LinkQS: Workshop on Linking The Quantified Self (LQS 2014)

Quantified Self (QS), also known as Personal Informatics (PI), is a school of thought that aims to use technology for acquiring and collecting data on different aspects of the daily lives of people. These data can be internal states (such as mood or glucose level in the blood) or indicators of performance (such as the kilometers run). The purpose of collecting these data is self-monitoring, performed in order to gain self-knowledge or some kind of change or improvement (behavioral, psychological, therapeutic, etc.). Although the current spread on the market of these kinds of tools, many issues arise when we consider their usage in the daily lives of common people, such as the meaningfulness and utility of the gathered data for the final users.

We can think to address some of these issues looking beyond the Quantified Self for finding new technologies and design techniques that could be applied to this field.

One of the main challenges of self-tracking data is that it comes in heterogeneous and often very unstructured form. One of the possible ways is leveraging *Semantic Web techniques* for integrating heterogeneous data originated from different devices and applications and give them some kind of structure. In Quantified Self, in fact, the information gathered by QS tools are scattered in autonomous silos, that can hardly be meshed together in order to provide users a complete and satisfying mirror of their behaviors and physical or psychological states. Besides, often QS tools simply juxtapose different data in their visualizations but they are not able to highlight meaningful correlations and provide structures for the data gathered.

Given that the quantified-self trend is just gaining momentum, it is not unlikely that we will soon have more and more users who create their own personal repositories, also referred to lifelogs. Structuring the data in these lifelogs is of particular importance in the context of user modeling. *User Modeling techniques* can provide useful insights for reasoning on data gathered, since users are not only in search of the possibility to visualize their behavioral data, but also to receive useful suggestions for improving their habits and behavior. Although QS tools have at their disposal huge amount of data on user behavior, they are not currently exploiting them for modeling users and providing them personalized recommendations.

In this workshop we tried to investigate challenges, open issues and new perspectives related to the dominion of data employed in Quantified Self and Personal Informatics technologies.

The workshop organizers:

Amon Rapp Università di Torino
Frank Hopfgartner, Technische Universität Berlin
Till Plumbaum, Technische Universität Berlin
Judy Kay, University of Sydney
Bob Kummerfeld, University of Sydney
Eelco Herder, L3S Research Center Hannover

Program
(accepted papers)

Federica Cena, Silvia Likavec, Amon Rapp, Martina Deplano and Alessandro Marcengo. Ontologies for Quantified Self: a semantic approach

Faisal Alquaddoomi, Cameron Ketcham, Deborah Estrin. The Email Analysis Framework: Aiding the Analysis of Personal Natural Language Texts

Timothy Wayne Cook and Luciana Tricai Cavalini. A Multilevel-Model Driven Social Network for Healthcare Information Exchange.

Program Committee

Rami Albatal, Dublin City University

Federica Cena, University of Torino

Na Li, Dublin City University

Alessandro Marcengo, Telecom Italia

Jochen Meyer, OFFIS, Germany

Ontologies for Quantified Self: a semantic approach

Federica Cena, Silvia Likavec,
Amon Rapp, Martina Deplano
University of Turin
Corso Svizzera 185, Torino, Italy
{cena,likavec,rapp,deplano}@di.unito.it

Alessandro Marcengo
Telecom Italia – Research and Prototyping Department
Via Reis Romoli 274, Torino, Italy
alessandro.marcengo@telecomitalia.it

ABSTRACT

The spreading of devices and applications that allow people to collect personal information opens new opportunities for user modeling (UM). In this new scenario UM together with personal informatics (PI) can offer a new way for self-monitoring that can provide the users with a sophisticated mirror of their behavior, attitudes and habits and their consequences on their life, on the environment and on contexts in which they live in. These new forms of self-reflection and self-knowledge can trigger and motivate the behavior change. In this paper we describe the first step in this direction, focusing on opportunities offered by semantic web ontologies for data integration and reasoning over data for recommendation purposes.

Categories and Subject Descriptors

H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous

General Terms

Languages.

Keywords

Ontologies, User model, Personal informatics, Quantified Self.

1. INTRODUCTION

Personalized systems are used to meet individual preferences and needs of each specific user, thus tailoring the system response to these particular requirements. Personalized systems extrapolate users' interests and preferences from explicit user ratings and from the observation of user behavior on the web: the system's assumptions about the user based on these observations are stored in a User Model (UM) [1]. A user model is the repository of personal information that has the potential to drive personalization and learning. The UM contains different types of information: from user demographic data to domain-specific preferences data (interest, knowledge…).

On the other hand, Personal Informatics (PI), also known as Quantified Self (QS), is a school of thought which aims to use the increasingly popular invisible technology means for acquiring and collecting data on different aspects of the daily lives of people. They allow users to self-track a variety of data about their own behavior: these data can be, on the one hand, user physical states (such as glucose level in the blood), psychological states (such as mood), behavior (such as movements), habits (such as food intake, sleep); on the other hand, they can be environmental parameters (such as CO2 content, temperature) and contextual information (such as people meeting) of the places passed through by the users during their everyday life. Thus, with this technology, we have the capability to automatically record at large scale the places that the users have been to, things they have seen, how they sleep, how active they are, etc., creating a constant stream of data that can reveal many aspects of their lives.

However, today all these data are scattered in autonomous silos and not integrated. UM techniques have the potential of aggregating and correlating data not only coming from web browsing but also provided by all these PI systems. A UM enriched with a plethora of personal data (behavioral, psychological, physical and environmental), related to different aspects of a person's daily life, will be able to provide the user with a "mirror" of herself, a sophisticated representation of interests, habits, activities in her life, in a novel way that is not yet achieved by any of the personal informatics tools available today [2]. This can support a new complex form of self-awareness and self-knowledge, which could foster behavior change processes [3], promoting more sustainable or healthier behavior, discouraging bad habits, sustaining therapeutic improvement and managing chronic diseases.

