

Supporting Taxonomy Management and Evolution in a Web-based Knowledge Management System

Binh Vu
Research Institute for
Telecommunication
and Cooperation
Dortmund, Germany
bvuv@ftk.de

Jana Mertens
Research Institute for
Telecommunication
and Cooperation
Dortmund, Germany
jmertens@ftk.de

Kerstin
Gaisbachgrabner
Graz University of
Technology
Graz, Austria
kerstin.gaisbachgrabner
@tugraz.at

Michael Fuchs
Wilhelm Büchner
University of Applied
Sciences
Pfungstadt, Germany
michael.fuchs@
wb-fernstudium.de

Matthias Hemmje
FernUniversität in
Hagen
Hagen, Germany
matthias.hemmje@
fernuni-hagen.de

The internet surpassed the zettabyte order of magnitude of traffic in 2016. With the growth of the internet, information categorization becomes even more important than before. By matching information to existing categories, humans as well as machines are able to organize, manage, access and re-use information and knowledge resources in a more efficient and effective way. Categorization also helps to narrow the choices among content, information, and knowledge resources. Content, information, and knowledge resources can be browsed, searched, and accessed faster if they are appropriately categorized. In order to make categorization possible, a taxonomy is needed to organize content, information, and knowledge resources into a set of topical categories. In this paper, we target the need to support human computer interaction related to applying taxonomies including the challenges in taxonomy management, development, and evolution, as well as approaches to solve the related challenges. A Taxonomy Management System is introduced in this paper as a solution for supporting the human computer interaction in management, development as well as the evolution of taxonomies. The underlying architecture as well as the user interface of the Taxonomy Management System will also be explained. Finally, the implemented proof-of-concept system prototype is evaluated with real world users along real-world application use cases within the scope of a European funded Research and Innovation Action project.

Taxonomy, taxonomy Management, Taxonomy Management System, Taxonomy Manager, Content and Knowledge Management, Ecosystem Portal, Delphi method, crowd sourcing, crowd voting, version control

1. INTRODUCTION AND MOTIVATION

Today's internet is a big source of content, information and knowledge resources. Every day there is a huge amount of data produced and used on the internet. Most companies in the U.S. today have at least 100 TB of data stored. It is estimated that by 2020, 40 Zettabytes (43 trillion Gigabytes) will be created, an increase of 300 times from 2005 (The Four V's of Big Data, 2005). Data not only needs to be indexed but also the index terms and synonyms need to be uniform. Otherwise an indexer would have classified documents that are about the same topic into various categories, despite the fact that these categories have the same meaning. It would make searching, accessing and comparing afterwards more difficult. In such cases, taxonomies can be used as controlled vocabularies. Their textual labels, i.e., vocabulary terms, can be used as a list or even hierarchy of agreed-on terms which are later used for indexing or cataloging documents.

Furthermore, with the support of taxonomies, classification consistency can be achieved (Hedden, 2010).

Information overload continues to be a challenge. In the corporate world, knowledge workers spend more than 11 hours a week searching for and analyzing information (Whittaker & Breininger, 2008). By dividing the material into many different small subsets, taxonomy support can make content, information, and knowledge resource retrieval and access faster and more accurate. Instead of having to know the exact textual labels, i.e., keywords that describe relevant content, information or knowledge resources, users can interactively browse and search for documents by selecting the taxonomy categories that are relevant for their information need. However, after each relevant category is selected, the list of relevant results returned will have to be reduced to a size that is small enough for users to be perceived and inspected in detail.

Taxonomies are also needed to support classification during navigation. Different from applications in information retrieval, another property of taxonomies is that they can support navigation, i.e., users in finding their way around the functions and features of a system's user interface. Websites can, e.g., use taxonomies as a table of contents to guide users through their relevant informational or also functional topics respectively features. Users then use the taxonomy to navigate in the website and, e.g., have a better understanding over the organization of the enterprise's information offering or of the underlying information system (Hedden, 2010).

2. PROBLEM STATEMENT AND APPROACH

"As much as taxonomies can be powerful enablers of sharing, coordination and common identity, so they can also fragment, sow discord, alienate, enforce violence and even destroy" (Lambe, 2007). This means that there are some inherent challenges in the human computer interaction supporting the management, development, and evolution of taxonomies.

Developing a taxonomy usually involves many people, such as IT staff, corporate librarians, departmental publishers, ... (Kon & Hoey, 2005). However, the more people are working together on a taxonomy management, development and evolution, the more problems this collaboration can potentially generate. Not only will it take more time to communicate and agree on decisions about taxonomy management, development, and evolution but group members of such a taxonomy collaboration group, e.g., tend to agree on majority views in order to keep workplace relationships, even when the majority has wrong judgments (Asch, 1956). On other hand, working alone can get users surrounded by information and knowledge that only supports their own individual and very subjective point of view and forgets other views as alternatives. Not to mention that it is a lot of work for only a few people to build a complete taxonomy. Therefore, easy to use, effective, and efficient collaboration tools are needed to support the collaborative working task of managing, developing, and evolution of taxonomies.

Of course, knowledge resource and more general information needs as well as knowledge and information representation always change. Therefore, to reflect the changing needs for content, information, and knowledge resources, taxonomies need to be maintained frequently. Without maintenance and governance, requiring special features to manage version and ownership, taxonomies can be drifting away from current application, i.e., e.g. business and organizational needs (Lambe, 2007).

