MetaMorPic:
Self-contained photo archival and presentation

Tomas Cerny and Michael J. Donahoo

Abstract Cost associated with the maintenance and scale of custom services presents one of the most significant barriers to entry for information providers. One solution proposes pushing computation into the cloud where providers like Amazon maintain a scalable, world-wide platform for virtualization. While this avoids certain hardware maintenance costs, service providers must still maintain clusters of virtual machines. To address this, contextual services like Flickr provide a complete solution; unfortunately, this limits information providers to the available services and look-and-feel. We propose a compromise solution that combines existing services and applications. Such a solution decreases development and maintenance costs by providing standardized services with third-party maintenance, while allowing customizable functionality and look-and-feel. We present a specific example of this type of blended solution in our MetaMorPic system, which provides photo archiving and presentation capabilities using third-party software and services.

1 Introduction

Public information systems are part of the critical infrastructure for distributing information. Much of this information is distributed as text, which fits well with Semantic Web [1]. The idea of Semantic Web is to provide standardized rules to structured information so that it is easy to find and process by computers [1]. More and more we also receive information in nontextual forms, including voice messages, images, videos, etc. Clearly, such data are also semantic instruments [2] [3]. Existing

Tomas Cerny
Department of Computer Science and Engineering, Czech Technical University, Charles square 13, 121 35 Prague 2, CZ, e-mail: tomas.cerny@fel.cvut.cz

Michael J. Donahoo
Department of Computer Science, Baylor University, P.O. Box 97356, 76798-7356 Waco, TX, US, e-mail: jeff_donahoo@baylor.edu

technology allows us to associate metadata with images, etc. that extends captured information. Many systems allow the gathering of metadata associated with images [4] [5] [17] [18] [19]. Such metadata should be application independent, which allows us to use images for archival purposes because we can access the metadata many years later without a need of a specific presenting application. The image format JPEG, for example, allows us to keep all the metadata within the image [21], but its integration within systems is partial [17] [18] [19] requiring additional information to be held apart from the image. In addition to its descriptive purpose, multimedia metadata provides searchable, machine-readable access to images similar to that of Semantic Web.

In this paper, we propose a self-contained photo archival and presentation system capable of metadata gathering. Such a system is capable of capturing events, their attendees, and extensive event descriptions in textual form. We store metadata within the image files to avoid application dependence. The images can be seen after many years using different presenting tools, and the semantics can be fully reproduced from just an image and its metadata. We evaluate existing, commonly-used services and applications and provide implementations compatible with existing third-party vendors. We develop the application by combining existing services [16] [19] and presentation tools [28] [29] [30] through a light-weight data-interchange format [24]. The application was evaluated in production for a large event, the 2010 ICPC World Finals in China, with nearly 1,000 local event participants and many more remote users, generating significant user load. The evaluation event is popular world wide, accessed by users and media [33] form various countries.

The paper is organized as follows. In Section 2, we introduce existing solutions and their drawbacks. The proposal, strategy for the development, and results are described in Section 3. An example of the use is described in Section 4, followed by conclusions in Section 5.

2 Existing Solutions and Drawbacks

Information systems are very popular these days, and multiple vendors provide free, open-source applications and web services. For our work, we look specifically at media presentation and archiving. Existing services provide a varying set of both features and interfaces. We base our media presentations evaluation on its capability to categorize media for selection and how it allows display of metadata to the user. For archiving, we look at storing the media with its metadata, how we supply the metadata and what are the requirements for the installation, maintenance and backup.

We evaluated many static media galleries [28] [29] [30] all of which provide attractive and rich user interfaces that can be customized. Various vendors [16] [18] [19] [20] provide the capability of sharing media where users can upload and organize a gallery, but these systems provide a vendor-specific appearance. We also consider Semantic Web approaches for our goals. In addition to existing system, there is the option of developing our own system. In this section, we investigate the pros and cons of various options for media presentation and archiving.