In this new scenario UM together with PI can offer a new way for self-monitoring people's own behavior, where self-monitoring refers to an assessment strategy to increase a person's awareness of targeted behavior [4], in order to promote behavior change [5].

UM and PI can provide users with a sophisticated mirror of their behavior, attitudes and habits, highlighting their consequences on their life, on the environment and on contexts in which they live in, promoting a new form of self-reflection and self-knowledge that can trigger and motivate the behavior change.

Our **goal** is to design a sophisticated UM-based PI system which can:

i) gather heterogeneous types of user data (from PI systems' sensors, from social web activities, from user's browsing behavior) and integrate them in an enhanced UM;

ii) reason on the gathered data in order to find aggregations and correlations among data;

iii) provide users with recommendations and meaningful UM visualizations to support self-awareness and self-knowledge.

The paper is structured as follows. We first present our solutions and then we focus on semantic modeling of the domain in order to allow data integration and reasoning.

2. STATE OF THE ART

Traditionally, **User Models (UMs)** [1,6] have the following features: (i) they are restricted to a single application; (ii) data are derived from the web; (iii) they concern short periods of time.

With the advent of ubiquitous computing technologies we are able to track and store large amounts of various personal information, scattered among applications and not integrated [7] even though it is possible to integrate them with semantic web techniques [8]. This project will advance the UM state of the art in the following:

- the integration of data derived from everyday life, in addition to the data derived from the web;
- reasoning on that data to gain further correlations about user behavior.

The opportunity is related to obtaining a Lifelong user model that stores user information for a long period of time and is able to manage user interest change [9]. This project is a first step in this direction.

According to [10], an **ontology** can be seen as a ‘‘formal, explicit specification of a shared conceptualization’’. With explicit specifications of domain objects and their properties, as well as the relationships between them, ontologies serve as powerful formalisms for knowledge representation, providing exact semantics for each statement and avoiding semantic ambiguities. For these reasons, ontologies are often used for semantic data integration and for resolving semantic conflicts, as in [11,12,13,14,15]. Also, the associated rigorous mechanisms allow for different forms of reasoning (for example, to deduce implicit classes), as in [16,17].

Measuring users' daily affective experiences is an important way to quantify their life. In [18], the authors measure users' emotions at various moments throughout the day. They asked the users to answer demographic and general satisfaction questions, to construct a short diary of the previous day, and then to answer structured questions about each episode. In [19], the authors investigate digital recordings of everyday activities, known as visual lifelogging, and elaborate the selection of target activities for semantic analysis. They investigate the selection of semantic concepts for life logging which includes reasoning on semantic networks using a density-based approach.

Motivating behavior change towards a more active lifestyle is a psychological, social and technological challenge. Several Personal Informatics Systems have been developed in order to try to modify a behavior by means of self-monitoring, such as [20,21,22]

3. A NOVEL SEMANTIC PI SYSTEM

We design a novel enhanced PI system, integrated in people’s everyday lives, able to gather data in a transparent way and to build and maintain a sophisticated user model able to aggregate data and provide meaningful visualization and personalized recommendations to the user for promoting behavior change. To reach this goal, we need the following components:

i) *data integration of different user data* for building a sophisticated model of user behavior, habits, needs and preferences coming from different sources (web and real life behavior)

ii) *advanced forms of reasoning* on user data for correlating different aspects of user daily behavior

iii) *personalized feedback for triggering behavior change* in the users:

- *recommendations* triggered by the correlation of different types of data (e.g., recommendations in accordance with user behavior, attitudes and habits in the UM)
- meaningful *visualization* of data for raising awareness and motivating people in changing their behavior.

In this paper we focus on data integration and reasoning over data (points i) and ii)) exploiting opportunities offered by semantic web ontologies [23]. Another challenging issue, namely gathering user data, is out of scope of this paper

4. ONTOLOGIES FOR QUANTIFIED SELF

In order to be able to:

integrate heterogeneous data coming from different devices and sources

reason on these data in order to provide meaningful visualization and recommendation

we design and develop three ontologies, modeling the three main concepts of the Quantified Self world: time, place and user activities. Vital parameters such as weight, blood pressure or blood sugar content are also important parameters, but we omit them from the preset analysis, since they are used primarily by medical experts and are hard to analyze by ordinary people.

Time ontology. We want to model the time from a user point of view, distinguishing work days, weekends and holidays (religious and civil ones), as well as dividing each day into meaningful slots (morning, afternoon, evening, night).

Place ontology. Again we want to model the place from a user perspective, labeling the places where the user lives, works or does the activities, dividing them into indoor (school, house, gym, work, cinema, restaurant, etc.) and outdoor (park, street..). (See Figure 1: Place ontology.)

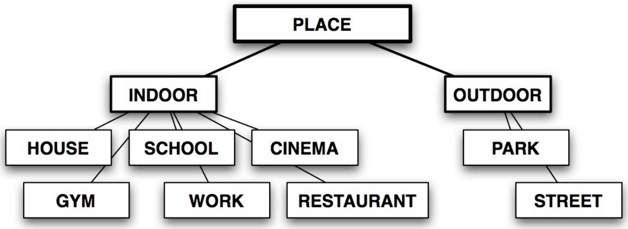


Figure 1: Place ontology

Activities ontology. We tried to model all the user activities, dividing them into two main categories: activities with place change (such as transportation or sports with place change) and activities with no place change (such as sports with no place change, intellectual activity, physical work, resting activity or feelings). Each of these classes has additional subclasses to better describe the performed activity, but we omit them from the picture for better clarity. For example, sports with place change has as its subclasses running, cycling, kayaking or downhill skiing, to name just a few. The design of this ontology was motivated by the categorization of activities in “Moves” application (https://www.moves-app.com). (See Figure 2: Activities ontology.) For lack of space, we included feelings into “Activities ontology”. We actually intend to have an additional “Wellbeing and emotions ontology” to model user’s emotional state and wellbeing, taking inspiration from [24].

