Thoughts on a Topic Mappish LAMP Stack

Johannes Schmidt[1]

[1]INSTANT Communities GmbH
js@sixgroups.com

Abstract. In this paper requirements and a possible impact of a Topic Mappish LAMP stack are outlined. Today, the LAMP stack underlays numerous leightweight CMS and blogging systems which - forming the *Blogosphere* - constitute a significant area of the WWW.

However, penetration with Topic Maps technology is minor in this context. An asymmetry between the topology of the Topic Maps engine landscape and the topology of the web regarding popular software set-ups like LAMP is detected. Therefore push effects can hardly be realized pervasively.

A subject-centric view *on* and access *to* content in leightweight CMS e.g. allows the establishment of more reliable semantics in content syndication processes. The resulting benefits for content producers and consumers enable the realization of pull effects. An overall vision regarding global integration and federation is finally essayed.

1 Introduction

In TMRA's 2006 poster session two then new Topic Maps projects PHPTMAPI [5] and QuaaxTM [13] were presented claiming i.a. “Topic Maps for the masses” [14]. In 2006 discussion on the web was dominated by buzzword *Web 2.0* which refered to a new generation of the WWW where now - amongst other phenomenons - the user was delegated to become a significant content producer and provider e.g. using leightweight CMS or specialized blogging systems. Meanwhile the *Blogosphere* has established as a significant information space.

Considering this background which is heavily abridged here the authors of the mentioned projects thought it would help increase the awareness and the adoption of Topic Maps if - generally speaking - Topic Maps software is provided for the widespread LAMP stack[1] on which many popular publishing systems are built. PHPTMAPI is a port of TMAPI [1] to PHP5 and provides an API for creating and manipulating topic maps; QuaaxTM is an implementation of PHPTMAPI.

The author of this paper believes that Topic Maps still suffers from a lack of awareness and adoption in the WWW. The assumption is expressed that Topic Maps

[1] LAMP: acronym which refers to a software stack consisting of free and open source software used on the web. “L” is for Linux, “A” for Apache webserver, “M” for MySQL RDBMS (alternatively PostgreSQL), and “P” stands for scripting languages such as PHP, Perl, or Python.

is faced with the issue of deficient technological penetration of the web. In this paper this issue is further discussed under consideration of the topology of the Topic Maps engine landscape. The objective of this paper is to identify approaches by outlining requirements for a Topic Mappish LAMP stack. Finally a decent vision of the potential of such an infrastructure is essayed.

2 Notices on the Context

Beside the partially shifted role allocation in the process of content *creation* content *syndication* is another major evolution observable on the web within the last years. By content syndication the author primarily refers to the exchange of data via REST APIs using XML formats such as RSS or Atom. Such portions of data - documents in fact - are generally called *feeds* or *news feeds*. The variety of dialects and extensions of RSS and Atom as well as the inconsistent use of these formats massively increases the efforts to syndicate content from different sources based on reliable semantics. The basic problem is that the content providers do not (or rather can not) establish a subject-centric perspective on their data. Moore describes a possible methodology which he calls "content item proxification and classification" [9]. This approach is enabled by Topic Maps. Topic Maps allows a) the *proxification* of any *subject* and b) the declaration of any assertion on such a proxy (a topic). This does not only apply to local contexts defined by individual CMS or blogging systems but also to the global context which is the WWW. The Topic Maps overall objective of Subject Location Uniqueness (SLUO) must be noticed here.

It is evident that common publishing systems built on the Relational Model are not able to make a *fully featured* subject-centric perspective possible while Topic Maps Data Model (TMDM) in contrast is fundamentally designed for this task.

2.1 The LAMP Stack

As mentioned above some of the most popular leightweight publishing systems rely on the LAMP stack. The main purpose of those systems is to enable individuals, journalists, and companies to run their weblogs. Conroy has identified the most popular CMS in blog search engine Technorati[2] analyzing the top 100 blogs [4]. 34 of them use WordPress[3], built on PHP and MySQL, 16 use Movable Type[4] which is built on Perl and MySQL or PostgreSQL, or SQLite. This results in an amount of 50% of the top 100! Since 2002 Technorati has indexed more than 133 million blog records [16].

