposed concept by building a granular knowledge cube out of a given dataset.

Acknowledgment

This research was supported, in part, by the Commission for Technology and Innovation CTI, Bern, Switzerland under Grant Number 14943.1 PFES-ES.

References

[1] E. Achtert, C. Böhm, H-P. Kriegel, P. Krüger, I. Müller-Gorman, A. Zimek, *Finding Hierarchies of Subspace Clustering*, Proc. 10th Europ. Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD), Germany, pp. 446-453, 2008.

[2] C.C. Aggarwal, A. Hinneburg, D.A. Keim, *On the Surprising Behavior of Distance Metrics in High Dimensional Space*, In: Lecture Notes in Computer Science, Volume 1973, pp 420-434, 2001

[3] R. Angles, C. Gutierrez, *Survey of graph databases*, ACM Computing Surveys (CSUR) Volume 40, Issue 1, Article 1, pp.1-39, 2008.

[4] D.P. Ausubel, J. Novak, H. Hanesian,*Educational Psychology: A Cognitive View*, 2nd Edition, Rinehart & Winston, New York, pp. 251-257, 1978.

[5] A. Chan, E. Pampalk, *Growing hierarchical self organizing map (GHSOM) toolbox: visualizations and enhancements*, In: Neural Information Processing, 2002. ICONIP '02. Proceedings of the 9th International Conference, Volume 5, pp. 2537-2541, 2002

[6] A.M. Collins, M.R. Quillian, *Retrieval time from semantic memory*, Journal of Verbal Learning and Verbal Behavior, volume 8, pp. 240-248, (1969).

[7] X.L. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, W. Zhang, *Knowledge Vault: A Web-Scale Approach to Probabilistic Knowledge Fusion*, In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 601-610, 2014.

[8] J. Hey, *The Data, Information, Knowledge, Wisdom Chain: The Metaphorical Link*, pp. 4-8, 2004.

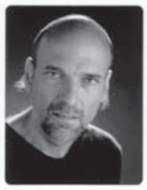
[9] S. Jouili, V. Vansteenberghe, *An Empirical comparison of graph databases*, IEEE, Social Computing (SOcialCom), pp. 708-715, 2013.

[10] J. Lampinen, E. Oja, *Clustering Properties of Hierarchical Self-Organizing Maps*,Journal of Mathematical Imaging and Vision, pp.261-272, 1992.

[11] H. Liu, H. Motoda, *Feature Extraction, Construction and Selection: A Data Mining Perspective*, Kluwer Academic Group, USA, 1998.
[12] M. Minsky: *A Framework for Representing Knowledge*, MIT-AI Laboratory Memo 306, 1974.
[13] M. Minsky, *The Emotion Machine: Commonsense Thinking, Artificial Intelligence, and the Future of the Human Mind*, Simon & Schuster, Inc., 2006.
[14] G. Palla, I. Derényi, I. Farkas, T. Vicsek, *Uncovering the overlapping community structure of complex networks in nature and society*, Nature 435, pp. 814-818, 2005.
[15] S. Puri, *A Fuzzy Similarity Based Concept Mining Model for Text Classification*, International Journal of Advanced Computer Science and Applications, Vol. 2, No. 11, 2011.
[16] R. A. Quilian, *A notation for representing conceptual information: An application to semantics and mechanical English paraphrasing*, SP-1395, System Development Corporation, Santa Monica, 1963.
[17] M.R. Quilian, *Semantic memory*, In: Semantic Information Processing (Ed. M. Minsky), Cambridge, MA: MIT Press, pp. 227-270, 1975.
[18] A-B. M. Salem, M. Alfonse, *Ontology versus Semantic Networks for Medical Knowledge Representation*, 12th WSEAS International Conference on Computers, pp.768-774, 2008.
[19] U. Sattler, D. Calvanese, R. Molitor, *Relationships with other formalisms, Description of logic handbook*, pp. 137-177, 2003.
[20] B. Shao, H. Wang, Y. Xiao, *Managing and mining large graphs: Systems and implementations*, in proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, SIGMOD 12, pp. 589-592, New York, USA, 2012.
[21] J. F. Sowa, *Conceptual Graphs for a Data Base Interface*, IBM Journal of Research and Development 20 (4), pp. 336-357, 1976.
[22] G Weikum, M. Theobald, *From information to knowledge: harvesting entities and relationships from web sources*, In: Proceedings of the twenty-ninth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, ACM, pp. 65-76, 2010.
[23] W. Wu, H. Li, H. Wang, K.Q. Zhu. *Probase: A probabilistic taxonomy for text understanding*, SIGMOD, ACM, pp. 481-492, 2012.
[24] W3C, *OWL Web Ontology Language Overview*, 2004.
[25] W3C, *Resource Description Framework (RDF) Schema*, 1998.
[26] Y.Y. Yao, B. Zhou, *A logic language of granular computing*, Proceedings of the 6th IEEE International Conference on Cognitive Informatics, IEEE Press, pp. 178-185, 2007.
[27] Y.Y. Yao, *The Art of granular computing*, Proceedings of the International Conference on Rough Sets and Emerging Intelligent Systems Paradigms, LNAI 4585, Springer, pp. 101-112, 2007.
[28] L. Zadeh, *Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic*, Fuzzy Sets and Systems, Volume 19, pp. 111-127, 1997.

Alex Denzler is currently obtaining a Ph.D. in Information and Communication Technology from the University of Fribourg, Switzerland. His previous studies include a Bachelor Degree in Management and Economics and a Master in Information and Communication Technology from the University of Fribourg, Switzerland. His research is focusing on Fuzziness, Expert Finder Systems, Recommender Systems, Profile Matching and Clustering.

Marcel Wehrle is currently obtaining a Ph.D. in Information and Communication Technology from the University of Fribourg, Switzerland. He received a B.A. in Communication Science and Economics, a Premaster in Informatics and a M.A. in Communication Science from the University of Fribourg, Switzerland. His research is focusing on Granular Computing, Information Granulation, Self Organizing Maps.



Andreas Meier is a member of the Faculty of Economics and Social Science and a professor for information technology at the University of Fribourg. He specializes in electronic business, electronic government, and information management. He is member of GI (Gesellschaft für Informatik), IEEE Computer Society, and ACM. After studying music in Vienna, he graduated with a degree in mathematics at the Federal Institute of Technology (ETH) in Zurich, studied his doctorate, and qualified as a university lecture at the Institute of Computer Science. He was a systems engineer at the IBM research lab in San Jose, California, director of an international bank, and a member of the executive board of an insurance company.