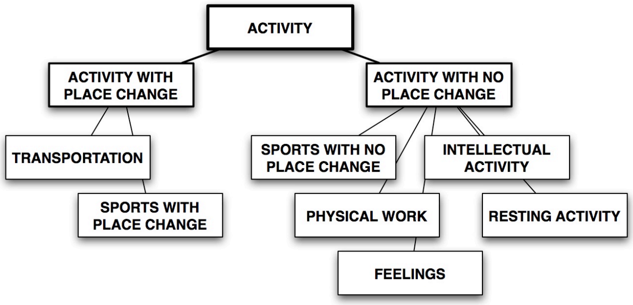


Figure 2: Activities ontology

Then, we use these ontologies in two ways.

First, we use ontologies to solve the possible data value and schema conflicts occurring among the data gathered from PI tools. As an example of data value conflicts, we gather “steps” both from the pedometer on the smart phone and from the smart bracelet and the collected numbers can differ: thus, in this case, we calculate an average number of steps. Even more challenging would be to deal with contradicting or seemingly unrelated data. For example, a pedometer might suggest that you were sedentary, while at the same time having the gym as your location. Pedometer forgotten in the locker or sitting in the gym bar? Another example concerns the mood levels: from an ad hoc app on the smart phone, we gather 4 mood values, whereas from the tangible channel we gather 6 mood values. Hence, the values should be normalized.

Schema conflicts are more complex: for example, what is modeled as an attribute in one relational schema may be modeled as an entity in another schema (e.g. "hour" as an attribute for the entity "sleep" and "hour" as an entity that has a relationship with "sleep"). As another example, two sources may use different names to represent the same concept (e.g. "running" and "jogging"), or the same name to represent different concepts, or two different ways for conveying the same information (e.g. "date of birth" and "age"). We solve these conflicts by mapping the data to our ontologies.

Second, we use these ontologies to make inferences useful for recommendation, in conjunction with Data Mining techniques for discovering correlations among data, where various forms of generalization can make correlations more powerful. For example, data mining techniques might provide a correlation between headache and running or biking activities. Since the two activities are two types of "outdoor activities" in the Activities Ontology, we can indicate a correlation between outdoor activities and headache. Alternatively, if we know that a certain user has a headache on December 24th, January 1st and August 15th, and from the Time Ontology we know that these are holidays, we can infer a correlation between holidays and headache.

Moreover, we could suggest a behavior that is similar or different but somehow related to what the user is used to doing. For example, if we know that the user loves running, but according to our data, we discovered that this is correlated with bad sleep, we might suggest some similar activities (in the same category) such as hiking or walking..

5. CONCLUSIONS

In this paper we tackle an important problem of long term management of users' data in PI systems and address a number of challenges including the need for data integration and interpretation. We motivate the introduction of suitable ontologies for modeling the core aspects of user behavior which would help overcome these problems.

This work is still at its early stage. We aim at experimentally evaluating our proposal by means of user tests to see short and long term effects of recommendations and visualizations on user behavior, as well as the acceptability of the solution.

6. ACKNOWLEDGMENTS

Our gratitude goes to Telecom Italia for their support.

7. REFERENCES

[1] Brusilovsky, P.: Methods and techniques of adaptive hypermedia. *User Modeling and User-Adappted Interaction*, v 6, n 2-3, pp 87-129, 1996

[2] Li, I., Dey, A. K., Forlizzi, J.: A Stage-Based Model of Personal Informatics Systems. In *Proceedings of SIGCHI Conf. on Human Factors in Computing Systems*, pp. 557-566. ACM, NY, USA, 2010

[3] Bandura, A.: Social Cognitive Theory of Self-Regulation. *Organizational Behavior and Human Decision Processes*, 50, 248--287 1991

[4] Burke, L. E., Wang, J., Sevick, M. A.: Self-monitoring in weight loss: a systematic review of the literature. *Journal of the American Dietetic Association*, 111(1), 92-102, 2011

[5] Bertram, C. L., Wang, J. B., Patterson, R. E., Newman, V. A., Parker, B. A., Pierce, J. P.: Web based self monitoring for weight loss among overweight/obese women at increased risk for breast cancer: the HELP pilot study. *Psycho-Oncology*, 22(8), 1821-1828, 2013

[6] Heckmann, D., Schwartz, T., Brandherm, B., Schmitz, M., von Wilamowitz-Moellendorff, M.: Gumo - the general user model ontology. In *User Modeling* 2005, pp. 428-432, Springer, 2005

[7] Aroyo, L., Dolog, P., Houben, G.-J., Kravcik, M., Naeve, A., Nilsson, M., Wild, F.: Interoperability in personalized adaptive learning. *Journal of Educational Technology and Society* 9(2), 4–18, 2006

[8] Sluijs, K.van der, Houben, G.-J.: A generic component for exchanging user models between web-based systems. *International Journal of Continuing Engineering Education and Life-Long Learning* 16(1–2), 64–76, 2006

[9] Kay, J., Kummerfeld, B. eds. Proceedings of the Lifelong User Modelling Workshop, at User Modeling Adaptation and Personalization Conf. UMAP '09, 2009

[10] Gruber, T. R.: A translation approach to portable ontology specifications. *Knowledge Acquisition Journal* 5 (2), 199–220, 1993.

[11] Arens, Y., Ciiee, Y., Knoblock. A.: SIMS Integrating data from multiple information sources. Information science institute, University of Southern California, U.S.A, 1992

[12] Goh, C.H., Bressan, S., Madnick, S. and Siegel. M.: Context interchange New features and formalisms for the intelligent integration of information. *ACM Transaction on Information Systems*, 17(3):270–290, 1999

[13] Beneventano, D., Bergamaschi, S., Guerra, F. Vincini. M. 2001: The MOMIS approach to information integration. In *ICEIS 2001, Proceedings of the 3rd Int. Conf. on Enterprise Information Systems*.