A taxonomy normally has a lot of nodes. For example, the ACM 2012 Computing Classification System (The 2012 ACM Computing Classification System, n.d.) contains about 2500 nodes in a hierarchical tree structure representation and its corresponding serialization file format. Supporting the work on a big tree of potentially thousands of nodes by any kind of machine readable representation and processing needs a lot of ICT resources, such as compute time, memory, and disk space. If this multiplied by the fact that hundreds or even thousands of users use the system at the same time, one ends up with even higher numbers of resources needed. Besides this, taxonomies which usually have big sizes and high complexity will become challenging in terms of computation but also in terms of human computer interaction. This means, any solution has to consider how to organize the taxonomy in the underlying database as well as the user interface in such a way that it requires less space and is supporting retrieval, access, and management operations sufficiently fast. Furthermore, if a taxonomy is stored in a relational database (such as, e.g., MySQL, MSSQL etc.) as hierarchical data, the resulting tables are simply a flat list. This means, that the hierarchical structure with parent-child relationships is not naturally representable in a relational database model. As there are different models to represents hierarchical data in relational database, suitable model for the as well as a suitable user interface for Taxonomy Management System is needed.

Finally, supporting the managing, developing and evolution of taxonomies can create all sorts cognitive as well as human computer interaction problems, especially if the chosen user interface design and implementation approach is too simple to allow the taxonomy to be compared to its informational context within a domain specific application environment. Therefore, not only testing the taxonomy regarding its classification and informational properties is of importance but also the degree to which a Taxonomy Management System supports its collaborative actors effectively and provides its functions and features efficiently from a cognitive point of view is an important aspect in the development of appropriate support mechanism.

2.1 Realising and Applied Gaming Ecosystem

Realising and Applied Gaming Ecosystem (RAGE) is a real-world application scenario that is utilized in the remainder of this paper to experimentally develop and apply a Taxonomy Management Support and for an exemplary evaluation of the methods and tools that are introduced in this paper. RAGE is a 48-months Research and Innovation Action project co-funded by EU Framework Program for Research and Innovation, Horizon 2020 (Salman, Heutelbeck, & Hemmje, 2015). The main objectives of RAGE are to allow its participants to

get hold to advanced and usable applied gaming assets, to get access to the associated business cases, and to create bonds with peers, suppliers and customers for the purpose of advocating their expertise and demands. Furthermore, the project will help participants to develop and publish their own assets and to contribute to creating a joint agenda and road-map (Salman, Heutelbeck, & Hemmje, 2015).

Beside using the existing ACM Taxonomy, the project needs to develop a taxonomy for applied gaming asset and corresponding content and knowledge resource classification. This taxonomy will help the process of enriching and transforming advanced gaming technologies into self-contained assets for applied gaming that facilitate essential pedagogical functions, that can be linked together into higher level aggregates and that can be easily integrated in existing game platforms (Georgiev, et al., 2016). In order to fulfil this target, the asset developers on the project will need a tool that support Taxonomy construction, management as well as collaborative evolution.

3. STATE OF THE ART

3.1 Knowledge and Content Management

Knowledge Management (KM), like most complex things, has many different definitions. Depending on the nature of the scientific area, the definition of KM might have different meaning. Nevertheless, what Devenport and Prusak wrote in their book "Working Knowledge: How Organizations Manage What They Know" was agreed and cited the most: "Knowledge management draws from existing resources that your organization may already have in place good information systems management, organizational change management, and human resources management practices. If you've got a good library, a textual database system, or even effective education programs, your company is probably already doing something that might be called knowledge management" (Davenport & Prusak, 1998).

Content Management (CM) is usually mentioned along with KM. It is about gaining control over the creation and distribution of information and functionality. CM is also about knowing what value you have to offer, who wants what parts of that value, and how they want you to deliver it. From different perspective, content management either distributes business value; balances organizational forces; combines content-related disciplines; collects, manages, and publishes information or is a technical infrastructure. CM is and does all these things (Boiko, 2005).

3.2. Taxonomy

The word taxonomy comes from two Greek stems "taxis" and "nomos". "Taxis", broadly, means the arrangement or ordering of things (Lambe, 2007) and "nomos" means law or science (Hedden, 2010). The term taxonomy means in general "the rules or conventions of order or arrangement" (Lambe, 2007). In the dictionary, taxonomy is defined as an "orderly classification of plants and animals according to their presumed natural relationships" (Definition of Taxonomy, 2017). Or as in term of computer science, taxonomy is "a hierarchical representation of categories, provides a navigation structure for exploring and understanding the underlying corpus without sifting through a huge volume of documents" (Abrol, et al., 2005). Because of the usual hierarchical nature, a taxonomy imposes a topical structure on information (Whittaker & Breininger, 2008).

Whittaker et al. defined taxonomy as a controlled vocabulary in which each term has hierarchical and equivalent relationship. Hierarchical relationship means each term has broader and narrower terms. For example, the term "cellphone" has broader term "telephone" and narrower term "smartphone". The user can move up or down on the hierarchy to retrieve more or less specific information. Equivalent relationship means each term has it synonyms (Whittaker & Breininger, 2008). For example, "cellphone" with "mobile" and "mobile phone".

For Lambe, taxonomy is (1) a classification scheme, (2) a sematic domain representation, and (3) a knowledge map. Classification schemes are designed to put related things to the same group. If you find one thing in a group, you can find other things related to it in the same group. Furthermore, taxonomies do not reply on codes but semantic representation. That means, a fixed vocabulary is provided to describe knowledge resources and other information assets, like e.g., content objects. For example: everything relates to "car" should be labelled "car". A taxonomy therefore provides a controlled vocabulary. Each term is carefully considered. They are only admitted to the taxonomy when they clearly describe a commonly understood category of content. A taxonomy is also a semantic representation in the way that it expresses the relationship between terms in the taxonomy (Lambe, 2007).

3.3. Crowdsourcing and Crowdvoting

From 2006 to 2011, there are 40 definitions for the concepts of Crowdsourcing that come from 32 distinct articles (Estellés-Arolas & González-Ladrón-de-Guevara, 2012). Although there are many definitions, crowdsourcing can be understood as outsourcing tasks to many people, rather than giving the tasks to only an in-house employee or contractor (Alonso & Lease, 2011). Crowdsourcing has some subcategories, such as crowdfunding where funding

was raised by a lot of people and each of them contributes a small amount. Crowdvoting describes a concept where, e.g., a website's users vote on a certain topic (Kitchens & Crane, 2014).

