Thymeflow, A Personal Knowledge Base with Spatio-temporal Data

David Montoya
Engie Ineo & ENS Cachan & INRIA
david.montoya@inria.fr

Thomas Pellissier Tanon
ENS Lyon
thomas.tanon@ens-lyon.fr

Serge Abiteboul
INRIA & ENS Cachan
serge.abiteboul@inria.fr

Fabian M. Suchanek
Télécom ParisTech
suchanek@enst.fr

ABSTRACT

The typical Internet user has data spread over several devices and across several online systems. We demonstrate an open-source system for integrating user's data from different sources into a single Knowledge Base. Our system integrates data of different kinds into a coherent whole, starting with email messages, calendar, contacts, and location history. It is able to detect event periods in the user's location data and align them with calendar events. We will demonstrate how to query the system within and across different dimensions, and perform analytics over emails, events, and locations.

Keywords

personal information; data integration; querying; open-source

1. INTRODUCTION

Today, typical Internet users have their data spread over several devices and services. This includes emails, contact lists, calendars, location histories, and many other types of data. However, commercial systems often function as data traps, where it is easy to check in information and difficult to query it. This problem becomes all the more important as more and more of our lives happens in the digital sphere.

With this paper, we propose to demonstrate a fully functional personal knowledge management system, called Thymeflow. Our system integrates personal information from different sources into a single knowledge base (KB). The system runs locally on the users' machine, and thus gives them complete control over their data. Thymeflow replicates data from outside services (such as email, calendar, contacts, location services, etc.), and thus acts as a digital home for personal data. This provides users with a high-level global view of that data, which they can use for querying and analysis.

Our demonstration will illustrate the features of Thymeflow, the connection to external services, and its capacity to answer to questions such as "Where did I have lunch with Alice last week?". We will present the architecture of Thymeflow and illustrate its main functionalities. In particular, we will show how an incremental change in a data source leads to changes in the KB and its enrichment. We will also illustrate the management of location data (such as GPS traces). We believe that such location data becomes useful only if it is semantically enriched with events and people in the user's personal space – which is what Thymeflow achieves. We will demo alignments based on time (a meeting in calendar and a GPS location) and on space (an address in contacts and a GPS location). Finally, we released the code under an open-source software license[1].

This paper is structured as follows: Section 2 describes our system, its model, data sources and knowledge enrichment processes. Section 3 discusses our demonstration setting and Section 4 the related work.

2. THE SYSTEM

Our system is a Scala program that the user installs locally. The user provides the system with a list of data sources (such as email accounts, calendars, or address books), together with authorizations to access them (such as tokens or passwords). The system accesses the data sources (as the user would), and pulls in the data. All code runs locally on the user's machine. Thus, the user remains in complete control of her data. The system uses adapters to access the sources, and to transform their data into RDF. We store the data in a Sesame based triple store [2]. Since the KB is persistent, we can restart the system at any time without losing information.

Architecture. One of the main challenges in the creation of a personal KB is that data sources may change, and these updates have to be reflected in the KB. To address this dynamics, our system uses software modules called *loaders* and *enrichers*. Figure 1 shows the loaders $L_1, \ldots, L_n$ on the left, and the enrichers $E_1, \ldots, E_p$ in the center. Loaders are responsible for accessing the data sources. Enrichers are responsible for inferring new statements, such as alignments between entities obtained by entity resolution. Loaders are triggered by updates in the data sources (e.g., calendar entries) and insertion of new pieces of information in the KB triggers the execution of a pipeline of enricher modules, as shown in Figure 1.

[1] https://github.com/thymeflow/thymeflow

CIKM'16 October 24-28, 2016, Indianapolis, IN, USA

ACM ISBN 978-1-4503-4073-1/16/10.

