MyDeepWeb: An Integration Service for your OWN Deep Web Data

Sherif Sakr Anna Liu
NICTA and University of New South Wales, Sydney, Australia
first.last@nicta.com.au

Abstract

Current Web users usually have their own files, work documents, communications and personal contacts distributed in the storage systems of many widely-used Internet services (e.g. Google Docs, Gmail, Facebook, Zoho). Therefore, they face the challenge of being not able to have an integrated view for their related data objects (e.g. mails, pics, docs, contacts). Recently, most of the major Internet services provide standard APIs that allow developing software applications that can read and write data from their underlying data store after providing the credential access information of registered accounts. The MyDeepWeb system is designed to let the Web users interact with their Internet services normally while, behind the scene, the information of their objects will be extracted, consolidated, linked and then populated into a single private cloud-based data store where the user can have integrated access to their data objects from anywhere through multiple devices.

1 Introduction

We are currently witnessing the great impact of the World Wide Web in rapidly transforming industries, business models and our work culture with no signs of slowing down. The emergence of many popular Internet services such as: Web-based email, multimedia sharing, collaborative tools and social networks plus the increased worldwide availability of high-speed connectivity have accelerated the trend of moving the computing and data storage from PC-like clients to large Internet services. The main advantage of these services is that they allow the users to access their own data from anywhere in the world through multiple devices. In practice, current Web users usually have their own files, work documents, communications and personal contacts distributed in the storage systems of many widely-used Internet services (e.g. Gmail, Google docs, Google Calendar, Facebook, Linkedin, Myspace, Flickr, Twitter, Zoho).

The amount of data that users are storing and accessing in these services is growing massively. Given such a situation, Web users face the challenge of not being able to have an integrated view for their related objects (e.g. mails, pics, docs, contacts) that are distributed in the repository of different systems. To illustrate, let us assume that Peter would like to see his recent interactions with John. The answer of such request may refer to some email messages in the Gmail account, some communications in the Linkedin inbox, recently uploaded photos in Myspace Album in addition to collaborative documents which are edited by Peter and John in the repository of Google docs. Currently, answering any of these sample requests and finding the required objects and the relationships between them can not be, currently, achieved by any single system. Neither traditional web search engines (e.g. Google, Yahoo!) nor desktop search tools (e.g. Google Desktop Search) can achieve that goal. The main reason behind this is that the required objects reside in online deep and hidden repositories that such systems are not authorized to access. Web search engines are only able to crawl and index the surface web while desktop search tools can only access data and files which are stored in the local file systems.

2 MyDeepWeb: System Design

Figure 1 depicts an overview of the MyDeepWeb system which is designed to enable the Web users to interact with their Internet services normally while, behind the scene, the information of their objects will be extracted, consolidated, linked and then populated into a single private cloud-based data store where the user can have integrated access to their data objects from anywhere in the world through multiple devices. The system makes use of a significant feature which is recently introduced by almost all of the major Internet services where they provide standard APIs that allow developing software applications that can read and write data from their services after providing the credential access information (usually a username an password) of registered accounts. Using this functionality, the main idea of our system is to let the Web users interact with their Internet services normally while behind the scene the metadata and the pointers (URIs) of the user distributed objects will be extracted, consolidated (after getting the required authorization) and then populated into a single private cloud-
based data store (e.g. Amazon SimpleDB, Google Megastore, Microsoft SQL Azure). Using these extracted data, we can provide integrated answers for user requests that target their own objects which are distributed in the repositories of different online services. Moreover, storing the data in a cloud-based data store ensures a high availability access to the stored data which is guaranteed by the SLA of these services (usually 99.99%). In addition, it allows any type of software applications (e.g. Web-based application, Mobile application, Desktop application) to have an access to the target data if they are able to establish the connection to this cloud-based data store. For example, in our MyDeepWeb system, we implemented three different search interfaces: Web-based\(^1\), mobile phone-based and a desktop-based windows application. The system also provides an API interface that allows reusing the system functionalities by other software applications.

The internals of the MyDeepWeb system include the following main components:

1) **Object Extractors**: The object extraction layer is designed in a very flexible fashion where a tailored crawler for each Internet service is implemented (using the service supported API) as an independent plugin. Thus, integrating any additional service simply requires utilizing its available API functionalities to implement an object Extraction plugin. The extracted objects from the repositories of the different service repositories are naturally heterogenous, i.e belongs to different object types (e.g. person, mail, document, calendar appointment). Therefore, each object has its own schema (metadata information). Due to this schema heterogeneity, we rely on Amazon SimpleDB\(^2\), a cloud-based key-value data store, as an efficient and flexible solution in this context. Additionally, we use the extracted schema information of the different object types in building special indexes that are used to provide enhanced results for the user’s search requests.

2) **Object Matcher**: Entity resolution is a well-know problem in data integration systems [1]. In practice, information about the same entity may be distributed across different systems. Therefore, different extracted entities may refer to the same real world object and thus they need to be re-linked together. For example, Peter may have John in his contact list of different services (e.g. Facebook, LinkedIn, Twitter, Gmail). However, these different contacts need to be treated as a single object as they are all refer to the same person. In our implementation, we used the flexible framework for mapping-based object matching, MOMA [4], to achieve this goal.

3) **Object Linker**: The extracted object from the repositories of different services can be usually related to each other through different types of relationships. Automatic and complete discovery of these relationships at once is a very challenging task. Therefore, we are going to follow a pay-as-you-go discovery philosophy [2] where these relationships can be gradually identified and discovered in multiple ways such as: a) Explicit indication by the end-user to link existing objects with specific type of relationships. b) Defining some heuristic rules that can suggest the potential existence of specific types of relationships between existing objects (e.g. Personal contacts with the similar email address are to be identify as potential work colleagues).

### 3 Conclusion

We described the design of the MyDeepWeb system which provides the Web users with an integration service for their own Deep Web Data. In practice, the amount of data for each user is growing massively. Therefore, we are currently in the process of designing a more convenient browsing interface for the user objects and the relationships between them using the Mind Maps techniques [3].

**Acknowledgment**

We would like to thank Hui Feng for his work on the prototype implementation.

**References**


---

\(^{1}\)http://www.mydeepwebau.com/

\(^{2}\)http://aws.amazon.com/simpledb/