[14] Visser, P. R., Jones, D. M., Beer, M. , Bench-Capon, T., Diaz, B. and Shave, M.: Resolving ontological heterogeneity in the KRAFT project. In 10th Int. Conf. and Workshop on Database and Expert Systems Applications DEXA'99, 1999

[15] Abel, F., Herder, E., Houben, G. J., Henze, N., Krause, D.: Cross-system user modeling and personalization on the social web. *User Modeling and User-Adapted Interaction*, 23(2-3), pp.169-209, 2013

[16] Wang, X. H., Zhang D. Q., Gu, T., Pung, H., K.: Ontology Based Context Modeling and Reasoning using OWL. In *Proceedings of the 2nd IEEE Ann. Conf. on Pervasive Computing and Communications Workshops (PERCOMW '04)*. IEEE Computer Society, 2004

[17] Eiter, T., Ianni, G., Polleres, A., Schindlauer, R., Tompits, H.: Reasoning with rules and ontologies (2006), Reasoning Web 2006

[18] Kahneman, D., Krueger, A.B., Schkade, D.A., Schwarz, N., Stone, A.A.: A survey method for characterizing daily life experience: The day reconstruction method, Science 306, pp. 1776–1780, 2004

[19] Wang, P., Smeaton, A.F.: Semantics-based selection of everyday concepts in visual lifelogging, *International Journal of Multimedia Information Retrieval* 1, pp 87–101, 2012

[20] Shumaker, A., Ockene, J. K., Riekert, K.: *The Handbook of Health Behavior Change*, Springer, 2008

[21] Froehlich, J., Dillahunt, T., Klasnja, P., Mankoff, J., Consolvo, S., Harrison, B., Landay, J.: UbiGreen: Investigating a Mobile Tool for Tracking and Supporting Green Transportation Habits. In *Proceedings of the SIGCHI Conf. on Human Factors in Computing Systems*. pp. 1043-1052. ACM, New York, USA, 2009

[22] Kay, M., Choe, E. K., Shepherd, J., Greenstein, B., Watson, N., Consolvo, S., And Kientz, J. A.: Lullaby: a capture & access system for understanding the sleep environment. In *Proceedings of 2012 ACM Conf. on Ubiquitous Computing*. pp. 226--234, ACM, New York, USA, 2012

[23] Guarino, N.: Formal ontology and information systems. In Proceedings of the 1st Int. Conf. on Formal Ontology in Information Systems, FOIS '98, IOS Press, pp 3-15, 1998.

[24] Patti, V., Bertola, F.: Organizing Artworks in an Ontology-based Semantic Affective Space, Proceedings of the 1st Int. Workshop on Emotion and Sentiment in Social and Expressive Media: approaches and perspectives from AI, ESSEM@AI*IA, 2013.

The Email Analysis Framework: Aiding the Analysis of Personal Natural Language Texts

Faisal Alquaddoomi
UCLA Comp. Science Dept.
4732 Boelter Hall
Los Angeles, CA, USA
faisal@cs.ucla.edu

Cameron Ketcham
Cornell NYC Tech
111 8th Avenue #302
New York, NY 10011
cketcham@cornell.edu

Deborah Estrin
Cornell NYC Tech
111 8th Avenue #302
New York, NY 10011
destrin@cs.cornell.edu

ABSTRACT

Free-form email text streams are a rich, yet seldom-tapped, source of information about an individual's internal state. The difficulty in using this source of information is due partially to issues with obtaining and parsing these streams, and the sensitivity of the personal data they may contain.

This work presents a framework for allowing a user to authorize the acquisition and processing of emails from their Gmail account in order to model the user's use of language. The framework exposes a RESTful HTTPS API for third-party apps to produce personal analytics for the user from their language model, over which the user maintains fine-grained control by selectively granting access via OAuth2. Candidate applications that consume the language models are discussed, including how they may derive personal analytics from the provided data.

Categories and Subject Descriptors

I.2.7 [**Natural Language Processing**]: Text analysis

General Terms

Design

1. INTRODUCTION

As we interact with the world, we produce a profusion of data across different modalities. Of particular interest is the data we produce in communicating with other human beings, which could if collected and analyzed provide insight into our relationships with others as well our own internal state. This data often takes the form of free text which by its nature is qualitative, and thus challenging to analyze with quantitative methods. It is also frequently strewn across various services. Some of these services expose the data for public consumption, as in the case of social networking sites like Twitter, Facebook, or posts on personal blogs. Other services are more private, such as email and text messaging, and special care must be taken to gain access to the data as well as to preserve its privacy.

To summarize, the primary concerns are to securely collect, integrate, and analyze this often sensitive qualitative data. This paper proposes the implementation of a framework, the "Email Analysis Framework" (EAF), that consumes a user's sent email and produces a set of quantitative models and statistics informed by the field of natural language processing. While the scope of the project is currently to collect and process email, the intent is to expand the framework to collect and integrate other sources of free text, for instance from social networking sites. It is hoped that the EAF will be used as a proxy for these qualitative data sources, providing a foundation upon which user-facing tools can be built to derive insights about this data for the individual in a privacy-preserving way.

The EAF is currently under active development, but an alpha version of the tool is available[1], as is the source code [2]. This paper principally describes the structure and design choices in acquiring, analyzing, and disbursing sensitive data. Applications are discussed in 4, which currently consist of a completed sample EAF consumer that produces trivial visualizations as well as two more significant applications that are currently in development.

2. APPROACH AND RELATED WORK

As described in [13], the overarching intent of quantifying the self is to collect, integrate, and analyze data streams that may be indicative of an individual's physical, emotional, and psychological state. The purpose of this analysis is to promote awareness of how these measurable quantities both affect and can be affected by the individual's state, and to provide support for decisions that change that state. As mentioned previously, free text is both relatively easy to collect and clearly carries much information about how we feel about others and ourselves; indeed, it has been demonstrated that even our choices of words reflect our psychological state [11]. While this data may be present, it is in an opaque form that must be parsed into usable quantitative data.

