Metadata Management for Large Information Spaces

Karol Rástočný*
Institute of Informatics, Information Systems and Software Engineering
Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava
Ilkovičova 2, 842 16 Bratislava, Slovakia
karol.rastocny@stuba.sk

Abstract

Due to size and heterogeneity of large information spaces, methods of data processing use metadata as their main source for their tasks. Usage of already created metadata decreases necessity of raw data preprocessing and it increases efficiency of data processing methods. But large information spaces as the Web or especially source code repositories are not stable. Information stored in them are modified continuously. These modifications affect quality of metadata.

In our work we challenge the problem of metadata management for large information spaces, while we focus to three main goals: (I) proposition of a metadata model suitable for information exchange and efficient metadata maintenance; (II) proposition of scalable metadata repository which respects characteristics the metadata model; (III) approach to medatada maintenance which keep metadata valid and consistent.

To fulfil these goals we propose novel metadata representation via information tags as class of descriptive metadata. We also proposed information tags model based on standardized Open Annotation model and information tags repository which provides effective access to information tag for main information tags use cases. We address metadata maintenance via proposition of robust location descriptor for anchoring information tags to source code and the maintenance approach based on querying a stream of events about tagged content.

Categories and Subject Descriptors

H.3.5 [**Information Storage and Retrieval**]: Online Information Services—*Data sharing*; D.2.8 [**Software Engineering**]: Metrics

*Recommended by thesis supervisor: Prof. Mária Bieliková
To be defended at Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava on June 29, 2016.

Keywords

information tags, descriptive metadata, metadata management, developer activities, empirical software metrics

1. Information Tags

Large information spaces like the Web or information spaces of software houses are create by humans with the main idea to share information to humans. So resources of these information spaces are structured with focus to readability and understandability by humans. In spite of this, these information spaces are not directly accessed by humans but they are processed by systems with the goal to make information accessible to humans. But only direct presenting resources to humans is often not enough and systems have to be able to find correct, requested information. To fulfil goals like this, systems have to process resources to obtain new information or knowledge from information spaces. In this cases it is not efficient for systems to directly process resources but they use descriptive metadata (describe resources with information identifying resources [4], e.g. titles of web-page) with previously obtained information about resources.

In the case of the Web is this problem addressed by the Semantic Web initiative, which stores descriptive metadata in ontologies [13] and maps them to web-pages via semantic links[1], microdata[1] or microformats[2], that are stored directly in web-pages. Although this solution can be generalized to other domains, it has to deal with the problem of instability of information spaces. Resources of information spaces are continuously created, modified and removed from information spaces. All these modifications leads to necessity of updating ontologies and references to them, what is in case of ontologies often problematic and it is ineffective to process all resources if they do not contain reference to updated parts of ontologies.

We address this problem by proposition of *information tags* as subset of descriptive metadata with semantic relations to a tagged content. We define information tags are defined by triplet (Type, Anchoring, Body) [2, 8]:

- *Type* – defines a type and a meaning of the information tag;
- *Anchoring* – identifies the tagged information artifact;

