The Troll Under the Bridge: Data Management for Huge Web Science Mediabases

Ralf Klamma, Zinayida Petrushyna
Databases & Information Systems
RWTH Aachen University, Ahornstr. 55, D-52056 Aachen, Germany
{petrushyna|klamma}@dbis.rwth-aachen.de

Abstract: In the emerging discipline of web science there will be a growing need for managed data sets for research purposes. In the moment, data sets are mainly managed on file systems in arbitrary formats. This situation leads to a lot of media breaks and double work in the scientific workflows. We argue for the concept of a Web 2.0 Mediabases, a data set of managed Web 2.0 data from blogs, wikis, podcasts etc. In such a Mediabases we can apply web science methods and orchestrate complex experiments and simulations.

1 Introduction

Web Science [BLHH+06] has become a new and thrilling area of computer science where the topic of research is the many traces people leave while using the Internet, especially a new class of software called *Social Software* on the Web 2.0 [O'R05]. Web Science is a truly interdisciplinary area with its roots in disciplines like social network analysis, statistical physics and data mining. Still, many of the methodologies and tools developed until now have problems. Many of these issues are because of data management problems.

- While almost all the data sets in Web Science are based on the mathematical model of graphs they are mostly stored as plain text files on file systems in different non interoperable formats or in some XML format like GraphML [1]. This leads to a lot of efforts transforming data sets into different formats, loading them with special tools for special computational tasks, finally ending in complex scientific workflows like in the early days of almost all natural sciences.
- Most of the datasets are one-shot datasets taken at a specific time. Therefore, it is almost impossible to apply dynamic social network analysis features over time to research growth and speed dynamics of Web Science data sets.
- Most of the datasets do not allow analysis on different level of aggregation at the same time. It is nearly impossible to create equivalent data sets out of identified sub-networks or to drill down in already aggregated graph data sets.

[1] http://graphml.graphdrawing.org

- All the single arguments above motivate the need of a Mediabase. It aims to design the datasets according to interoperable standards allowing to follow dynamic changes and aggregate new instances. Nowadays, the possible combination of the approaches mentioned is beyond the scope of every system aware to us.

The Mediabase approach tries to overcome the inconveniences and complexity of current research in Web Science by delivering a tailorable out-of-the-box analysis environment for Web Science data sets. A Mediabase always consists of a set of Web 2.0 media (cf. **Media**