3.4. Taxonomy Management

Whittaker and Breininger suggest five steps in the work of developing a taxonomy: (1) determine requirements, (2) identify concepts within the taxonomy, (3) develop a draft taxonomy, (4) review draft taxonomy and (5) refine taxonomy (Whittaker & Breininger, 2008). The review is, e.g., done by getting feedback from experts, e.g., using the Delphi method (Linstone & Turoff, 1975). At this step, a questionnaire is designed and sent to a group of experts. After the questionnaire is returned, the results are summarized and, based upon the results, the taxonomy is being refined in step 5. After the change, the refined taxonomy is getting reviewed again. A new questionnaire is developed. The expert group is usually given at least one opportunity to reevaluate its original answers based upon examination of the group response.

The traditional method of using experts for reviewing, while providing many benefits and advantages, has its own problems. One example is that experts are not always available. They have other jobs to do and rounds of reviewing takes too much time from their schedule. Another example is that people tend to ignore disagreements. The result is, e.g., a poor design decision which is ignored during reevaluation and not getting fixed.

From the lessons taught by Wikipedia and Open Source Software, the power of crowdsourcing can be learned. This means, that the overall idea, of e.g., the RAGE project's crowd-sourced Knowledge Management Ecosystem Portal (KM-EP) is a combination of Wikipedia (Wikipedia, n.d.), SlideShare (SlideShare, n.d.), Mendeley (Mendeley, n.d.), GitHub (GitHub, n.d.) and StackOverflow (StackOverflow, n.d.) where users can submit their software assets as well as related contents, information, and knowledge resources and experts can choose which asset to integrate to their system development through a collaborative and co-creative innovation process. As James Surowiecki wrote in his book "under the right circumstances, groups are remarkably intelligent, and are often smarter than the smartest people in them" (Surowiecki, 2005). This is not the first example of using the wisdom of the crowd as we can find the other examples of crowd sourcing and crowd voting everywhere. One example is Threadless.com, a web-based t-shirt company, where user can submit and vote for t-shirt design (Brabham, 2008).

3.5. Content and Knowledge Management Ecosystem Portal

The RAGE KM-EP (Vu, 2015), has been developed to provide powerful tools for managing knowledge and content. The initial version was developed on the basis of Typo3, an open source content management system (Typo3, n.d.). Since then, the KM-EP was constantly upgraded with new features and functionalities. There are scientific research and technical development works in the area of Social Media Platform Integration (Salman, Heutelbeck, & Hemmje, 2015), Course Authoring (Then, et al., Innovative Authoring Tools for Online-Courses with Assignments - Integrating Heterogeneous Tools of e-Learning Platforms into Hybrid Application Solutions, 2016) and Competence-Based Learning (Then, et al., A Competence-Based Course Authoring Concept for Learning Platforms with Legacy Assignment Tools, 2016). Currently, the KM-EP tool suite has 10 components in 5 subsystems: BigData Analysis, Management and Access support in the Information Retrieval Subsystem; Scientific Training Online and Competence Manager in the Learning Management Subsystem (LMS); Ingest, Export, Content Manager, and Taxonomy Manager in the Content and Knowledge Management Subsystem (CKMS); Authentication and User Manager in the User Management Subsystem (UMS); and Storage Manager in the Storage Management Subsystem. CKMS and LMS act as a central repository for documents, media and courses, while UMS are single points of entry for managing identities and communities (Vu, 2015). Figure 1 presents the KM-EP's architecture.

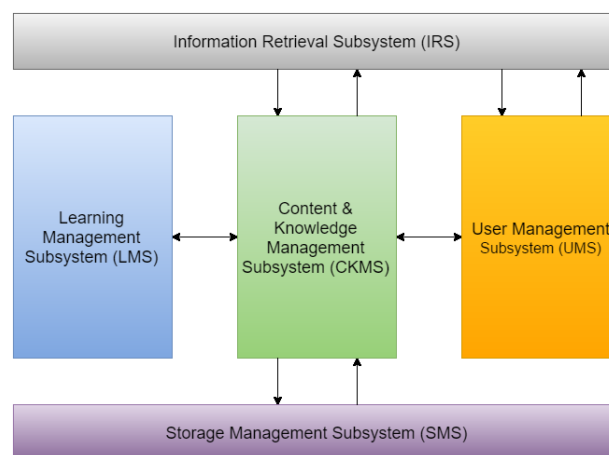


Figure 1: KM-EP architecture (Vu, 2015)

With the growth of the KM-EP tool suite, developers needed more support and freedom from the development framework. Meanwhile, Typo3 was designed only for content management, so it is difficult to extend it to serve other purposes. Furthermore, Typo3's small community and lack of documentation creates a high learning curve. It is difficult to have technical questions answered since there are not so many Typo3 expert out there. Therefore, the KM-EP's development framework was recently switched from Typo3 to Symfony

(Symfony Framework, n.d.). The Symfony Framework is currently one of the leading PHP frameworks supporting the creation of web applications. It has a large community, many reusable components and high-quality documentation (What is Symfony, n.d.). All the factors mentioned above made Symfony the best option to replace Typo3. After the upgrade, the KM-EP tool suite now is available in version 2.

3.6. Simple Knowledge Organization System

The Simple Knowledge Organisation System (SKOS) is a formal language for representing controlled structured vocabularies such as thesauri or classification schemes (Miles, 2006). SKOS is recommended by W3C as a standard for expressing knowledge organization systems (KOS) such as thesauri or taxonomies (Nagy, Pellegrini, & Mader, 2011). It is widely used by organizations including NASA, UN and the New York Times (SKOS Datasets, 2015).