[2] http://technorati.com/

[3] http://wordpress.org/

[4] http://www.movabletype.org/

There are some arguments to explain the popularity of LAMP that - in turn - explains the popularity of the named CMS software (which is bound to the underlying technology). As each component of the stack is free and open source software this a) means no costs to set up such an environment and b) this only involves low costs for operations as any licence fees are omitted. Further, knowledge about the stack especially the programming languages is relatively widespread. PHP for example is not only discussed in the computer sciences but also often in the library and information sciences and other fields of study. These arguments constitute evident pull effects.

According to Netcraft's July 2009 web server survey the Apache webserver is the leading application gaining 47.17% followed by Microsoft with 23.34% [11]. In order to identify the distribution of the “P” in LAMP the respective Apache modules can be considered. Securityspace's July 2007 Apache module report specified the following order: PHP 38.00%, perl 8.11%, PHP-CGI 3.04%, Python 2.05%, mod_python 2.05% [15]. The outstanding role of PHP is obvious.

It is plausible that there is a positive feedback loop between the considerable supply of technology resulting in push effects on the one hand and the mentioned pull effects on the other hand.

3 The Potential of Topic Maps

While millions of leightweight CMS provide huge amounts of content there hardly exist means to establish reliable semantics in content syndication processes. A fundamental step towards an approach is the identification of subjects in a *proxification* which must consider a later federation in a global heterogenous space like the WWW. This is i.a. enabled by Topic Maps. *Subject identifiers* allow the referencing to subjects or rather their representations in a global context.

Providing Topic Maps technology such as Topic Maps engines which are pluggable to e.g. blogging software without higher efforts means realizing push effects for - generally speaking - a subject-centric approach in the described context.

For the vast field of online communities which is another trend on the web for years, Kindsmüller, Milz, and Schmidt have detected a paradigm shift. As syndication processes and sometimes even backchannel features between different technical systems are blurring the boundaries of these, a certain technical carrier like a community platform is no more the defining feature of community membership [7]. Now, a community can emerge around a “shared common subject” [7] in a global or meta space.

The exchange of data is thereby eased by two aspects. On the one hand various popular platforms such as Twitter, Facebook, or FriendFeed provide public content APIs. On the other hand standardization approaches such as OAuth[5] which allow secure API authorization are upcoming. In summary, the commonly shared *subject* itself becomes the defining feature for online community membership. This conclusion

[5] http://oauth.net/

underlines the potential of the subject-centric approach as well as a promising area of application for Topic Maps in general.

3.1 Topology of the Topic Maps Engine Landscape

Basically, a Topic Maps aligned push effect is possible with the supply of technology which enables subject-centric processing of content. The landscape of Topic Maps engines which allow the creation and manipulation of topic maps has the following topology.

Table 1. Topology of the Topic Maps engine landscape[6]

Engine	Programming Language/ Platform (topic maps storage)	License
Ontopia, http://code.google.com/p/ontopia/	Java (different backends)	Apache License 2.0
tinyTiM, http://tinytim.sourceforge.net/	Java (in memory storage)	Apache License 2.0
TM4J, http://tm4j.org/	Java (different backends)	Apache License
XTM4XMLDB, http://xtm4xmldb.sourceforge.net/	Java (native XML database)	GPL
Perl TM, http://search.cpan.org/dist/TM/	Perl (different backends)	Artistic License
Mappa, http://code.google.com/p/mappa/	Python (different backends)	New BSD License
RTM, http://rtm.topicmapslab.de/	Ruby (multiple RDBMS)	custom license
QuaaxTM, http://quaaxtm.sourceforge.net/	PHP5 (MySQL RDBMS)	LGPL
TM++, http://tmplusplus.sourceforge.net/	C++ (XTM filestore)	custom MIT License
SharpTM, http://code.google.com/p/sharptm/	C# (in memory storage)	New BSD License
TopiEngine, https://launchpad.net/~topiengine	C++ (SQLite embedded RDBMS)	GPL
TM4Jscript, http://tm4jscript.sourceforge.net/	JavaScript (Browser based, filestore)	custom license