[1]http://www.w3.org/TR/microdata/
[2]http://microformats.org/

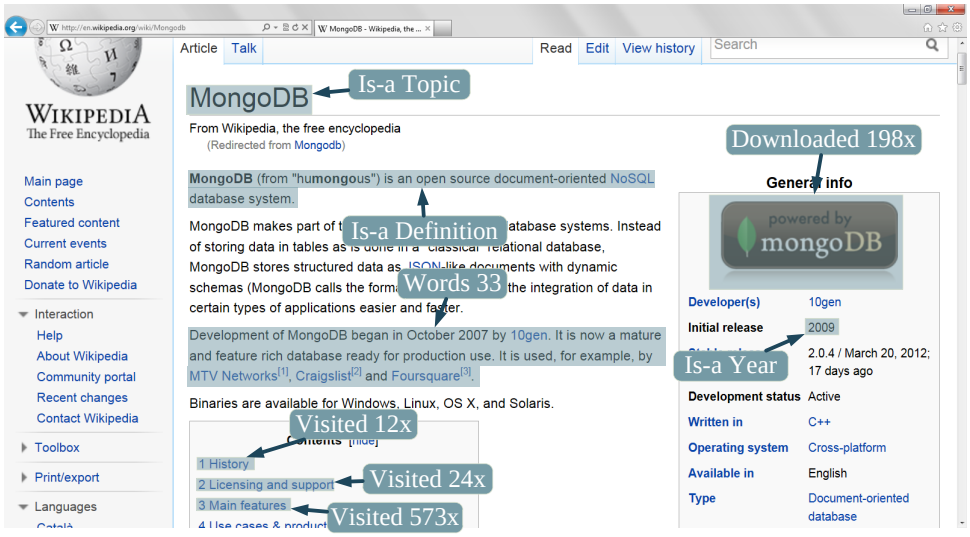

Figure 1: Examples of simple information tags in a human-readable format. An arrow and a highlighted text represent anchoring, the first word defines a type and the second word represents a body of an infor-mation tag.

- *Body* – represents a structured information, the structure of which corresponds to the type of the information tag.

An information tag is a tag which contains structured machine-readable information which tags information artifact with its property, e.g. number of words in a paragraph, type of phrase, number of downloads or visits of some resource (see Figure 1). As a result information tags can describe the whole resource in detail with its global (e.g., the topic of the resource) and partial (e.g., a number of clicks on hyperlinks in the resource) properties that can be used for various purposes such as comparing documents or mining new information.

Main advantage of information tags is their independence from tagged resources. All informations stored in information tags can be used without necessity to access resources and information tags directly reference tagged resources so edits of resources do not affect references to them.

2. Information Tags Model and Repository

An information tag is a triplet of a body, an anchoring to an information artifact and a type. These features are common with human annotations, so information tags can be modelled as annotations. Annotations' "data structure" has quite long history and so several annotation models have been standardized. A problem is that annotation models have not been proposed with the requirement to efficiency of annotation maintenance by reason that maintenance of a freeform, human-readable body is too complex task.

To supply acceptable information tags model, we based the information tags model on existing the Open Annotations Model [11]. The Open Annotation Model has been proposed by wide Open Annotation community and it provides lot of possibilities that covers almost all requirements of different types of annotations. Since an information tag is not so complex data structure as general annotation and we have to respect specific requirements of information tags repository âĂŞ efficient access, efficient maintenance, ease of a use and scalability [9], we have lightened the Open Annotation Model (we have retained only inevitable elements) and redesigned the model to the object model.

Based on this redesign of the model we are able to store information tags in scalable document databases. These databases give as possibility to manipulate with information tags as with whole units and we do not have to request multiple RDF triplets. On the other side it decrease inference possibilities of the Open Annotation Model. We deal with this problem by proposition of SPARQL query processing algorithm base on MapReduce programming model. By evaluation of this algorithm we proved, that it has performance comparable with native RDF repositories [9].

3. Information Tags Maintenance

We split the problem of information tags maintenance to two independent sub-problems:

- Maintenance of information tag anchors – information tags refers parts of resources, so when tagged resources have been changed, these references should be repaired;
- Maintenance of information tag bodies – bodies of information tags contains main information so they are the most sensitive to changes in information spaces.

3.1 Maintenance of Information Tags Anchors

Repairing metadata anchoring after a modification of documents belongs to basic problems of the metadata maintenance. The complexity of this maintenance is anchoring descriptor dependent. In case of textual documents, a popular anchoring descriptor is based on column and line indexes that characterize metadata position in a document. The descriptor based on indexes is easily interpretable, but it is very sensitive to changes in documents, that can affect metadata positions in three ways [7]:

- *Without a change* – a document has been modified after metadata anchoring position;
- *Simple shift* – a document has been modified before metadata anchoring position. This modification causes simple shifting metadata to new position;
- *Complex modification* – a document has been modified on the place, where metadata is anchored. In this case, a determination of new position may be complicated and a resolution whether metadata still have original meaning or they have to be updated or deleted has to be made.