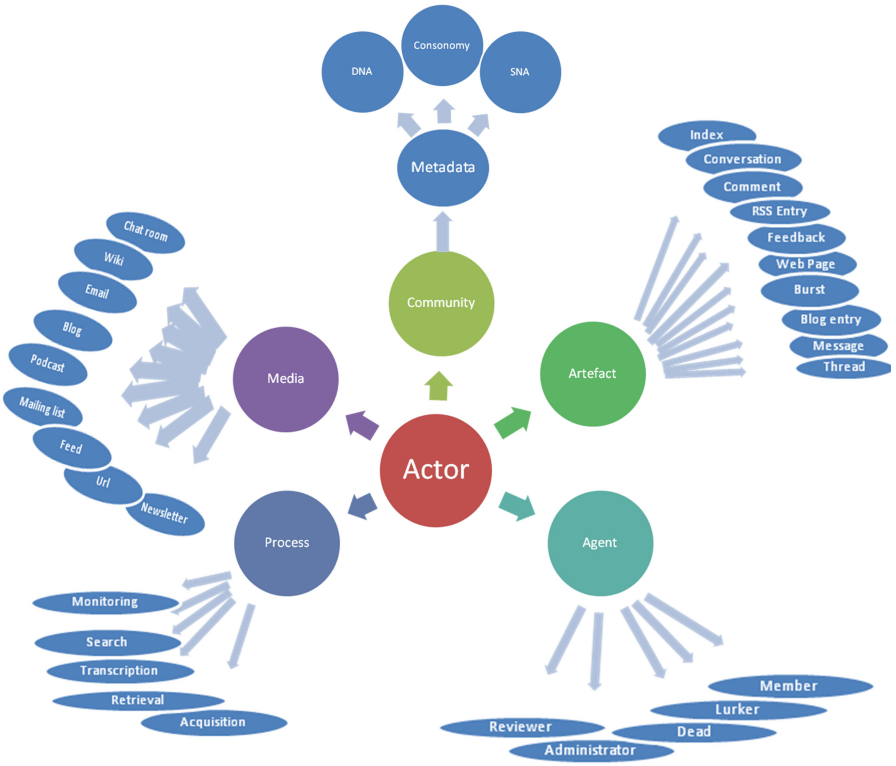

Figure 1: The general model of the repository of media

in Figure 1) like Blogs, Wikis or Podcasts, a set of artefacts (cf. **Artefact** in Figure 1) depending on the media in the Mediabase, a set of media operations (cf. **Process** in Figure 1), a set of agents (cf. **Agent** in Figure 1), a set of Communities (cf. **Community** in Figure 1) consisting of agents and a set of community-specific metadata (cf. **Metadata** in Figure 1). Following Actor-Network Theory [Lat99] we do not differentiate between human and non-human actors in networks, even media operations themselves can be actors [Kla00]. Having the concept actor, we have a very strong concept for aggregating social entities defined through the traces they left in Social Software.
Technically, a Mediabase consists out of a database backend for relational and XML

database like IBM DB2, Exists and MySQL, a collection of PERL/Python scripts for crawling different Web 2.0 media automatically, a collection of analysis routines both on the database side (PL/SQL) or on the client side (Java) and different front-ends for graph visualization (yFiles) and user interaction (Zope/Plone).
The reminder of the paper is organized as follows. In the next section we define formally the Mediabase and its concepts. The paper concludes with a discussion and an outlook.

2 Mediabases: A Primer

We define the Mediabase as a graph with labels on vertices and edges. Every edge of the graph can be directed or indirected. Subgraphs of the graph are considered as contributors to different dimensions of Mediabase evolution. Following we present a number of definitions that we are using for the graph representation of the huge repository of media. The key word is huge. We aimed to create the repository with various media types although it is impossible to follow the inception of new media thus we try to move from particular examples to abstract one.

Definition 1 *A Mediabase is a six-tuple graph* $M = (A, R, \mu, \nu, \eta, L)$*, where A - is the set of nodes,* $R \subseteq A \times A$ *- is a set of edges,* $\mu : A \rightarrow L$ *- is the function assigning labels to nodes,* $\nu : R \rightarrow L$ *- is the function assigning labels to edges,* $\eta : R \rightarrow \{0, 1\}$ *- is the function assigning the direction property to the edges, i.e 0 for undirected and 1 for directed. L - is a set of labels of nodes and edges.*

Definition 2 $A \subseteq \{$*Medium*, *Artefact*, *Process*, *Agent*, *Network*$\}$. *Medium*$\subseteq \{$*Mailing lists*, *Newsletter*, *Newsgroup*, *Feed*, *Web-site*, *Blog*, *Podcast*, *Chat room*, *Wiki*, *Forum*, *Social bookmarking site*, *Folksonomy*$\}$. *Artefact*$\subseteq \{$*Message*, *E-mail*, *Index*, *Comment*, *RSS Entry*, *Transaction*, *Host*, *Feedback*, *Conversation*, *Burst*, *Blog entry*, *Thread*, *Executions*, *Tag*, *Trackback*, *Review*, *URL*, *Rating*, *Multimedia*, *Rankíng*, *Reference*$\}$. *Process*$\subseteq \{$*Acquisition*, *Search*, *Monitoring*, *Retrieval*, *Transcription*, *Addressing*$\}$. *Agent*$\subseteq \{$*Administrator*, *Member*, *Lurker*, *Reviewer*, *Dead*, *Answering person*, *Questioner*, *Troll*, *Spammer*, *Conversationalist*, *Expert*$\}$.

Definition 3 *The observation service* ***Watcher*** *is a system that uses the data presented of the following graph* $W = Media \cup Artefact$ *for the inspection process.*

Definition 4 *The interaction of a user with a medium is presented by the graph* $I = Medium \cup Artefact \cup Process \cup Agent$*.*

Definition 5 *The graph of the community analysis system in the Mediabase repository is* $G = Medium \cup Artefact \cup Process \cup Agent \cup Network$*.*

The application of the definitions to the current components of the media base implies into the following definitions:

Definition 6 *The observation service **Blogwatcher** is the system that uses the graph* BW=$Blog \cup Blogroll \cup Blogentry \cup Comments \cup Index$. *There are a medium and artefacts of the medium that conjuncts within the graph. The other watchers use the same mechanism for the graph construction, e.g Listwatcher, Feedwatcher, Sitewatcher, etc.*

We define a specific socio-computational task here which serves as an example for the kind of investigations we have in mind using the Mediabase. Many years of study of online communication has lead to a differentiation between agents, i.e. members of the community, (cf. [MN04, CLIW05, TSFW05, FSW06]. These are e.g. questioners, answering people, trolls, conversationalists, and spammers [KSJ06]. Internet trolls are agents posting irrelevant messages in community media like forums, blogs or mailing lists [2]. In the Mediabase we can find trolls in different media by defining that trolls are agents answering only in threads started by themselves. Therefore, we need artefacts like threads, messages and agents like authors and creators. On the graph definition level we can do that by the following set of rules.