[6] In this overview only standalone Topic Maps engines or software primarily used as TM engine are considered. Beside (more or less) pure engines there is various free and commercial Topic Maps software such as converters, browsers and visualizers, or CMS. See e.g. http://www.topicmap.com/tools for further information.

Isidorus, http://common-lisp.net/project/isidorus/	Lisp (Berkeley DB)	Lisp-LGPL
TMCore, http://www.networkedplanet.com/Products/TMCore/	.NET platform (Microsoft SQLServer RDBMS)	Commercial

In the examination of this topology three engines pluggable to the LAMP stack ("P" engines) can be discovered: Perl TM, Mappa (Python), and QuaaxTM (PHP). (Beside the programming language the license must be considered too.) The Java engines play a leading role not only from a quantity perspective but also a quality perspective. The Ontopia engine e.g. has become open source in 2009 after some years of commercial alignment. It is richly featured and is driving a couple of commercial/professional applications and portals. Today - as an open source project - Ontopia involves 9 project members while a considerable amount of the other projects is a single developer business. This also holds for the "P" engines except Mappa which involves two members. TinyTiM, another Java engine, is an up to date project e.g. having implemented the new generation of the Topic Maps API - TMAPI 2.0[7] - as first project worldwide.

While the Java engines play a quite leading role in the Topic Maps sphere an analysis of Java's relevance on the web is dared. Securityspace's Apache module report allows some indication on this aspect. In production environments an Apache webserver is often pre-set to a Tomcat Java server [17]. Certain modules then are used as connectors which are mod_jk, mod_jk2, and mod_webapp. The frequency of these modules is: mod_jk 2.73%, mod_jk2 0.52%, mod_webapp 0.46%. Module "Tomcat" gains 0.03% [15].

3.2 Status

Compared to the frequency of "P" engine enabling modules (see 2.1) the situation on the web can be characterized differently. However, it must be emphasized that the consulted references do not allow waterproof assumptions but approximation! The following is expressed though: It is the detection of an asymmetry between the topology of the Topic Maps engine landscape and the topology of the web especially regarding popular software set-ups like LAMP which e.g. significantly underlays the Blogosphere.

[7] http://www.tmapi.org/2.0/

4 Requirements for a Topic Mappish LAMP Stack

The technological asymmetry mentioned in the last paragraph prevents a pervasive push effect into the wider area of the LAMP stack driven WWW to some extent. This - in turn - respectively prevents the pervasive adoption of Topic Maps in this context. The objective of a Topic Mappish LAMP stack is to enable subject-centric processing in general. For the numerous leightweight CMS in particular a Topic Maps enabled infrastructure enables a subject-centric view *on* and access *to* existing content - thus a Topic Maps layer.

The basic LAMP stack Topic Maps component is a “P” Topic Maps engine which constitutes the Topic Maps infrastructure (internal dimension). The requirements for this dimension are

- Supply of reliable TMDM compliant Topic Maps engines
- Supply of plug-in architecture for popular WordPress or Movable Type blogging systems[8]
- Re-use of LAMP components such as MySQL or PostgreSQL RDBMS for topic maps storage
- Use of common web standards
- Use of widely adopted and thus widely available web technologies such as memcached[9] for caching strategies

Other Topic Maps components allow the exchange of topic maps and/or topic maps fragments (IO dimension). The requirements for this dimension are

- Supply of reliable XTM 2.0 serializers and deserializers
- Additional supply of leightweight Topic Maps formats such as JSON Topic Maps (JTM) [3] convenient for the use on the web
- Re-use of established syndication formats such as Atom for the exchange of topic maps fragments (which then forms a Topic Maps feed; Moore and Küster have described and implemented an approach [9])

In summary, on the one hand LAMP stack Topic Maps components should integrate easily into the stack (e.g. by re-using existing components) while on the other hand ISO/IEC 13250 Topic Maps standards such as TMDM must be supported. Further, web convenient Topic Maps technologies like JTM should be provided. Such are possibly maintained outside the standards body.