In case of textual documents there are many solutions to this problem, e.g. SGDOM based anchoring [5] or tree-based descriptors [3, 6]. But there is no specialized solution for source code, which respect source code characteristics, support real-time interpreting anchors and direct comparison of anchors. For this reason we analyzes more than 60,000 C# source code files and identified their characteristics that can be utilized for proposition of robust location descriptor for source code [9].

We propose descriptor, which consist of index location decriptor and context based descriptor. The index location descriptors are directly comparable and can be resolved effectively, if the tagged source code has not been modified. On the other side the context-based descriptor is robust to source code modification. For interpreting the context based descriptor we proposed algorithm which combines tokenization and string similarity algorithms with Smith-Waterman algorithm. This combination of algorithms allows real-time interpreting even after complex source code modifications [9].

3.2 Maintenance of Information Tags Bodies

Maintenance of information tags bodies has to react to all types of changes in characteristics of resources of information systems. In general we can define structural, semantic and empirical characteristics of resources. To which characteristic information tags are sensitive, depends on types of information tags. E.g. when an information tag contains LLOC source code metric, simple renaming tagged class does not affects the information tag. But if an information tag contains number of views of the class, the information tags is affected even by scrolling in a source code file.

We reflect this diversity by proposition of the tagger [10], which transforms users' and systems' activities over resources of information space to a stream of events in form of linked steam data [12]. After that the tagger queried the stream and executes maintaining actions after obtaining results of stream queries.

4. Conclusions and contributions

In the dissertation thesis we discussed problems of metadata management for large information spaces. As main problem we identify invalidation of medata caused by instability of information spaces. A solution of this problem we split to four main contributions of the thesis:

- *Information tags* – we proposed novel representation of descriptive metadata, that is natural for systems. This representation is independent from described resources what fulfils initial requirement of effective metadata maintenance.
- *Information tags repository* – to utilize contributions of information tags, we proposed information tags repository based on MongoDB that stores information tags in the model based on the standardized Open Annotation Model. This in combination of proposed SPARQL query processing algorithm guarantees integration possibility with existing systems.
- *Robust descriptor for source code* – information tags reference tagged information artifact via robust descriptors. For this reason we proposed robust location descriptor and its interpreting algorithm, which is able to identify tagged source code artifact in real-time.
- *Stream-based metadata maintenance* – to keep information tag space valid and consistent we proposed method for creating, updating and removing information tags based on querying steam of events about users and systems actions over information spaces. The method executed necessary maintenance actions after receiving results from stream queries.

We evaluate contributions of our work in the domain of the project PerConIK[3] (Personalized Conveying of Information and Knowledge). In the project we used proposed methods for management of metadata about source codes. We deploy implemented repository as main information store in the PerConIK architecture. The tagger continuously processed stream of developers' activities in IDEs (Microsoft Visual Studio and Eclipse) and web browsers and source code changes from git repositories.

In addition we proposed set of information tag types for developers, that can be used by developers for manual tagging of source code [10]. These tags are proposed mainly for reviewing source code. To support this process we develop system CodeReview[4] (see Figure 2), that has been used in school course Team project for two years.

Acknowledgements. This work was partially supported by grants No. APVV-0233-10, APVV-0208-10, VG 1/0752/14 and VG 1/0646/15 and is the partial result of the Research & Development Operational Programme for the project Research of methods for acquisition, analysis and personalized conveying of information and knowledge, ITMS 26240220039, co-funded by the ERDF.

References

[1] S. Araujo, G.-j. Houben, and D. Schwabe. Linkator: Enriching Web Pages by Automatically Adding Dereferenceable Semantic Annotations. In B. Benatallah, F. Casati, G. Kappel, and G. Rossi, editors, *Web Engineering, 10th International Conference, ICWE 2010*, volume 6189 of *Lecture Notes in Computer Science*, pages 355–369, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[2] M. Bieliková, I. Polášek, M. Barla, E. Kuric, K. Rástočný, J. Tvarožek, and P. Lacko. Platform Independent Software Development Monitoring: Design of an Architecture. In V. Geffert, B. Preneel, B. Rovan, J. Štuller, and A. M. Tjoa, editors, *SOFSEM 2014: Theory and Practice of Computer*