Example 1 $\exists$ *Thread* $th : th \subseteq$ *Artefact*

$\exists$ *thread* th, $th_1 : th, th_1 \in$ *Thread.*

$\exists$ *Medium* m, $\exists$ *Artefact* $a : a \in$ *th* $\wedge\ v(m,a) =$ *stored on*.

$\exists$ *Message* $msg : msg \subseteq$ *Artefact.*

$\exists$ *Process* $P : P \subseteq$ *Process.*

$\exists$ *Agent* $Ag : Ag \subseteq$ *Agent.*

$\exists$ *Agent* cr, $\exists$ *Process* p, $\exists$ *Artefact* $a : cr \in$ *Ag* $\wedge\ cr \subseteq$ *Member* $\wedge\ p \in$ *P* $\wedge\ p \subseteq$ *Autoring* $, \wedge\ a \in th\ v(cr,p) =$ *performs*

$\exists$ *Agent* au, $\exists$ *Process* p, $\exists$ *Artefact* $a : au \in$ *Ag* $\wedge\ au \subseteq$ *Member* $\wedge\ p \in$ *P* $\wedge\ p \subseteq$ *Authoring* $\wedge\ a \in msg\ v(au,p) =$ *performs.*

$\exists$ *Thread* $msgTh$, $\exists$ *Process* p, $\exists$ *Artefact* $a : msgTh \in$ *th* $\wedge\ p \in$ *P* $\wedge\ a \in$ *msg* $v(p,msg) =$ *performs;* $v(p,a) =$ *performs*; $v(msg,msgTh) =$ *postedIn*;.

$\exists$ *troll* $t : t \in$ *Ag* $\wedge\ creator = t \wedge creator \in cr \wedge author = t \wedge author \in au \wedge messageThread \in msgTh \wedge |\{$*msg*$\}| < minPosts \wedge \neg\ creator_1 \neq t \wedge creator_1 \in cr \wedge \neg\ author_1 = t \wedge author_1 \in au \wedge \neg\ messageThread_1 \in msgTh$

Such an expression can be written down in SQL using a relational data model for the graph structure. A first attempt to present such a kind of knowledge in a more intuitive way for the definition of observable behaviour and traces in Social Software was done in [KSD06] by using a pattern language. It includes a pattern structure definition, the Formal Expression Language for Patterns (FELP) and algorithms for the application of the patterns on the Mediabase. Each pattern consists from a name, a disturbance, a description, forces, force relations, a solution a rationale and pattern relations. The existence of patterns in digital social networks is defined by a repeatedly occurring condition, i.e. a disturbance. It is described in FELP as a formal expression that defines the problem the pattern aims

[2] `http://en.wikipedia.org/wiki/Troll_(Internet)`

to solve. The forces are relevant actors of the pattern; the forces relations are relations between actors. The solution provides the advices to solve the pattern situation. The relations is used for reasoning about the forces and the disturbances. The patterns relations defines how the pattern is related to the others. A pattern expressions that is an instance of an existing pattern $troll$ looks like

Example 2 $\exists[troll|\,(\exists[thread|\,(thread.author = troll) \wedge$
$(count[message|\,(message.author = troll) \wedge (message.posted = thread)]) > minPosts])$
$\wedge\,(\neg\exists[thread_1, message_1|\,(thread_1.author \neq troll) \wedge$
$(message_1.author = troll \wedge message_1.posted = thread_1)])]$

A troll exists when there exists a thread where the author is the troll and a number of messages posted by the troll is more than defined. We exclude cases when the troll posted messages in threads that he hadn't created. The described pattern is applicable to any particular case of data in the Mediabase. Patterns allow to create equivalent interpretations of data sets on an abstract level as well as to inspect changes over time in the data sets.

The existence of a common graph model makes the scholars free of the time-consuming transformation of data sets into different other formats since the Mediabase is capable of this.