2.1 Traditional Development Approaches

The traditional approach of custom application development forces us to manage data persistence ourselves, typically in a relational database. While the developer has the freedom to specify any data schema, systems dealing with photo archiving typically produce similar schemas. Custom persistence solutions also include the onerous task of conducting, storing, and managing backups. Once persistence and backup is in place, the developer must start to design model-view-controller (MVC) [6] type application. Modern enterprise frameworks rely on object oriented design (OOD) [7] and mapping the database schema to the object domain model using some object-relational mapping framework (ORM) [34]. This mapping adds repetitive effort to development because it parallels data schema development. Having ORM in place allows us to define various queries to obtain data we wish to show to the application user. We must deal with functions for the application manager concerned with photograph management. To provide application users with suitable presentation, we must build some display view, most probably for a web browser. Similarly we can provide a potentially faster solution using a standalone application capable of a dialog with the controller over web services.

While this approach provides design freedom, we must deal with an expensive and error-prone development process, test modules, apply backups and spend time on application maintenance. One complicated issue involves application scalability where we might be required to invest in hardware resources without knowing demand. Developing a new application is very time consuming, and we may not get the features and scalability that existing systems already provide because of time constraints and resource limitations.

2.2 Existing Semantic Approaches

In Section 1, we suggested a relationship with Semantic Web. The research in techniques of image content annotation has long history [9] [10] [11]. We consider PhotoStaff [4] [5], a tool for annotating images on the Semantic Web. Semantic Web uses ontology languages like Resource Description Framework (RDF) [31] and Web Ontology Language (OWL) [32], but research does not address the images with their metadata. PhotoStaff uses metadata embedded in images, extract them and encodes them in RDF/XML. The tool then takes advantage of of the built infrastructure to allow search and navigate.

Although this tool is capable of advanced searches and advantages of Semantic Web, we also find several drawbacks. First, if we rename or shuffle images, we lose the metadata information, although we can partially recover from the image embedded metadata. Second, this approach binds our systems to a specific tool. Third, we do not receive any annotation automation. Finally, this system lacks the features and stability to be used in production [16] [19]. To address these problems, we propose embedding all the metadata into an image, which helps to avoid information loss. From a functional decomposition perspective, we get advantages if the system or data are broken down into units of behavior or information [12] [13] [14]. A solution based on this approach would naturally store image metadata in the image itself, and if an ontology is built, it should be build on the top of the low-level metadata.

2.3 Media Sharing Systems

Media sharing is rapidly growing in popularity on social networks or web galleries to provide access to a variety of consumers. The last few years have witnessed the rapid expansion of the social networking site Facebook [18], and we have seen cutting edge features brought by Google's Picasa [17] and its competitor Yahoo's Flickr [19]. These media sharing applications provide options to manage and organize media and metadata annotation automation. They allow image resizing, cropping, and other modification. These systems utilize media metadata [21] [22] [23] for media specific information (GPS, camera parameters and settings) and for storing a user defined information (keywords). The latest features include face recognition and detection where the application detects similar faces and we tag the set of faces by the person name. This allows us to see all media related to a particular person and detect him/her in other media. These systems provide cutting edge features, and we expect this trend to continue. The advantage of their use is that we receive all the nice features already developed (patented) and the benefit of storage, but we need not be concerned with the database definition, maintenance, and backups. Drawbacks include restricted capability of structuring the gallery, the vendor specific appearance and its binding to the vendor. Existing media sharing systems hide their internal implementation from the developer. The critical services for managing metadata exist as well as the sufficient search capabilities. The key thing for a developer is the production level and scalability.

2.4 Existing Media Gallery Presenters

Plenty of presentation galleries [28] [29] [30] can be found online, where designers provide a sleek user interface attractive to users. These galleries are often available for free and provide just the presentation without connection to any archive. Static media galleries provide just a part of the functionality we need. These are also easy to customize or modify to fit our company expectation providing specific information and references. Many modern elements that make the presentation attractive for the user utilize JavaScript (JS); consequently, most of the galleries utilize JS in either its pure form or using a JS framework [25] [26] [27]. JS not only enable a desktop-like interface but also decreases the server load by implementing a part of the presentation logic at the client side. From our perspective, we cannot solely use a static gallery, but we may consider its use in a connection with some photo archive.