3.7. Remaining Challenges

There are some challenges that need be considered in the development of the KM-EP's Taxonomy Manager.

- If the Taxonomy Manager is designed in a user centered way, so that normal, i.e., naïve or non-expert users can use it easily, there cannot be many features. This means experts will not have full control over the system and cannot produce complex taxonomies. On other hand, if the Taxonomy Manager supports many complex features for experts, normal, i.e., naïve or non-expert users cannot use it easily.
- When more than one user is working on the same taxonomy at the same time, we need a mechanism to detect this and notify users.
- There are different formats used to store taxonomy, such as OWL, RDF, Turtle, etc. The Taxonomy Manager need to support these formats in both import and export process.

4. CONCEPTUAL MODELLING

The goal that is targeted in the remainder of this paper is to build a user-centered Taxonomy Management System, which lets the crowd create and modify their own taxonomies in a very easy to use but still effective and efficient way. With support of crowd sourcing, not only the experts or administrators can build taxonomies for the KM-EP, but everyone can join and build their own taxonomy now. Ideally such producers and consumers (prosumers) (Kotler, 1986) of taxonomies later will be the people that vote for the best taxonomies to

become the next root taxonomies. The KM-EP uses root taxonomies to classify contents. Branching and merging features allow to support multiple streams of work independent of each other. In this way it avoids conflicts when different persons work on the same taxonomy. And later their works can be merged back to the main line and become the next version of a root taxonomy.

With support of version control, the changing and change management of a taxonomy (Taxonomy Evolution) will be faster, more efficient and agile. It is a great way to keep track of taxonomy builds by being able to identify which version is currently in development, what are the changes etc. A complete long-term history of changes of every taxonomy will ideally support such Taxonomy evolution. In such a support, users can ideally compare different versions of a taxonomies together, see which parts were changed etc. They should also be supported to rollback to any earlier version at will. This is a crucial feature for supporting debugging processes which always happen in the development of taxonomies, and which will help to easily fix any mistake.

Furthermore, the evolution of the Taxonomy will not affect the classification of resources. By using persistent identification methods, the terms in the taxonomy are always kept unique. Even in the case when a term's content is changed completely, the classification of contents using the term will not be lost. Fast and efficient algorithms will speed up the managing process and require less resources. Combining the taxonomy management with caching mechanisms, thousands of terms can be retrieved in matter of milliseconds.

The Taxonomy Manager manages knowledge. Therefore, it is a part of the Content and Knowledge Management Subsystem of the KM-EP. To achieve its goals, the Taxonomy Manager is separated into 3 modules: Taxonomy Editor, Version Control and Categorization. Figure 2 presents the overall architecture of the Content and Knowledge Management System in which the Taxonomy Manager is a part of.

The Taxonomy Manager was designed and implemented to initially support the above introduced project RAGE. As stated in the definition, a taxonomy consists of terms. In a SKOS ontology, a term can have properties of their own as well as relationship with each other. Figure 3 describes how a taxonomy is modelled, not just with SKOS, but with other ontologies as well. The class "Taxonomy" describes a taxonomy with its basic properties, such as title, description, authors, etc. A taxonomy has at least one term. The class "Term" describes a term of a taxonomy. A term, depending on the current ontology, can have properties, e.g. "skos:prefLabel", which describes the preferred label of this term in SKOS. It also can have a relationship with other terms. The class "Relation" describes the relation