DOI: http://dx.doi.org/10.1145/2983323.2983337

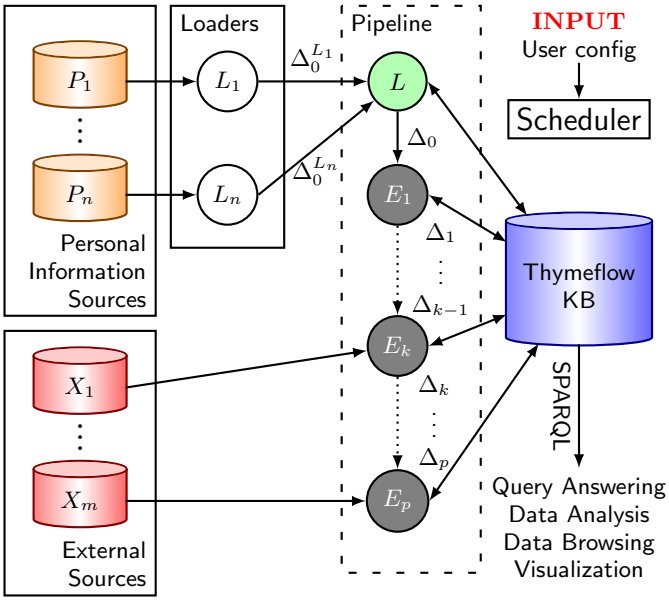

Figure 1: System architecture

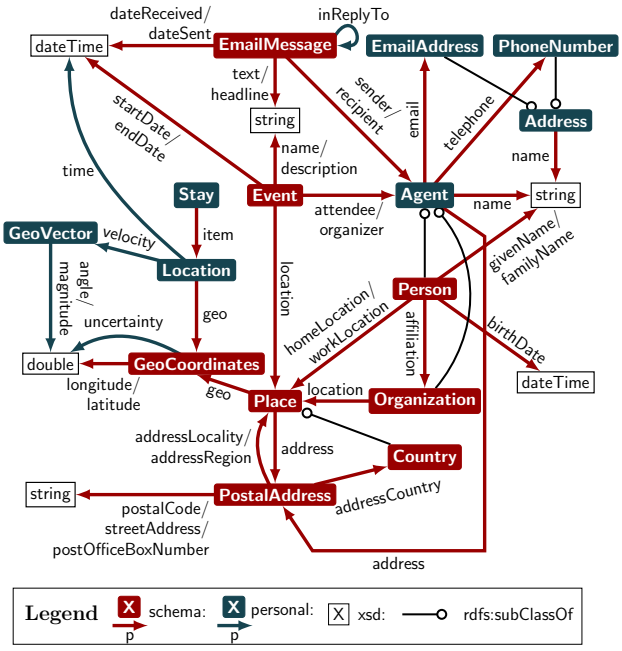

Figure 2: Personal Data Model

Data model. We use the RDF standard [7] for knowledge representation inside of the knowledge base. Figure 2 illustrates a part of our schema. Nodes represent classes, rounded ones are non-literal classes, and an arrow edge with label p from X to Y means that the property p links instances of type X to instances of type Y.

Loading. The loader modules are responsible for retrieving new data from a data source. We have designed adapters that transform items from the data source into our schema. For each data source that has been updated, the adapter for that particular source transforms the source updates since last synchronization into a Δ, a set of insertions/deletions in RDF. We have implemented loaders for the following protocols:

- CardDav [8] provides contacts encoded in the vCard format [15]. They are represented as instances of type `personal:Agent`.
- CalDav [9] provides calendar events and todos in the iCalendar format [10]. We consider only events for now and we extract a new `schema:Event` for each of them. The location of an event is typically given as a postal address, and we will discuss later how to associate it to geo-coordinates and richer place semantics.
- IMAP [5] provides emails in the RFC 822 format [6]. We represent senders and receivers as instances of `personal:Agent`. We will discuss later how to map these agents to the ones extracted from other sources.
- Location data in the Google Location History format. This format provides the user's location history as points with a timestamp and geo coordinates, which we represent as instances of `personal:Location`. Competing formats for location data include GPX and KML, but no standardized protocol exists for managing a location history (i.e. with synchronization capabilities).

Enrichers. After a loading phase, the enricher modules perform inference tasks such as entity resolution, event geolocation, and other knowledge enrichment tasks. Such an enricher takes as input the current state of the KB, and some collection Δ_i of changes that have happened recently. It computes a new collection Δ_{i+1} of enrichments. Intuitively, this allows reacting to changes of a data source. When some Δ_0 is detected (typically by some adapter), the system runs a pipeline of enrichers to take these new changes into consideration. For instance, when a new entry is entered in the calendar with an address, a geocoding enricher is called to attempt to locate it. Another enricher will later try to match it with a position in the location history. We have for instance implemented the following enrichers:

- An enricher that detects that two `personal:Agents` are about the same real world agent. Agents are matched based on similar attributes such as email addresses or names. The enricher can thus link emails to contacts.
- An enricher that analyzes the location history to detect places where the user stayed for a prolonged period of time. The enricher uses an algorithm similar to the time-based clustering algorithm in [13].
- An enricher that matches the stays extracted by the previous enricher with events in the calendar. This enricher works by detecting shared periods of time.
- A geocoder that links each calendar event with one or more real world places (street addresses, landmarks, etc.). This module uses both the location stated by the calendar and the coordinates the previous enricher associates to events.