3 Discussion and Outlook

The Mediabase approach is a first answer to the demanding problems of Web Science to have comprehensive data management platforms for the storage, the maintainance, the analysis and in future the simulation of Web Science data sets, namely large dynamic social graphs. The Mediabase is the outcome of several research projects dealing with Social Software (EU Network of Excellence PROLEARN, German Excellence Cluster Ultra High-Speed Information and Communication UMIC) and the future design of workplaces for scientists (German Collaborative Research Cluster FK 427 "'Media and Cultural Communication"'). For those projects we have set up different Mediabases. We use different watchers for Mediabase installation: $listwatcher$ for mailing lists, $newswatcher$ for newsletters, $sitewatcher$ for sites, $feedwatcher$ for feeds, $blogwatcher$ for blogs, and $podcastwatcher$ for podcasts, etc.

The PROLEARN Academy Mediabase is collecting materials from E-Learning communities with altogether 230000 artefacts, the GRAECULUS Mediabase is collecting materials from the humanities with 530000 artefacts and the Bamyian Valley Mediabase is collection materials from cultural heritage management with several hundert artefacts. The number of artefacts is growing creating a demanding need for advanced data management capabilities both on the frontend and the backend side. On the frontend we need more server-farm like architecture to load balance the many requests of external users, on the backend we need huge RAID arrays for a fault tolerant data management, computing and main memory facilities for the extremely challenging computation of advanced dynamic social network patterns on graph with more than 1.000.000 nodes and 40.000.000 edges.

For many media in the Web 2.0 we have defined and implemented databases backends, crawler scripts and interfaces to the user. The next steps include a media warehouse approach for the Mediabase to compare different Mediabases on an even more integrated level, the handling of data uncertainty and inconsistent data through users (user generated data), Web 2.0 style interfaces to the Mediabase, e.g. with the AJAX version of yFiles and many more.

References

[BLHH+06] Tim Berners-Lee, Wendy Hall, James Hendler, Nigel Shadbolt, and Daniel J. Weitzner. Computer Science: Enhanced: Creating a Science of the Web. *Science*, 313:769–771, 8 2006.

[CLIW05] Karen S. K. Cheung, Fion S. L. Lee, Rachael K. F. Ip, and Christian Wagner. The Development of Successful On-Line Communities. *International Journal of the Computer, the Internet and Management*, 13(1):77–89, 4 2005.

[FSW06] Danyel Fisher, Marc Smith, and Howard T. Welser. You Are Who You Talk To: Detecting Roles in Usenet Newsgroups. In *Proceedings of the 39th Hawaii International Conference on System Sciences*, 2006.

[Kla00] Ralf Klamma. *Vernetztes Verbesserungsmanagement mit einem Unternehmensgedächtnis-Repository*. PhD thesis, RWTH Aachen, 2000.

[KSD06] Ralf Klamma, Marc Spaniol, and Dimitar Denev. PALADIN: A Pattern Based Approach to Knowledge Discovery in Digital Social Networks. In K. Tochtermann and H. Maurer, editors, *Proceedings of I-KNOW '06, 6th International Conference on Knowledge Management, Graz, Austria, September 6 - 8, 2006, J.UCS (Journal of Universal Computer Science) Proceedings*, pages 457–464. Springer, 2006.

[KSJ06] Ralf Klamma, Marc Spaniol, and Matthias Jarke. Pattern-Based Cross Media Social Network Analysis for Technology Enhanced Learning in Europe. In Wolfgang Nejdl and Klaus Tochtermann, editors, *Proceedings of the First European Conference on Technology Enhanced Learning, Crete, Greece, October 3-5*, volume 4227 of *LNCS*, pages 242–256, Berlin Heidelberg, 2006. Springer-Verlag.

[Lat99] Bruno Latour. On Recalling ANT. In J. Law and J. Hassard, editors, *Actor-Network Theory and After*, pages 15–25. Oxford, 1999.

[MN04] T.R. Madanmohan and Siddhesh Navelkar. Roles and knowledge management in online technology communities: an ethnography study. *International Journal of Web Based Communities*, 1:71–89, 2004.

[O'R05] T. O'Reilly. What Is Web 2.0 - Design Patterns and Business Models for the Next Generation of Software. http://www.oreillynet.com/lpt/a/6228, 2005.

[TSFW05] Tammara Combs Turner, Marc A. Smith, Danyel Fisher, and Howard T. Welser. Picturing Usenet: Mapping Computer-Mediated Collective Action. *Journal of Computer-Mediated Communication*, 10:7, 2005.