The analysis of free text has been extensively addressed in the field of natural language processing (NLP). NLP con-

[1] https://eaf.smalldata.io

[2] https://github.com/falquaddoomi/social_text_processor/

cerns itself with the broad task of comprehending (that is, unambiguously parsing) and extracting structured information from human language, which is accomplished through two main approaches: rule-based (aka grammatical) and statistical methods. The EAF primarily makes use of these statistical methods, specifically n-gram language modeling, to build a sequence of generative models of an individual's use of language over time.

n-gram models are sufficiently descriptive of an individual's use of language that they can be used to discriminate one author from another purely by comparing descriptive statistics computed over them, such as the entropy or the perplexity of the distributions [10, 14]. Descriptive statistics, such as the entropy of a language model mentioned previously, are of special appeal to privacy because they provide an essential determination about the author without compromising the original content from which the statistic was derived.

A user's email is a unique corpus in that each document (i.e. email) is tagged with a host of metadata, including the time it was sent. Thus, computing language models over brackets of emails close in time can provide "snapshots" of the evolution of a user's use of language over time. These snapshots can be compared against each other to determine if there are shifts in the style of the user's written communications which could perhaps correlate to life events. There may be regularities in the changes of these models, or similarities to other people's models with whom the individual interacts. The snapshots can be filtered by recipient or by communication mode to determine if the audience or medium determines the way an individual writes, or if there are detectable groupings. Many more examples could be proposed for these temporal language models, especially when other sources of time-based data (location, activity, calendar events, etc.) are introduced. One of the EAF's main goals is to provide infrastructure to build and maintain these models, as well as allow them, and the descriptive statistics derived from them, to be released at the user's discretion for comparison to other data sources.

There are other frameworks which provide similar analytical capabilities, notably the General Architecture for Text Engineering (GATE) [2]. There are also numerous libraries and toolkits [3, 6] that include the same features that the EAF provides – in fact, the EAF makes use of the popular nltk library [1] to perform many of its functions. The EAF differs from these projects in its context: it is a deployable system focused on centralizing the secure acquisition and processing of emails for many users. It provides user-facing administrative interfaces to control it, and app-facing APIs to make use of its results. The EAF's intent is to allow users to make sense of their own data, and uses a fine-grained opt-in permission system fully controlled by the user to help protect against malicious or unintended use of the user's email data.

In the context of email analysis, the MIT Media Lab's Immersion project[7] shares the EAF's goal of using one's email for the purpose of personal insight and self-reflection. Unlike the EAF, the Immersion project restricts itself to analysis of the user's social group through reading the "From" and "To" fields of email header – no examination of the body text is performed. Further, the output of the Immersion project is an infographic and not raw data that can be reused by other components, whereas the EAF's purpose is to facilitate analysis by other tools.

3. ARCHITECTURE

The EAF's first task is to transform a user's sent email messages into a series of tokens, where each token is tagged with the time at which it was sent. This series of time-tagged tokens constitutes a "stream", from which the n-gram models mentioned previously are built. The stream is quantized into intervals; the ordering of tokens within these intervals is not preserved from their originating messages (outside of their order in the n-grams), with the express intention of making it difficult to reconstruct the original text. After quantization, the stream is then made available at the user's discretion to third-party applications ("consumers"), with the ability for the user to configure per-consumer filters that control what information that consumer can access. A few candidate consumers are discussed in the "Applications" section 4. In order to mitigate the danger of storing sensitive user credentials, the EAF makes extensive use of the OAuth2[4] standard, both as an OAuth2 consumer (of Gmail, currently) and as an OAuth2 provider. The use of OAuth2 also allows the user the freedom of revoking access to the EAF should they wish to discontinue its use, or to revoke access to third-party apps that had been authorized to consume the EAF's API. After the initial synchronization, future emails that the user sends are automatically acquired by the system by periodically polling the provider.

3.1 Structure

The EAF consists of three main components, as depicted in figure 1: a web interface through which the user authorizes access to their Gmail account and performs administrative tasks, a task processor which acquires the user's email and produces a token stream from it, and a second web interface which faces consumers of the token stream. Both web interfaces are implemented in Django 1.7, a framework for rapid development of web applications in Python. Authorization to third-party services is facilitated by Django-Allauth, a project that allows Django's built-in authentication system to interoperate with a host of OAuth2 providers, including Gmail. The task processor makes use of Celery, a distributed task queue processor that is often used in concert with Django. Both components communicate via a shared database, specifically PostgreSQL, which was chosen for its performance under heavy, highly concurrent loads.

The framework exposes a RESTful HTTPS interface to allow third-party applications to consume the token stream. The implementation of this interface was aided by the Django-REST-framework, and the specifications of the interface follow the openmHealth DSU specification v1.0, [9]. The user-facing web interface makes use of the RESTful interface itself for querying the token stream. In order to allow registered third-party sites to gain access to the user's email data for analysis and visualization, the EAF acts as an OAuth2 provider; third-party sites must involve the user in their request for a temporary access token, which they can subsequently use to make requests on the user's behalf.

3.2 User Interaction

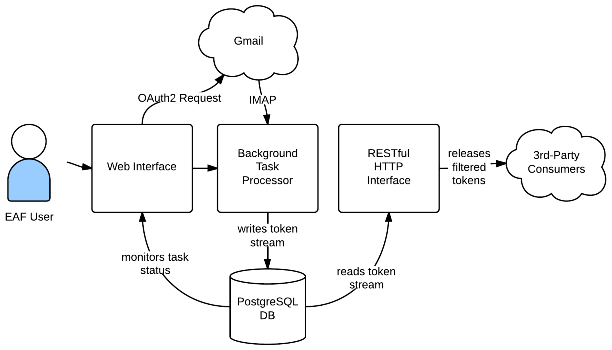


Figure 1: Structure of the Email Analysis Framework

Prior to using the system the user first creates an EAF site account which acts as an aggregation point for the multiple email accounts they might want to use. At the moment this account creation is performed automatically when the user authorizes their first email account; the account they authorize (or any other linked account) then implicitly logs them in to their site account, although this behavior is subject to change in the future.