The Topic Mappish LAMP stack must not end in itself. Therefore benefits for content publishers (or providers) and consumers have to be evident.

[8] There is a WordPress plugin [12] which, however, only exports feeds in outdated XTM 1.0 format in a predefined way. In contrast, the Drupal Topic Map Module [6] is a more promising plugin approach due to its flexibility.

[9] http://www.danga.com/memcached/

4.1 Pull effect: Benefits for Publishers

The main objective for both perspectives, publishers (or providers) and consumers, is to establish more reliable semantics in content syndication processes. A popular and frequently applied way to organize content in blogs is to use *tags*. In fact, a collection of tags can be regarded as a one-level index. Once CMS are Topic Maps enabled, tags can be casted to topic names - and thus become part of a subject proxy (a topic). Further, properties such as subject identifiers can be appended to proxies easily. This enables publishers to increase the quality of their content (meta)description as well as content extension. Not only plain strings - maybe even ambiguous - could be provided but subject proxies which allow the establishment of more reliable semantics e.g. by providing subject identifiers. Basically, subject identifiers enable (a naive) referencing to subjects in a global context; referencing is a precondition in any semantic approach.

As declaration of associations is "built-in" in Topic Maps further services on content could be provided (see 4.2).

4.2 Pull effect: Benefits for Consumers

If publishers increase the quality of their content (meta)description this will have impact for content consumers. *Consumer* not only refers to individuals but also to technical systems fetching remote content by e.g. REST APIs. If such procedures are *tag-driven* (in order to further exploit the tag example) requests not only rely on plain strings but alternatively on subject identifiers which possibly occur multiple times for a subject proxy. On the one hand this helps to increase the precision of a request result as e.g. ambiguities caused by single-tag-only processing can be reduced. On the other hand consumers could benefit from other Topic Maps enabled services if e.g. related subject proxies (and - further - related content) are supplied.

5 Conclusions and Further Work

The LAMP stack underlays a significant area of the WWW. However the adoption of Topic Maps is minor in this context although the benefits of the application of the Topic Maps paradigm are evident. In order to be able to realize push effects an amount of LAMP stack Topic Maps components which build the Topic Maps infrastructure must be available. A more precise description of *amount* is hardly to provide. It must be emphasized that a sole supply of software can not launch a vast penetration process. The coherencies of successful penetrations of technology are quite complex. In general such processes involve a marketing dimension (in the broadest sense). Nevertheless, a crucial aspect is the effort of integration into the stack itself as well as the integration into existing software such as the numerous leightweight CMS e.g. by plugins. The lower the effort is the more powerful the push effects are able to evolve.

Another aspect in this context is the degree of convenience in the interaction with the Topic Maps engines. APIs such as TMAPI are “explicit” due to their OO alignment and therefore less efficient than “implicit” interfaces (e.g. Ruby Topic Maps [2] provides such features). A kind of shortcut or “implicit” API on top of e.g. TMAPI could further ease the integration of Topic Maps engines.

In this paper the requirements of a technical infrastructure are outlined. Providing an infrastructure is a foundation. In order to be able to realize benefits the Topic Maps paradigm must be applied to data. Thus, efficient methodologies are necessary whose elaboration must be the focus of further work (see [9] for an approach).