[3] http://perconik.fiit.stuba.sk/

[4] https://perconik.fiit.stuba.sk/CodeReview

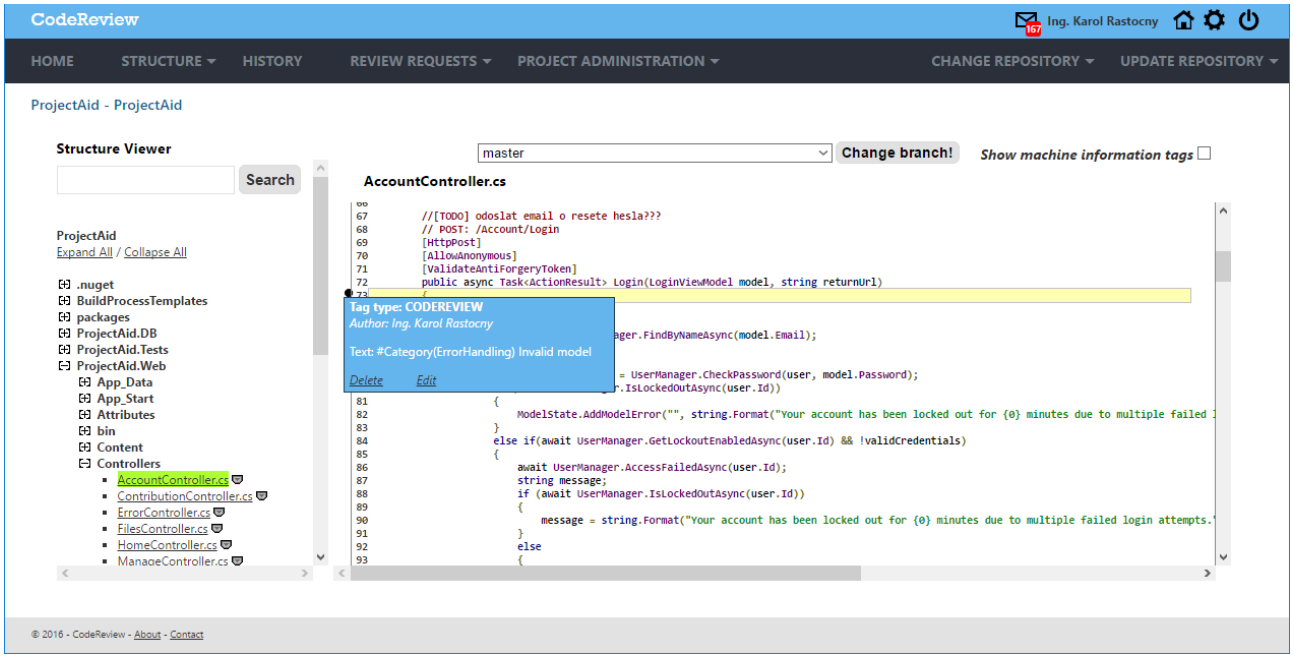

Figure 2: System CodeReview, displaying one source code file with information tag

Science, volume 8327 of *LNCS*, pages 126–137. Springer International Publishing, Cham, 2014.

[3] J. Kahan and M.-R. Koivunen. Annotea: An Open RDF Infrastructure for Shared Web Annotations. In *Proceedings of the 10th International Conference on World Wide Web - WWW'01*, pages 623–632, New York, 2001. ACM Press.

[4] NISO. *Understanding metadata*. NISO Press, Bethesda, 2004.

[5] T. A. Phelps and R. Wilensky. Robust Intra-Document Locations. *Computer Networks*, 33(1-6):105–118, jun 2000.

[6] B. Plimmer, S. H.-h. Chang, M. Doshi, L. Laycock, and N. Seneviratne. iAnnotate: Exploring Multi-User Ink Annotation in Web Browsers. In *Proceedings of the Eleventh Australasian Conference on User Interface - Volume 106*, volume 106, pages 52–60. Australian Computer Society, Inc., 2010.