3 MetaMorPic Proposal and Implementation

Previously, we introduced existing approaches and applications, summarizing their properties. Here we propose a solution to build a self-contained photo archival and presentation system, MetaMorPic.

The system should archive photographs with metadata describing the context such as who is on the picture, where was it taken, who is the author, what camera was used, what event it contains, or if there is a specific delegation or group captured by the picture. JPEG EXIF [21] and IPTC [22] formats allow us to store metadata directly in the image file. The presentation should have access to entire photo archive as well as search capabilities to select specific categories of people or events. The system should not replicate any of the information for the archival and presenter.

Our approach utilizes the services of existing archival and image presentation services by simply providing the necessary glue code to compose these services. The combination of the existing systems presents many benefits. The main issue we face is that existing applications work on their own; we need to find a connection mechanism to connect them. Some of the systems we introduced in relation to archival capabilities [16] [18] [19] provide an API that allows other programs to integrate their functionality, although often only a functionality subset of the application is available. We use these systems for photo archival and management, and via their API provide the images and metadata to the presentation subsystem. Archival services store and support basic query operations on all galleries. The galleries should be nalurally structured to smaller sets of photos to perform the queries on. Presentation is covered by static media galleries [28] [29] [30], but we need to modify their static behavior.

By using the existing archival and presentations systems, we reduced the development efforts to the connection layer. The connection layer may either be implemented as a server-side application (i.e., using Java, PHP, .NET, etc.) or on the client-side (i.e., using JavaScript, Flash, etc.) [8]. The first option provides simplification for the development but adds the requirement to host the connection layer application at some server, requiring both maintenance and hardware costs. The client-side connection layer might be more challenging from the development perspective but simplifies the hosting requirements to a static web location with no need of a server-side interpreter engine. The second solution has advantages for the practical use.

Among the presentation applications, we choose Galleriffic [28], which is a jQuery plug-in for rendering rich, fast-performing photo galleries. It provides suffi-

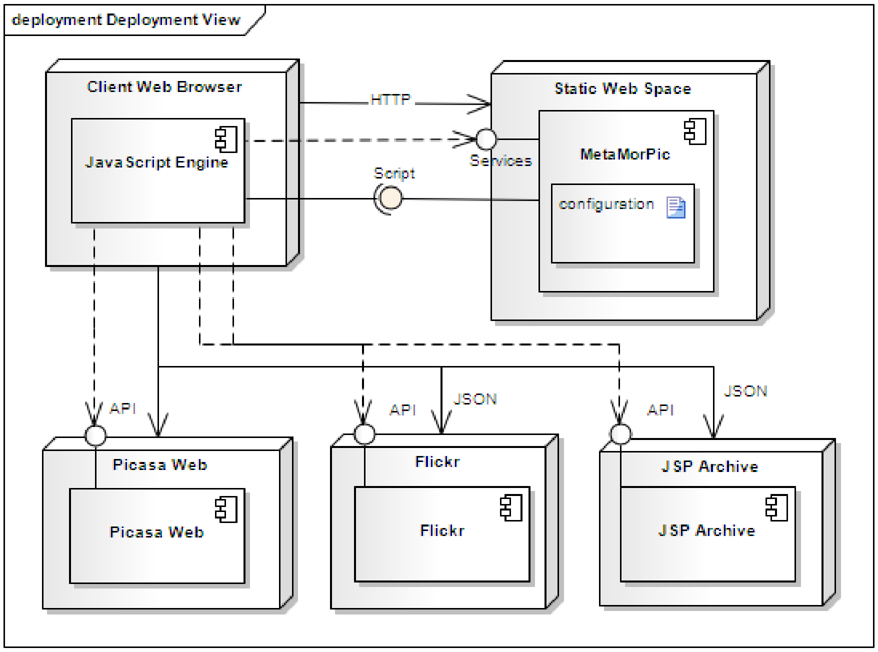


Fig. 1 Deployment of MetaMorPic

cient documentation for its modification and allows us to customize the appearance. Next we need to connect the client-side presentation to the archive. Specifically, we want a lightweight, data-interchange format that is supported by both client and server side so we adopt JavaScript Object Notation (JSON) [24]. This notation is well supported by the archive API's and jQuery.