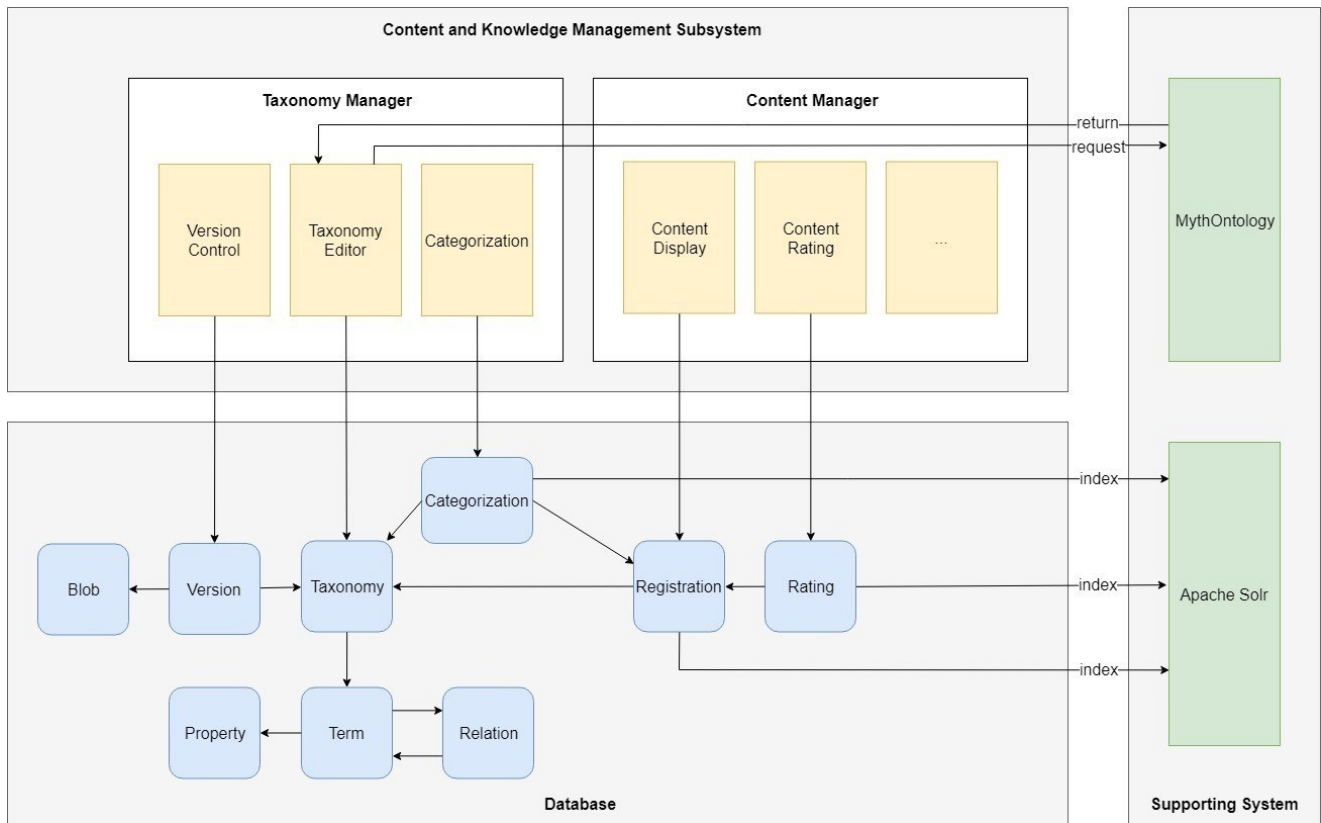


Figure 2: Architecture of the Content and Knowledge Management Subsystem

between terms in this taxonomy. Field “name” is used for storing the name of the relationship, e.g. “skos:broader” which describes term “A” as a broader term than term “B”.

The Version Control module takes a snapshot of a taxonomy and stores its metadata into a table “Version” and its data into table “Blob”. However, not only the Taxonomy Management can use the version control, but any component of the KM-EP can access it and take advantage of it.

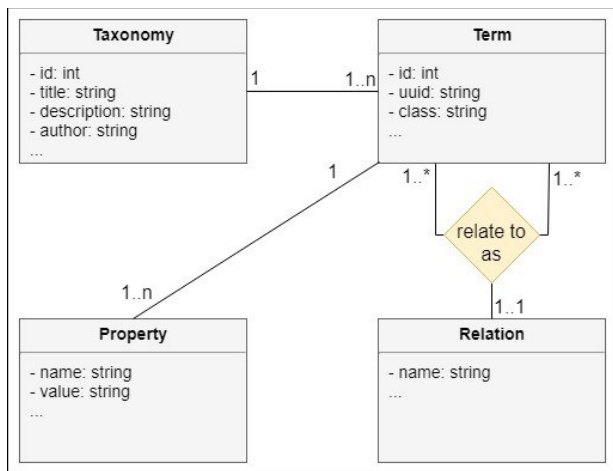


Figure 3: The Representation Model of a Taxonomy

The table “Categorization” of the module Categorization stores the ID of the content which was categorized, along with the taxonomy which the

content was categorized in. The categorization is indexed by an Apache Solr (Apache Solr, n.d.) server and later can be retrieved using the KM-EP’s Information Retrieval Subsystem. Solr is an open source search platform, which is highly scalable and fault tolerant. It enables many powerful matching capabilities which helps searching for contents more effective and efficient (Solr Features, n.d.). The indexed categorizations are used for supporting faceted search in the future which is an important feature and is used by many big websites, such as Amazon, eBay, Google, ... to retrieve information. Faceted search helps to address weaknesses of conventional search approaches and provides more effective information-seeking support to users (Tunkelang, 2009).

5. IMPLEMENTATION APPROACH AND PROTOTYPE

The Taxonomy Manager, like the KM-EP itself, was implemented in PHP using the Symfony Framework version 3.3, later was updated to version 3.4. The relational database system MySQL (MySQL, n.d.) is used to store taxonomy metadata, properties, relations as well as version metadata and snapshot’s data. On the frontend, the taxonomy tree is built with the help of JavaScript library JsTree (jsTree, n.d.) and jQuery (jQuery, n.d.). The JavaScript’s framework AngularJS (AngularJS, n.d.) is used to make a seamlessly experience for the

user. Data is sent to backend using Ajax calls (Ajax, n.d.), so users can interactively modify data without reloading the page.

The first module of the Taxonomy Manager is the Taxonomy Editor. It was implemented based on the models in Figure 3. The component lets users create and modify their own taxonomy using SKOS. The created taxonomy is represented as a tree. Each term is a node. When users click on a term, information of this term will be displayed on a panel. Users then can choose to add, edit or delete properties as well as relations of the term. Figure 4 shows the Taxonomy Editor's user interface.

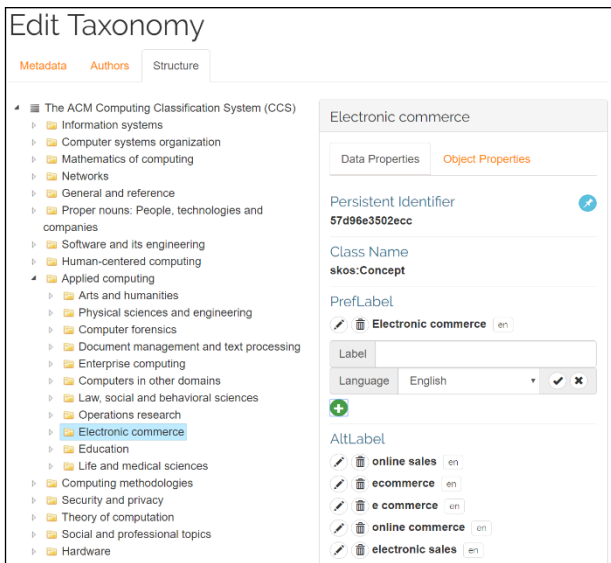


Figure 4: User interface of the Taxonomy Editor

The taxonomy is read from the database and built into a tree. The tree then will be cached using the distributed memory object caching system MemCached (Memcached, n.d.) to reduce the time and resource needed from the server. Next time, users view the taxonomy, the data will be read directly from the cache. To calculate the performance gain of using MemCached, a test was performed. The 2012 ACM Computing Classification System was imported into the Taxonomy Editor. The ACM taxonomy tree with 2300 nodes then was loaded and constructed several times from both database and cache. The result shows the construction time reduces nearly 20 times from average of 0.13 second with database to average of 0.069 second with MemCached.