3. DEMONSTRATION SETTING

Our system is equipped with a SPARQL 1.1 compliant engine [12] with optional full-text search capabilities based on Apache Lucene. This allows users to launch SPARQL queries on their personal data. Since the KB unites different data sources, queries span different types of data.

During the demonstration we will present the system as seen by a particular user, say Mr. S. Our user will first

launch the system, and connect it to his Google account (Figure 3). The system will then begin to load the data contained in his account, including emails, contacts, calendar, and location history. (For the interested, we will show the loading progress.) Once his data loaded, Mr. S. can execute SPARQL queries on it. This allows him to see, e.g.,

- what are the telephone numbers of his birthday party guests? (So he can send them a last-minute message.)
- what are the latest emails sent by any participant of the "Financial Restructuring" meetings?
- what are the places he visited during his previous trip to London? (So he does not go there a second time.)

Mr. S. can also perform analytics, e.g.,

- who does he most frequently communicate with?
- what are the places where he usually meets one particular person (based on his calendar)?
- how much time does it usually take to get an email answer from that particular person?

Figure 4 shows a sample query, which asks for emails sent by the organizer of an event during the month of the event.

We will show how users can verify that the enrichments proposed by the algorithm are correct, and possibly manually correct some of the knowledge. For this, the system keeps track, if necessary, of overwrites made to source data within the KB, ensuring that they are not overridden by the next incremental update. Finally, we will also present the results of some experiments we performed with real users and real data to illustrate the quality of the enrichers.

4. RELATED WORK

This work is motivated by the general concept of personal information management, see e.g. [1].

Personal Knowledge Bases. The problem of building a knowledge base for querying and managing personal information is not new. Among the first projects in this direction were IRIS [3] and NEPOMUK [11]. They used Semantic Web technologies to exchange data between different applications within a single desktop computer. They also provided semantic search facilities for desktop data. Our work is different from these projects: We do not tackle personal information management by reinventing the user experience for reading/writing emails, managing a calendar, organizing files, etc. We embrace personal information as being fundamentally distributed and focusing on the need of providing integration on top for creating completely new services (complex query answering, analytics).

Information Integration. Data matching (also known as record linkage, entity resolution, information integration, or object matching) is extensively utilized in data mining projects and in large-scale information systems by business, public bodies and governments [4]. Example application areas include national census, the health sector, etc. Recently, contact managers from known vendors have started providing de-duplication tools for finding duplicate contacts and merging them in bulk. However, these tools restrict themselves to contacts present in the user's address book.

Location History and Calendar. A lot of studies have already been done related around user location data. Few of them, however, have exploited the user's calendar and other available user data for creating richer and more semantic activity histories. Recently, a study has recognized the importance of fusing location histories with location data for improving the representation of information contained in the user's calendar: e.g. for distinguishing genuine real-world events from reminders [14].

Commercial Solutions. Some commercial providers, such as Gmail and Apple, have arguably come quite close to our vision of a personal knowledge base. They integrate calendars, emails, and address books, and allow smart exchanges between them. Google Now even pro-actively interacts with the user. However, these are closed-source. They do not allow the scientific community to build on their technology.

5. CONCLUSION

We propose to demonstrate a fully functional open-source personal knowledge management system. Our system integrates data from emails, calendars, address books, and the location history. Users can query the system for people, locations and events, or for any combination of these entities.

References

[1] S. Abiteboul, B. André, and D. Kaplan. "Managing your digital life". *Communications of the ACM* (2015).

[2] J. Broekstra, A. Kampman, and F. Van Harmelen. "Sesame: A generic architecture for storing and querying RDF and RDF schema". In: *ISWC '02*. Springer, 2002.

[3] A. Cheyer, J. Park, and R. Giuli. *IRIS: Integrate, Relate. Infer. Share.* Tech. rep. DTIC Document, 2005.

[4] P. Christen. *Data matching: concepts and techniques for record linkage, entity resolution, and duplicate detection*. Springer Science & Business Media, 2012.

[5] M. Crispin. *Internet Message Access protocol - version 4rev1*. RFC 3501. IETF, Mar. 2003.

[6] D. H. Crocker. *Standard for the format of ARPA Internet text messages*. RFC 822. IETF, Aug. 1982.

[7] R. Cyganiak, D. Wood, and M. Lanthaler. *RDF 1.1 Concepts and Abstract Syntax*. W3C, Feb. 2014.

[8] C. Daboo. *CardDAV: vCard Extensions to Web Distributed Authoring and Versioning (WebDAV)*. RFC 6352. IETF, Aug. 2011.

[9] C. Daboo, B. Desruisseaux, and L. Dusseault. *Calendaring Extensions to WebDAV (CalDAV)*. RFC 4791. IETF, Mar. 2007.