In their interaction with the system, the user proceeds through three stages:

1. **Authorization**, in which the user is prompted to release temporary credentials used to access their email account via OAuth2.
2. **Acquisition**, during which the user monitors the progress of the system as it downloads their emails and performs filtering/transformations before inserting them into the database as a stream of tokens.
3. **Release**, in which the user selects which consumers can access their token stream and what filtering/transformations will be applied for that consumer.

3.2.1 Authorization

The authorization stage is initiated when the user visits the web interface. Using a standard OAuth2 handshake, the user is redirected to Google's Gmail authorization page, where they log in (or use a previously logged-in session) and then accept the permissions which the framework requests, specifically access to the user's email. If the user provides their consent, they are returned to the EAF where they can proceed to acquisition. If the user does not provide consent or some other error occurs, they are returned to the framework with an error message and are prompted to try again. Multiple email accounts can be associated with a single EAF site account, in which case selecting an account from the list of registered accounts begins the next stage, acquisition.

3.2.2 Acquisition

Initial Acquisition. Acquisition starts immediately after authorization and is handled by the background task processor. The user is shown a view of the task's progress which is periodically updated. The process can be quite lengthy, especially in the case where there is a large backlog of messages to process, so the user is permitted to close the view and return to the site at their convenience to check in on the task's progress. Upon completion, the framework sends a notification email which includes the total duration of the acquisition task. At this point, the user can view the results of the acquisition process in a small built-in visualization dashboard that shows a few summarizing statistics about their token stream plotted over time. Incremental acquisition tasks that occur after the initial acquisition do not trigger a notification.

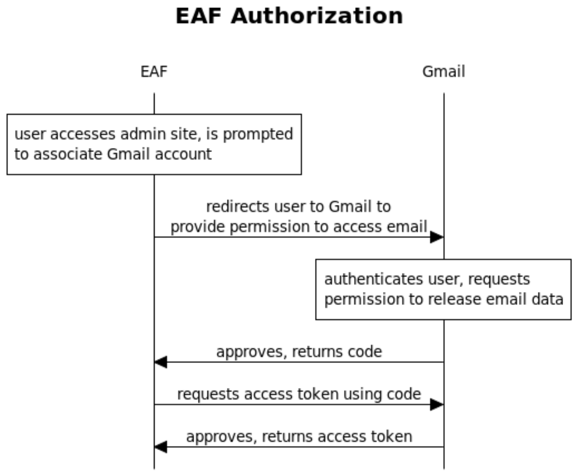

EAF Authorization
EAF
Gmail
user accesses admin site, is prompted to associate Gmail account
redirects user to Gmail to provide permission to access email
authenticates user, requests permission to release email data
approves, returns code
requests access token using code
approves, returns access token

Figure 2: Gmail Authorization

Since the framework is intended to model the user's use of language and not the language of other individuals with whom the user is conversing, it is necessary to strip quotations and reply text from the emails prior to processing. Isolating only the user's text in sent mail is accomplished through an adapted version of the email_reply_parser [3] library, developed by GitHub.

Ongoing Acquisition. In the background task processor, the acquisition task consists of using the previously-obtained OAuth2 credentials to authenticate to Google's IMAP server. The task then proceeds to download the user's sent email (that is, the contents of "GMail\[Sent Mail]") in chronological order, skipping messages which have been recorded as processed in a previous iteration of the task. Each email is passed through a series of filters, called the "pre-filter chain", which ultimately results in a sequence of tokens that are associated with the email account, the user's EAF site account, and the time at which the email was sent. By default, the first filter in the chain performs tokenization: each email is split on newlines and punctuation into tokens, which are converted to lowercase to reduce the number of possible tokens due to capitalization differences, and stripped of numbers and quotation marks. The second filter is the "ignored words" filter, which allows the user to selectively prohibit certain words from ever entering the database. At the mo-

[3] `https://github.com/github/email_reply_parser`

ment, the ignored words must be manually entered, which makes filtering passwords and other sensitive information problematic, given that the ignored list itself is then sensitive. This will be addressed in the subsection on filter types, 3.3.

After the filter chain runs, the tokens are then written to the database. Rather than store repeated tokens individually, each token is stored with a count of the number of times it occurred within its message. If same token occurs in different messages, it is stored separately for each message. This choice was made as a compromise between allowing for flexible choice of the interval into which tokens are combined when the stream is consumed and consuming less space in the database; if the system were designed with a fixed interval rather than a flexible one, the tokens would simply be combined into a histogram for each interval.

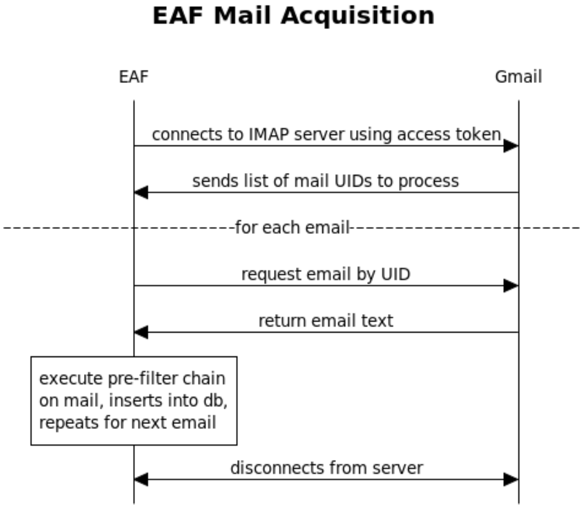

EAF Mail Acquisition
EAF
Gmail
connects to IMAP server using access token
sends list of mail UIDs to process
for each email
request email by UID
return email text
execute pre-filter chain on mail, inserts into db, repeats for next email
disconnects from server

Figure 3: Mail Acquisition

3.2.3 Release

Once the user has found an EAF-compatible application, they can authorize that application to access their token stream via OAuth2. In this stage the EAF acts as an OAuth2 provider, providing a page to which the third-party application can redirect the user to be prompted for authorization of their identity via Gmail (used also as credentials to access their EAF data) and permission to access their token stream. In the case where the user has multiple token streams, they will be prompted to choose the streams to which they are granting access. On this page, the user selects a filter chain for each stream that will be used specifically with this consumer, or simply opt not to filter the stream at all. The process is detailed in figure 4.