[7] R. Priest and B. Plimmer. RCA: Experiences with an IDE Annotation Tool. In *Proceedings of the 6th ACM SIGCHI New Zealand Chapter's International Conference on Computer-human Interaction Design Centered HCI - CHINZ'06*, pages 53–60, New York, 2006. ACM Press.

[8] K. Rástočný and M. Bieliková. Maintenance of Human and Machine Metadata over the Web Content. In M. Grossniklaus and M. Wimmer, editors, *Current Trends in Web Engineering (ICWE 2012)*, volume 7703 of *LNCS*, pages 216–220. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[9] K. Rástočný and M. Bieliková. Metadata Anchoring for Source Code: Robust Location Descriptor Definition, Building and Interpreting. In H. Decker, L. Lhotská, S. Link, J. Basl, and A. M. Tjoa, editors, *Database and Expert Systems Applications*, volume 8056 of *LNCS*, pages 372–379. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[10] K. Rastocny and M. Bielikova. Empirical metadata maintenance in source code development process. In *Engineering of Computer Based Systems (ECBS-EERC), 2015 4th Eastern European Regional Conference on the*, pages 25–31, Aug 2015.

[11] R. Sanderson, P. Ciccarese, and H. Van de Sompel. Designing the W3C open annotation data model. In *Proceedings of the 5th Annual ACM Web Science Conference on - WebSci '13*, pages 366–375, New York, 2013. ACM Press.

[12] J. F. Sequeda and O. Corcho. Linked Stream Data: A Position Paper. In *Proceedings of the 2nd International Workshop on Semantic Sensor Networks*, pages 148–157, CEUR-WS, 2009. CEUR-WS.

[13] N. Shadbolt, T. Berners-Lee, and W. Hall. The Semantic Web Revisited. *IEEE Intelligent Systems*, 21(3):96–101, may 2006.

Selected Papers by the Author

K. Rástočný, M. Tvarožek, M. Bieliková. Web Search Results Exploration via Cluster-Based Views and Zoom-Based Navigation. *Journal of Universal Computer Science*, 19(16): 2320–2346, 2013.

K. Rástočný, M. Tvarožek, M. Bieliková. Supporting Search Result Browsing and Exploration via Cluster-based Views and Zoom-based navigation. In *Proceedings 2011 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Workshops*, pages 297–300, Lyon, France, 2011. CS IEEE Press.

K. Rástočný, M. Bieliková. Maintenance of Knowledge Tags within Heterogeneous Web Content. In *Proceedings of Current Trends in Web Engineering: ICWE 2012 International Workshops MDWE, Composable Web, WeRE, QWE, and Doctoral Consortium*, LNCS 7703, pages 216–220, Berlin, Germany, 2012. Springer.

M. Bieliková, K. Rástočný. Lightweight Semantics over Web Information Systems Content Employing Knowledge Tags. In : S. Castano et al., editors *ER Workshops 2012*, LNCS 7518, pages 327–336, Lyon, Italy, 2012. Springer.

K. Rástočný, M. Bieliková. Metadata Anchoring for Source Code: Robust Location Descriptor Definition, Building and Interpreting. In : H. Decker et al., editors *DEXA 2013, Part II*, LNCS 8056, pages 372–379, Prague, Czech Republic, 2013. Springer.

M. Bieliková, I. Polášek, M. Barla, E. Kuric, K. Rástočný, J. Tvarožek, P. Lacko. Platform Independent Software Development Monitoring: Design of an Archi-tecture In : V. Geffert et al., editors *SOFSEM 2014*, LNCS 8327, pages 327–336, Starý Smokovec, Slovak Republic, 2014. Springer.

K. Rástočný, M. Bieliková. Enriching Source Code by Empirical Metadata. In *ESEM 2014: 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, page 1, Torino, Italy, 2014. ACM.

K. Rástočný, M. Bieliková. Empirical Metadata Maintenance in Source Code Development Process. In *ECBS-EERC 2015: 2015 IEEE Fourth Eastern European Regional Conference on the Engineering of Computer Based Systems*, pages 25–31, Brno, Czech Republic, 2015. CS IEEE Press.