For the user domain, there is a need of media categorization. Such a generic categorization is not provided by any of the archival. The solution is to define a metadata name-space for the structure we expect and use archival search capability to search over such an name-space. The archival embeds the metadata in each media [21] [22] [23], which has the advantage that exports of the media contains the metadata. If we export all our galleries to another system, the metadata will be preserved. Currently, most systems do not include facial recognition data in the image metadata. In addition, most image archival services do not share this information. But since we can read this position out of the original archival application, we provide a Perl script[1] for porting external information to image metadata. This allows us to preserve all metadata and provide it also to the presentation subsystem. Second we encode the facial information in such a way that indexing system cannot detect this information by indexing our application to provide privacy. Besides the Picasa Web and Flickr compatibility, we also built a JSP-based archive capable of media storage and JSON

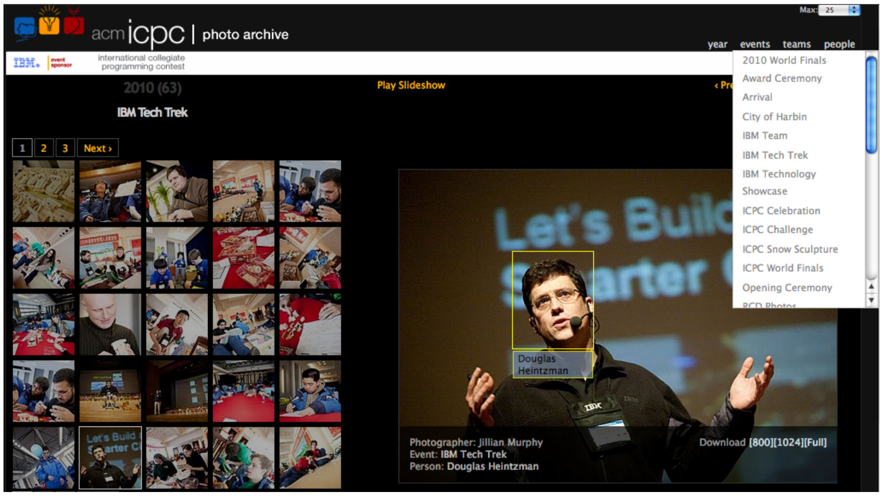
[1] In addition we work on a Java based Picasa desktop plugin which would allow to simplify the metadata port.

services. Since our media contains all metadata, we can switch the archives with no compatibility issues. The application structure is shown by Fig. 1.

MetaMorPic [15] is an application built on the top of existing infrastructure provided by third-parties. We reduce the development effort to the connection layer to make the presentation provided by Galleriffic compatible with media archives Picasa Web, Flickr, or our own custom JSP Archive. The MetaMorPic web application contains all of its logic at the client-side, which speeds up the application reaction time. The logic is built to provide the user with pay-as-you-go experience so only a small media page is provided by prefetching and the rest of the media pages are loaded on demand. Every modern web browser with enabled JavaScript can display our application. The scalability of the application is moved to the archival provider (Picasa Web, Flickr), which is generally better than what individuals can provide. This pushes the maintenance and backup responsibility to the archival provider. Using MetaMorPic does not require installation of any specialized software; instead, it simply needs a static web location for the MetaMorPic client-side code and a Picasa Web/Flickr account. Managers of Picasa Web may also gain from the use of standalone Picasa photo editing software [17] that allows the user to manage and organize the content and synchronize it with the web archive. MetaMorPic is shown in Fig. 2.