After this point, the consumer can request updates from the token stream at any time. The EAF audits all accesses, displays the last time of access, and allows the user to revoke the consumer's access at any time or change the filter chain associated with that consumer.

3.3 Filtering

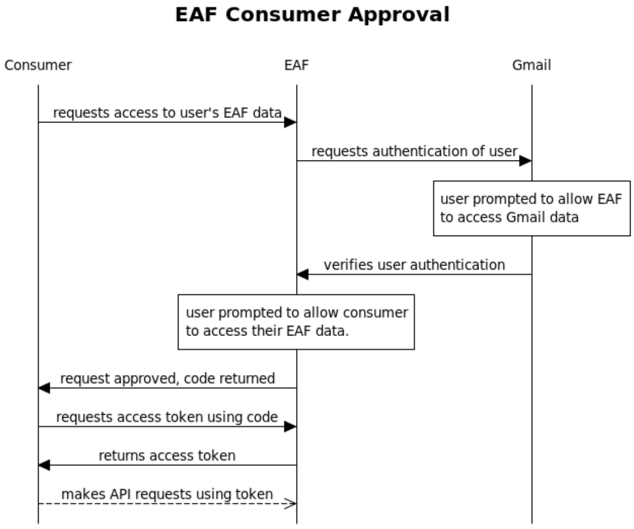

EAF Consumer Approval
Consumer
EAF
Gmail
requests access to user's EAF data
requests authentication of user
user prompted to allow EAF to access Gmail data
verifies user authentication
user prompted to allow consumer to access their EAF data.
request approved, code returned
requests access token using code
returns access token
makes API requests using token

Figure 4: Release to Consumer

As previously mentioned, both the acquisition and release stages employ a sequence of filters that allow the input data to be selectively screened for sensitive terms and otherwise transformed to better suit the requirements of the consumers. The acquisition stage's filter chain is referred to as the "pre-filter chain" and the release stage's is the "post-filter chain". There is only a single pre-filter chain, but there can be as many as one post-filter chain for each registered consumer.

The pre-filter chain always has a special "tokenize" filter as its first filter, which produces the initial sequence of tokens for filtering and transformation, and may only be used in the pre-filter chain. A second special filter that may only be used in the pre-filtering step is the "ignore word sequence" filter, which ignores the sequence of tokens configured in the filter, and was initially created to ignore signature blocks. This filter can only function in the pre-filtering step as the exact sequence of the tokens is lost upon insertion into the database.

Aside from the special "tokenize" filter, there are a few other filters which can only be used in the pre-filtering step, namely:

- **Parts-of-Speech Tagger**, which replaces each token with its detected part of speech (noun, verb, etc.)
- **Fork**, which produces an identical stream to the current one, but with its own sub-filter chain. The tokens that are produced from a fork are tagged with a unique ID corresponding to that fork.

The "fork" filter is especially useful in conjunction with the part-of-speech tagger, as both the original text and the parts-of-speech stream can be either individually released or released together, which allows for analysis of the user's grammar. Note that the parts-of-speech stream does preserve the order of tokens in the original stream, but not the text of the tokens themselves.

The filter framework is modular, with the potential to add new filters easily in the future. At the moment, a few parameterizable filters are implemented to change the case of tokens, strip specific characters, and to remove words that are either on a user-specified list or not contained within aspell's "en_US" dictionary. Detecting and ignoring named entities is a work in progress.

- **Change Case**, which transforms the case of the tokens;
- **Strip Characters**, which can be used to remove numbers and other special characters;
- **Ignore Listed Words**, which removes tokens on an "ignore" list from the token stream; and
- **Ignore Non-Dictionary Words**, which removes tokens not found in a common English dictionary

By utilizing the "ignore words" filters, the user is allowed fine-grained control of both the contents of the EAF's database and the views of the token streams presented to different consumers.

4. APPLICATIONS

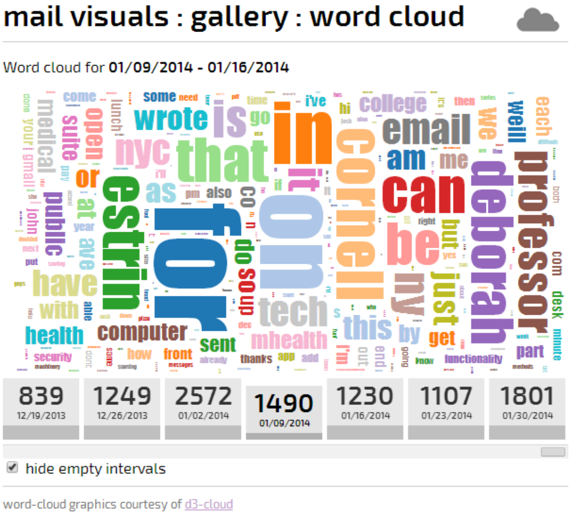
As mentioned, third-party applications can gain temporary access to a user's data for the purpose of visualizing or otherwise processing it. Granting this access is currently at the user's discretion; the user should make use of the per-consumer post-filter controls to limit the release of sensitive information to potentially untrustworthy parties. Consumers sign requests to the EAF's RESTful JSON API with an access token, obtained through the process described in the "Release" section above 3.2.3.