To extend the ability of the Taxonomy Editor, an ontology server was implemented. The server, called "MythOntology", is written in Java and being used for processing ontology files. The file content will be read and send to MythOntology server along with the configuration. The server will read the content and send the processed taxonomy back to the Taxonomy Editor. In its export process, the Taxonomy Editor sends the taxonomy along with the

configuration, such as file name, format, ... to the MythOntology server. The server will process the data and send the content of the export file back to the Taxonomy Editor. The Editor then lets users download the file through a web browser. The MythOntology server can work with many ontology formats, such as RDF/XML, OWL2, LaTeX and Turtle.

The second main module of the Taxonomy Manager is the Version Control. Similar to the distributed version control system Git (Git, n.d.), when users decide to save (commit) the current state of a taxonomy (take a snapshot), the hash of the data of each term is first calculated. This hash then is compared with the hash of the same term but at last version. If the two hashes are the same, that means this term was not changed since the last version. Therefore, there is no need to save the data again. Only the hash in this case is saved. If two hashes are different, it means this term has changed. In this case, the term's data and its hash are saved with the snapshot. Users rarely change the whole taxonomy in one version, therefore this commit mechanism uses much less space than compared to saving the whole taxonomy. Figure 5 presents data storing as snapshots over time.

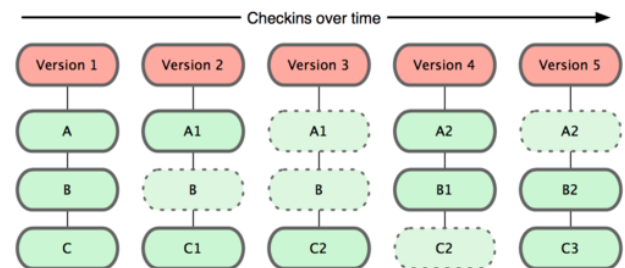


Figure 5: Snapshots over time (Git Basics, n.d.)

Each taxonomy has its own history. User can review this history to see the list of snapshots, when it was taken, and the changes made each snapshot. If there is a mistake or if users just want to reset the taxonomy to a certain version, there is a "Rollback" feature available. With this feature, a snapshot data of the taxonomy is read, reconstructed, and then replaced the current taxonomy. The Version Control component reads every node, that belongs to the snapshot of the version, which users want to rollback to. Data of the node is retrieved and used to reconstruct the node. If a node was not changed since previous versions, the script reads the flag "last_change" to detect when it was changed for the last time and goes there to retrieve the data.

The Delphi method was replaced by crowd sourcing and crowd voting, where users have the ability to vote for each taxonomy using Content Rating module. The highest rated taxonomy will later be merged back to the its base and become the next official version. Each taxonomy has a unique identifier. The system uses this id to detect cloned

versions of a base taxonomy and to maintain the persistence of the classification during the evolution of a taxonomy.

Categorization is the third module of the Taxonomy Manager. When users edit their contents, the list of the KM-EP's taxonomies is presented on the left side of the page. After user can selected a taxonomy, the taxonomy structure with terms in tree format is loaded. Users can then select terms which are related to the current content and can click "Assign" to save these relationships. These assignments are indexed with content metadata by Apache Solr as presented in Figure 2. The indexed data is later retrieved when users search for contents using the Information Retrieval Subsystem.

6. EVALUATION

The Taxonomy Manager was deployed in several projects as part of the KM-EP. In the project RAGE, the Taxonomy Manager was then taken a step further by organizing an evaluation. The goal of this evaluation was to evaluate the functionality and user interface of the management system itself – i.e. gathering feedback on initial features and functionality. With the gathered answers and results from the evaluation, questions like "Is the taxonomy manager useful as it is? Does the tutorial help to work more efficient? Is the taxonomy manager mature or are there any improvements to be taken?" should be answered by interpreting the main results.

6.1. Target Participants

The Taxonomy Manager was analyzed, like the KM-EP itself, in detail and with different user groups. The involvement of stakeholders is considered critical for the success of KM-EP, and therefore for the Taxonomy Manager. Users' attitudes and the actual usage of a system like the Taxonomy Manager should be evaluated to be attractive to relevant stakeholder groups. To make sure that the evaluation covers both, internal and external opinions and ratings, the target participants consisted of two groups: "internal" participants (from the consortium) and "external" participants (game developers or target group with similar focus, students-potential users). 10 members from different consortium partner institutions and 10 external participants took part in this evaluation of the Taxonomy Manager and provided their feedback.

6.2. Evaluation Instruments

As evaluation techniques a combination of questions/items from standardized questionnaires UMUX (Finstad, 2010), USE (Lund, 2001), Münsteraner Fragebogen zur Evaluation – Zusatzmodul Diskussion (Thielsch & Stegemöller,

2014) and questionnaire items used by Papadakis, Andreou, and Chrissikopoulos (Papadakis, Andreou, & Chrissikopoulos, 2012) and open questions concerning the functionality of some features, were used. The survey largely covered questions to be answered on a 7-point-likert-scale from strongly disagree (1) to strongly agree (7) and contained the category "no answer" as well. In the end of the survey three open questions concerning the improvement of the taxonomy manager and the tutorial were asked as well. Overall, there were seven different kind of evaluation categories: Usability, Usefulness, User Interface, Tutorial Quality, Experience with Taxonomies/ Taxonomy Management Systems, Taxonomy Manager, Version Control, Export/Import Features and improvements concerning the Taxonomy Manager.