4 Example Use

To evaluate MetaMorPic, we deployed it at the 2010 ACM-ICPC World Finals in Harbin, China. This competition hosted nearly 1,000 participants with five pho-



Photographer: Jillian Murphy
Event: IBM Tech Trek
Person: Douglas Heintzman

Fig. 2 MetaMorPic in producation

tographers and numerous events [15] [33]. For the past few years, the media team established a process for the competition media archiving and presentation. A proprietary system was used, and the management work was tedious, mostly because every face had to be identified manually, images resized and the system did not support all management features.

Every photograph goes normally through correction and metadata tagging. We keep camera information (flash, lens, camera type, GPS), author and classification of the image. For this event, we archive who is in the image (along with face position), what event and, if applicable, which university team. With MetaMorPic, we automate the process and greatly simplify the work of the media team. The benefit is that the media team can use the Picasa's desktop photo editing software. A manager can simply tag one face, and the Picasa desktop auto-recognizes the person in the galleries.

A standard web server in Texas at Baylor University currently serves the static web page implementing the presentation aspect of MetaMorPic for ACM-ICPC. Google's cloud provides the web services and hosts the images. We have evaluated the application responsiveness and interaction on multiple web browsers in the USA, Canada, China, and the Czech Republic.

5 Conclusion

To build a production-level information system based on images and enhanced with metadata is a challenging task. Recent research in the area is moving toward the use of Semantic Web, but there are difficulties with respect to development and archival. Our proposal provides a modern approach to application development, which leverages recent advances in services while minimizing complexity and cost. Rather than building a brand new application, we propose using services provided by a third party and focus only on specifics we expect from the application. We have proposed and implemented a self-contained photo archival and presentation application called MetaMorPic. This application combines multiple services to provide many advantages of the final system and a great user experience in form of client-side web application. Our application is build on the top of the jQuery plug-in Gallerific for presentation, readily available services for archiving and searching such as Picasa Web and Flickr, and a metadata managing desktop application (i.e., Picasa). Our application does not require any server-side interpreting, just static web space. Finally, we evaluated the application in production with positive feedback from its users and managers.

Potential future work includes building a MetaMorPic plugin to Picasa desktop to simplify metadata propagation. For advertisement of the galleries, we plan to provide a virtual tour for selected photos from the competition. As there exists multiple conferences/competitions for our target organization, we plan to extend MetaMorPic functionality so it would dynamically load a gallery based on the URL GET parameter specifying the gallery name. This allows keeping a central access point and naturally scaling in terms of the gallery management. In case of the ACM-ICPC this allows to put the responsibility of archiving on the regional contest managers and keep the presentation accessible from the main web page. Finally, we plan to mesh it with other network services such as Facebook, blogs, Google maps, etc.

Acknowledgements We would like to thank Aaron Kelley for the JSP Archive development and Hans Domjan, the ICPC Photo Management Director, for his feedback and suggestions. Our thanks goes to the ACM-ICPC for supporting this project.