[10] B. Desruisseaux. *Internet Calendaring and Scheduling Core Object Specification (iCalendar)*. RFC 5545. IETF, Sept. 2009.

[11] S. Handschuh, K. Möller, and T. Groza. "The NEPOMUK project-on the way to the social semantic desktop". In: *I-SEMANTICS '07*. 2007.

[12] S. Harris, A. Seaborne, and E. Prud'hommeaux. *SPARQL 1.1 Query Language*. W3C, Mar. 2013.

[13] J. H. Kang, W. Welbourne, B. Stewart, and G. Borriello. "Extracting Places from Traces of Locations". In: *WMASH '04*. 2004.

[14] T. Lovett, E. O'Neill, J. Irwin, and D. Pollington. "The calendar as a sensor: analysis and improvement using data fusion with social networks and location". In: *UbiComp '10*. 2010.

[15] S. Perreault. *vCard Format Specification*. RFC 6350. IETF, Aug. 2011.

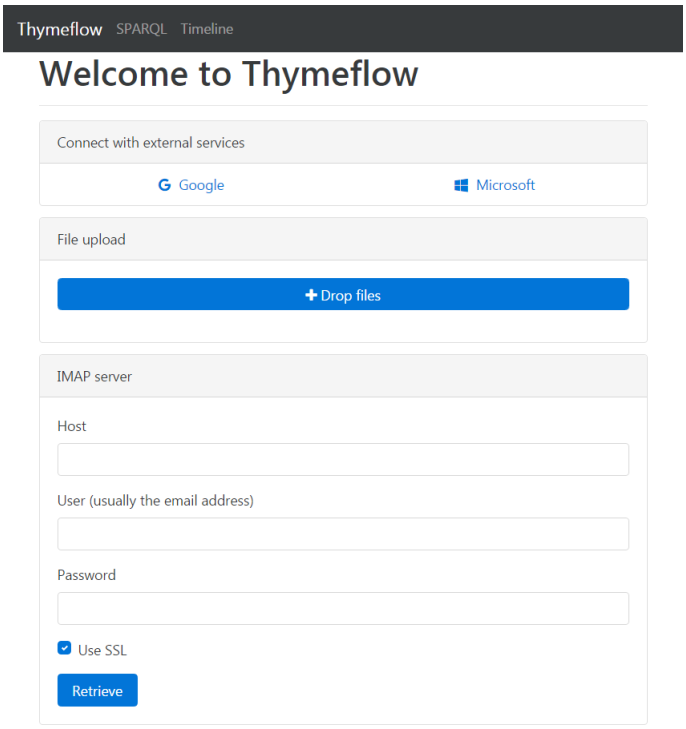

Figure 3: Thymeflow's user interface home page, which allows one to connect with external data sources.

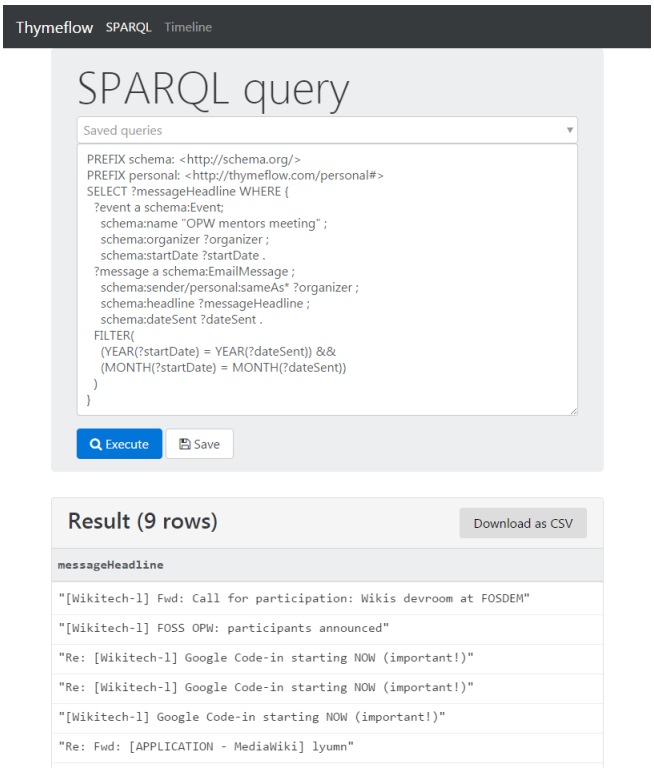

Figure 4: Thymeflow's user interface query capabilities – a query that retrieves the subject of all emails sent by the organizer of the "OPW mentors meeting" event during the month of the event.