6.3. Procedure

Teams from Work Package 6 (WP6) Ecosystem Development and Work Package 8 (WP8) Evaluation of the project RAGE were working together to create access to the Taxonomy Manager and to develop the evaluation questionnaire. Members of the consortium and external game developers as well were contacted by WP8 members to participate in the evaluation of the taxonomy manager. To prepare the participants for the evaluation, a tutorial for the Taxonomy Manager, prepared by members of WP6, was sent out. To get familiar with the Taxonomy Manager a specific task was prepared, to make sure that all the participants work with the Taxonomy Manager in a comparable way in a comparable timeframe. Processing the task took approximately around half an hour (up to 45 minutes for unexperienced users), including time to go through the provided tutorial. The survey itself was carried out via Lime Survey and took about 15 minutes.

6.4. Results

As mentioned in the section above, all statistical analyses were conducted using the software IBM SPSS statistics version 23 (IBM Corp., NY, USA). The general level of significance was fixed at $p < .05$ (two tailed) for all analyses. Figure 6 provides an overview of the results obtained from the evaluation categories (subtests). The usability of the Taxonomy Manager was in general assessed as rather moderate to good ($M=3.92$; $SD=0.60$), indicating that there is still room for improvements, like the results from the open question concerning personal feedback showed. Usefulness was assessed as also rather moderate to good ($M=3.57$; $SD=1.20$) but based on the quite high standard deviation it must be assumed that the rating was quite diversely. The rating concerning the user interface was assessed as good ($M=4.19$; $SD=1.13$), but again with a high standard deviation which allows the same

conclusion as before. The quality of the tutorial was evaluated the worst, with a mean of 3.52 and a standard deviation of 0.58. That suggested that the tutorial need a revision. The evaluation of the participants' experience showed that all participants had a rather solid experience/knowledge about taxonomies in general (M=5.07; SD=1.02). The experience with the Taxonomy Manager or a similar system was rated moderately in general (M=3.60; SD=0.54). Version control (M=4.36; SD=1.12) and the export/import features (M=4.43; SD=0.97) as well were perceived as moderate to good.

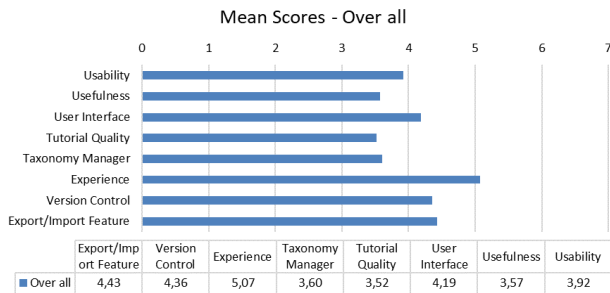


Figure 6: Mean scores of all evaluation categories

7. SUMMARY AND FUTURE WORK

In this paper, we have described the concept of taxonomy as well as content and knowledge management, development and evolution. We identified the importance of taxonomy management, development, evolution and corresponding categorization and where and why it is needed. We presented the problems and requirements in the development of taxonomy management, development, and evolution support and our approaches to solve them.

Our research has contributed to solving the problems of taxonomy development, evolution, and management by combining crowd sourcing, crowd voting and version control. In result, the KM-EP in the context of the RAGE application was introduced. The Taxonomy Manager was implemented as a component of the KM-EP to support users to develop and manage taxonomies during their evolution. The tool enables users to keep track of the development process. Users have full power to browse, compare, merge or rollback to taxonomy snapshots. Users being prosumers of taxonomies themselves will later also be the people that vote for the best taxonomies to become the next base taxonomies.

Finally, an evaluation was conducted in the scope of the application context of the project RAGE to test if the Taxonomy Manager fulfils all the requirements and how well it performs in the different evaluation dimensions related to Human Computer Interaction. The result proved the importance of a taxonomy management system. The scores show the system

has a good performance, user interface, and functionality. In general, the participants appreciate the Taxonomy Manager. But there are some improvements to make too.

Due to time limitation, only SKOS is being used in the current implementation. In general, any ontology can be used in the Taxonomy Manager. The taxonomy model was designed with this requirement in mind. An Ontology Management System can be developed in the near future. This system will help extending the Taxonomy Manager to support more taxonomy structures and import/export formats.

8. ACKNOWLEDGMENT



This publication has been produced in the context of the RAGE project. The project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 644187. However, this paper reflects only the author's view and the European Commission is not responsible for any use that may be made of the information it contains.

9. REFERENCES

- Abrol, M., Doshi, B., Kanihan, J., Kumar, A., Liu, J., & Mao, J. (2005). Intelligent Taxonomy Management Tools for Enterprise Content. The 2005 IEEE/WIC/ACM International Conference on Web Intelligence, 809-811. IEEE Computer Society Washington, DC, USA.
- Ajax. (2018). <https://developer.mozilla.org/en-US/docs/Web/Guide/AJAX> (March 5th, 2018).
- Alonso, O., & Lease, M. (2011). Crowdsourcing 101: Putting the WSDM of Crowds to Work for You. Proceedings of the fourth ACM international conference on Web search and data mining, 1-2. ACM New York.
- AngularJS. (n.d.). <https://angularjs.org> (March 5th, 2018).
- Apache Solr. (n.d.). <http://lucene.apache.org/solr> (March 5th, 2018).
- Asch, S. E. (1956). Studies of Independence and Conformity: I. A Minority of One Against a Unanimous Majority. Psychological Monographs: General and Applied, 1-70.
- Boiko, B. (2005). Content Management Bible. John Wiley & Sons.
- Brabham, D. C. (2008). Crowdsourcing as a Model for Problem Solving : An Introduction and Cases. Convergence. The International Journal of Research into New Media Technologies.