References

1. Phil Cross, Libby Miller, Sean Palmer (2001), Using RDF to annotate the (semantic) web, http://ilrt.org/discovery/2001/07/KCAP-ann
2. Confucius
http://en.wikipedia.org/wiki/A_picture_is_worth_a_thousand_words
3. Jun Zhao, Graham Klyne, and David Shotton (2008), Building a Semantic Web Image Repository for Biological Research Images, Department of Zoology, University of Oxford, 5th European Semantic Web Conference
4. Christian Halaschek-Wiener, Jennifer Golbeck, Andrew Schain, Michael Grove, Bijan Parsia, James A. Hendler (November 2005), PhotoStuff An Image Annotation Tool for the Semantic Web, Yolanda Gil, Enrico Motta, V. Richard Benjamins, Mark A. Musen (Ed.), Posters 4th International Semantic Web Conference, Galway, Ireland.
5. Christian Halaschek-Wiener, Jennifer Golbeck, Andrew Schain, Michael Grove, Bijan Parsia, and Jim Hendler (2006), Annotation and Provenance Tracking in Semantic Web Photo Libraries, University of Maryland, Springer-Verlag Berlin Heidelberg
6. Gamma Erich; Richard Helm, Ralph Johnson, John M. Vlissides (1995), Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley. pp. 395. ISBN 0201633612.
7. Larman, C. (2004), Applying UML and Patterns: an Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd Edition). Prentice Hall PTR. ISBN:0131489062.
8. Fowler, M. (2002), Patterns of Enterprise Application Architecture. Addison-Wesley Longman Publishing Co., Inc. ISBN:0321127420
9. Dupplaw, D., Dasmahapatra, S., Hu, B., Lewis, P., and Shadbolt, N. Multimedia (November 2004), Distributed Knowledge Management in MIAKT. ISWC 2004 Workshop on Knowledge Markup and Semantic Annotation. Hiroshima, Japan
10. Addis, M., Boniface, M., Goodall, S., Grimwood, P., Kim, S., Lewis, P., Martinez, K. and Stevenson (2003), A. SCULPTEUR: Towards a New Paradigm for Multimedia Museum Information Handling. Second International Semantic Web Conference 582-596
11. Hollink, L., Schreiber, G., Wielemaker J., and Wielinga. B. (2003), Semantic Annotation of Image Collections. Knowledge Capture - Knowledge Markup & Semantic Annotation Workshop
12. Parnas D. L. (1978), Designing Software for Extension and Contraction, in proc. 3rd International Conference on Software Engineering, pp. 264-277.
13. Parnas D. L. (1974), On a 'Buzzword': Hierarchical Structure, in proc. IFIP Congress 74, pp. 336-339.
14. Parnas D. L. (1972), On the Criteria to be Used in decomposing Systems into Modules, in Communications of the ACM, vol. 15(2).
15. MetaMorPic: Self-contained photo archival and presentation, (Cited 2010)
http://cm.baylor.edu/gallery/MetaMorPic.html
16. Google Picasa Web Album - photo sharing application, (Cited 2010)
http://picasaweb.google.com

17. Google Picasa - photo management application, (Cited 2010)
http://picasa.google.com
18. Facebook - social network, (Cited 2010)
http://facebook.com
19. Flickr - online photo management and sharing application, (Cited 2010)
http://flickr.com
20. Kodak - photo sharing application, (Cited 2010)
www.kodakgallery.com
21. Exchangeable Image File Format (EXIF), (Cited 2010)
http://www.exif.org
22. IPTC Photo Metadata, (Cited 2010)
http://www.iptc.org/cms/site/index.html?channel=CH0089
23. Extensible Metadata Platform (XMP), (Cited 2010)
http://www.adobe.com/products/xmp
24. JavaScript Object Notation (JSON) - lightweight data-interchange format, (Cited 2010)
http://www.json.org
25. Scriptaculous - JavaScript library, (Cited 2010)
http://script.aculo.us/
26. jQuery - JavaScript library, (Cited 2010)
http://jquery.com
27. Prototype - JavaScript framework, (Cited 2010)
http://www.prototypejs.org/
28. Galleriffic - jQuery plugin, (Cited 2010)
http://www.twospy.com/galleriffic
29. Galleria - JavaScript image gallery, (Cited 2010)
monc.se/galleria/demo/demo_01.htm
30. Imago - JavaScript image gallery, (Cited 2010)
http://imago.codeboje.de
31. Resource Description Framework (RDF), (Cited 2010)
http://www.w3.org/RDF
32. OWL Web Ontology Language, (Cited 2010)
http://www.w3.org/TR/owl-features
33. NYTimes - article reference to example event, source MetaMorPic, (Cited 2010)
http://www.nytimes.com/2010/02/22/technology/22cyber.html?scp=1&sq=jillian%20murphy
34. Java Persistence API, (Cited 2010)
http://java.sun.com/developer/technicalArticles/J2EE/jpa