5.1 Vision

In [8] Maicher has proposed a bottom-up approach for global integration of distributed subject proxies. As a design recommendation Maicher determines the assignment of two (prominent) subject identifiers. This could be regarded as a quality dimension. Maicher's approach implicitly relies on a certain amount (*critical mass*) of subject identifiers. A Topic Mappish LAMP stack could help just to approximate to this state. The vision expressed in this paper is to enable the millions of leightweight CMS to become *subject identifier providers* as a side effect to the primary implementation objectives of Topic Maps components motivated by the described benefits. This quantity dimension forms a complement to the quality dimension. Such infrastructure based on subject identifiers then can be exploited for global integration and federation approaches.

References

1. Ahmed, Kal; Garshol, Lars-Marius; Grønmo, Geir-Ove; Lischke, Stefan; Heuer, Lars; Moore, Graham: *TMAPI - Common Topic Map Application Programming Interface.* http://www.tmapi.org/
2. Bock, Benjamin: *Ruby Topic Maps*. http://rtm.topicmapslab.de/
3. Cerny, Robert: *JSON Topic Maps 1.0*. http://www.cerny-online.com/jtm/1.0/
4. Conroy, John: *Results: Most Popular CMS in Technorati's Top 100.* http://www.cmswire.com/cms/micro-cms/results-most-popular-cms-in-technoratis-top-100-002198.php
5. Holje, Espen; Schmidt, Johannes: *PHPTMAPI*. http://phptmapi.sourceforge.net/
6. Hunting, Sam: *Drupal Topic Map Module*. http://www.universalpantograph.com/drupal-topic-map-module
7. Kindsmüller, Martin Christof; Milz, Jan; Schmidt, Johannes: *Instant Online Communities as a means to foster conferences.* In: Ozok, A. Ant; Zaphiris, Panayiotis (eds.) Online

Communities and Social Computing. LNCS, vol. 5621/2009, pp. 62-71. Springer, Heidelberg (2009). ISBN 978-3-642-02773-4

8. Maicher, Lutz: *The Impact of Semantic Handshakes*. In: Maicher, Lutz; Sigel, Alexander; Garshol, Lars M. (eds.) Leveraging the Semantics of Topic Maps - Second International Conference on Topic Maps Research and Applications, TMRA 2006. LNCS, vol. 4438, pp. 140-151. Springer, Heidelberg (2006). ISBN 978-3-540-71944-1
9. Moore, Graham: *Implementing Topic Maps Content Management Integration using Atom and Live Mesh.* http://www.networkedplanet.com/ontopic/2009/07/implementing_topic_maps_conten.html
10. Moore, Graham; Küster Marc Wilhelm: *Protocol for the Syndication of Semantic Descriptions*. In: Maicher, Lutz; Garshol, Lars M. (eds.) Subject-centric Computing - Fourth International Conference on Topic Maps Research and Applications, TMRA 2008. Leipziger Beiträge zur Informatik, vol. 12, pp. 225-234. LIV, Leipzig (2008). ISBN 987-3-941152-05-2
11. Netcraft.com: *July 2009 Web Server Survey.* http://news.netcraft.com/archives/web_server_survey.html
12. Pettersen, Trond: *Wordpress Topic Maps (WP2TM) Plugin*. http://www.topicobserver.com/blog/wordpress-topic-maps-wp2tm-plugin/
13. Schmidt, Johannes: *QuaaxTM*. http://quaaxtm.sourceforge.net/
14. Schmidt, Johannes; Stephan, Andreas: *PHPTMAPI poster.* http://www.informatik.uni-leipzig.de/~tmra/2006/contributions/tmra2006-PHPTMAPI.png
15. Securityspace.com: *Apache Module Report*. http://www.securityspace.com/s_survey/data/man.200706/apachemods.html. July 1st, 2007
16. Technorati.com: *State of the Blogosphere / 2008*. http://technorati.com/blogging/state-of-the-blogosphere/
17. Wikipedia: *Apache Tomcat.* http://de.wikipedia.org/wiki/Apache_Tomcat