- Davenport, T. H., & Prusak, L. (1998). *Working Knowledge: How Organizations Manage What They Know*. Harvard Business Press.
- Definition of Taxonomy. (2017). <https://www.merriam-webster.com/dictionary/taxonomy> (May 26th, 2017).
- Estellés-Arolas, E., & González-Ladrón-de-Guevara, F. (2012). Towards an integrated crowdsourcing definition. *Journal of Information Science*, 1-14.
- Finstad, K. (2010). The Usability Metric for User Experience. *Interacting with Computers*, 22, 323-327.
- Georgiev, A., Grigorov, A., Bontchev, B., Boytchev, P., Stefanov, K., & Bahreini, K. (2016). The RAGE Software Asset Model and Metadata Model. *JCSG 2016. Lecture Notes in Computer Science*, vol 9894. Springer, Cham.
- Git. (n.d.). <https://git-scm.com> (March 5th, 2018).
- Git Basics. (n.d.). <https://git-scm.com/book/en/v2/Getting-Started-Git-Basics> (March 5th, 2018).
- GitHub. (n.d.). <https://github.com> (March 6th, 2018).
- Hedden, H. (2010). *The Accidental Taxonomist*. Information Today Inc.
- jQuery. (n.d.). <https://jquery.com> (March 6th, 2018).
- jsTree. (n.d.). <https://www.jstree.com> (March 6th, 2018).
- Kitchens, F., & Crane, C. (2014). Customersourcing: to Pay or be Paid. *Proceedings of the 27th Bled eConference, Bled, Slovenia*.
- Kon, H., & Hoey, M. (2005). Leveraging collective knowledge. *Proceedings of the 14th ACM international conference on Information and knowledge management, Bremen, Germany*, 560-567. ACM New York, NY, USA.
- Kotler, P. (1986). Prosumers: A New Type of Consumer. *Futurist*, 20(5), 24.
- Lambe, P. (2007). *Organising Knowledge: Taxonomies, Knowledge and Organisational Effectiveness*. Chandos Publishing.
- Linstone, H. A., & Turoff, M. (1975). *The Delphi Method: Techniques and Applications*. Addison-Wesley Educational Publishers Inc.
- Lund, A. (2001). Measuring Usability with the USE Questionnaire. *STC Usability SIG Newsletter*.
- Memcached. (n.d.). <https://memcached.org> (March 6th, 2018).
- Mendeley. (n.d.). <https://www.mendeley.com> (March 6th, 2018).
- Miles, A. (2006). SKOS: Requirements for standardization. *Proceedings of the 2006 international conference on Dublin Core and Metadata Applications*, 55-64.
- MySQL. (n.d.). <https://www.mysql.com/de> (March 6th, 2018).
- Nagy, H., Pellegrini, T., & Mader, C. (2011). Exploring Structural Differences in Thesauri for SKOS-based applications. *Proceedings of the 7th International Conference on Semantic Systems, Graz, Austria*, 187-190. ACM New York.
- Papadakis, I., Andreou, I., & Chrissikopoulos, V. (2012). Interactive search results. *Proceedings of the 6th European Conference on Digital Libraries-ECDL*, 448-462.
- Salman, M., Heutelbeck, D., & Hemmje, M. (2015). Towards social network support for an applied gaming ecosystem. *Proceedings of the 9th European Conference on Games Based Learning*, 721-728. Academic Conferences and Publishing International Limited, Reading, UK.
- SKOS Datasets. (2015). <https://www.w3.org/2001/sw/wiki/SKOS/Datasets> (March 6th, 2018).
- SlideShare. (n.d.). <https://de.slideshare.net> (March 6th, 2018).
- Solr Features. (n.d.). <http://lucene.apache.org/solr/features.html> (March 6th, 2018).
- StackOverflow. (n.d.). <https://stackoverflow.com> (March 3rd, 2018).
- Surowiecki, J. (2005). *The Wisdom of Crowds*. Anchor.
- Symfony Framework. (n.d.). <http://symfony.com> (March 3rd, 2018).
- The 2012 ACM Computing Classification System. (n.d.). <https://www.acm.org/publications/class-2012> (March 3rd, 2018).
- The Four V's of Big Data. (2005). <http://www.ibmbigdatahub.com/infographic/four-vs-big-data> (October 17th, 2013).
- Then, M., Wallenborn, B., Ianniello, B., Vu, B., Fuchs, M., & Hemmje, M. (2016). Innovative Authoring Tools for Online-Courses with Assignments - Integrating Heterogeneous Tools of e-Learning Platforms into Hybrid Application Solutions. *International Journal of Emerging Technologies in Learning*.
- Then, M., Wallenborn, B., Ianniello, B., Vu, B., Swoboda, T., Fuchs, M., & Hemmje, M. (2016). A Competence-Based Course Authoring Concept for Learning Platforms with Legacy Assignment Tools. *GSTF Journal on Computing (JoC)*.

- Thielsch, M. T., & Stegemöller, I. (2014). Münsteraner Fragebogen zur Evaluation - Zusatzmodul Diskussion . Zusammenstellung sozialwissenschaftlicher Items und Skalen.
- Tunkelang, D. (2009). *Faceted Search*. Morgan & Claypool.
- Typo3. (n.d.). <https://typo3.org> (March 3rd, 2018).
- Vu, B. (2015). *Realizing an Applied Gaming Ecosystem - Extending an Education Portal Suite towards an Ecosystem Portal*. TU Darmstadt.
- What is Symfony. (n.d.). <http://symfony.com/what-is-symfony> (March 3rd, 2018).
- Whittaker, M., & Breininger, K. (2008). *Taxonomy Development for Knowledge Management*. World Library and Information Congress: 74th IFLA General Conference and Council, Québec, Canada.
- Wikipedia. (n.d.). <https://www.wikipedia.org> (March 3rd, 2018).