4.1 Example: Mail Visualization Front-End

In order to demonstrate the functionality of the framework, a visualization front-end was developed that consumes the framework's API and produces a few example visualizations. The front-end also serves as a reference implementation of EAF client authentication; it first requests permission from the user before gaining access to their token stream. The visualization front-end currently offers the following modules:

- **Word Cloud** - a "word cloud" infographic that can be viewed on a per-week basis (figure 5).
- **Rhythm** - a table of the days of the week as the columns and hours of the day as the rows is colored according to the number of emails sent within each interval, with darker colors corresponding to more emails sent (a heatmap, essentially; figure 6).
- **Alters** - a bar chart of the number of people contacted per week; when a week is selected, displays a bar chart of emails sent to each person.

These visualization modules are intended to be a jumping-off point for more useful visualizations to be constructed, which would ideally incorporate data from other sources to strengthen inferences about the user's overall state.



mail visuals : gallery : word cloud

Word cloud for **01/09/2014 - 01/16/2014**

☑ hide empty intervals

word-cloud graphics courtesy of d3-cloud

t]

Figure 5: "Word Cloud" Visualization

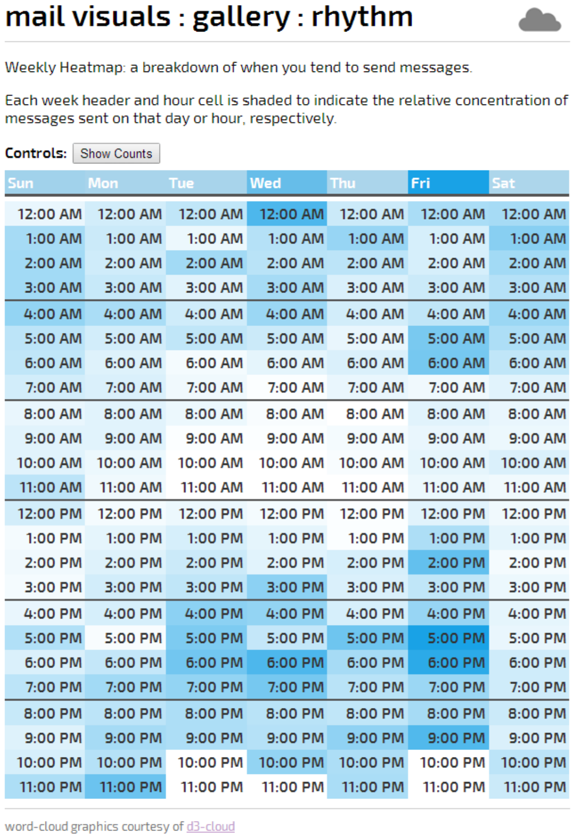
4.2 Pulse and Partner

In addition to the sample application discussed above, our group is currently developing two applications that make use of statistics computed against the user's email. The first is "Pulse", which makes use of location traces from the user's smartphone as well as the frequency and variety of individuals with whom one communicates to compute a score that indicates **how** rather than **what** the individual is doing. This score is visualized as a waveform over a short window of time (i.e. a week), which can be shared with family members and friends. The second is "Partner", which is intended to measure the degree to which linguistic style matching occurs among individuals who interact with each other face to face, a fairly well-documented phenomenon [8], [5]. Partner makes use of the location traces of two or more individuals as well as computed statistics over their emails to produce two scores, a "proximity" and a "language-style matching" score, which will be visualized as individual timeseries. A third timeseries will display their computed correlation over time.

5. CONCLUSION, FUTURE WORK

The Email Analysis Framework, a system for extracting structured, easily-consumed data from time-tagged natural-language text was proposed in this work. At the moment it is limited to acquiring text from Gmail, computing, and exposing language models to other tools via a RESTful HTTPS API, but it is hoped to be extended to other sources of personal natural-language text, such as Facebook and Twitter streams. A few candidate visualizations were described to both demonstrate how the data could be used and to stimulate investigation into more novel applications.

In terms of future work, there are extensions planned to all the stages of the framework. As mentioned, the scope of providers is intended to be expanded to other text providers, which will allow analysis to be performed on how different media affect the language model. Additional streams can be extracted in the processing phase, such as identifying



mail visuals : gallery : rhythm

Weekly Heatmap: a breakdown of when you tend to send messages.

Each week header and hour cell is shaded to indicate the relative concentration of messages sent on that day or hour, respectively.

Controls: Show Counts

Sun	Mon	Tue	Wed	Thu	Fri	Sat
12:00 AM	12:00 AM	12:00 AM	12:00 AM	12:00 AM	12:00 AM	12:00 AM
1:00 AM	1:00 AM	1:00 AM	1:00 AM	1:00 AM	1:00 AM	1:00 AM
2:00 AM	2:00 AM	2:00 AM	2:00 AM	2:00 AM	2:00 AM	2:00 AM
3:00 AM	3:00 AM	3:00 AM	3:00 AM	3:00 AM	3:00 AM	3:00 AM
4:00 AM	4:00 AM	4:00 AM	4:00 AM	4:00 AM	4:00 AM	4:00 AM
5:00 AM	5:00 AM	5:00 AM	5:00 AM	5:00 AM	5:00 AM	5:00 AM
6:00 AM	6:00 AM	6:00 AM	6:00 AM	6:00 AM	6:00 AM	6:00 AM
7:00 AM	7:00 AM	7:00 AM	7:00 AM	7:00 AM	7:00 AM	7:00 AM
8:00 AM	8:00 AM	8:00 AM	8:00 AM	8:00 AM	8:00 AM	8:00 AM
9:00 AM	9:00 AM	9:00 AM	9:00 AM	9:00 AM	9:00 AM	9:00 AM
10:00 AM	10:00 AM	10:00 AM	10:00 AM	10:00 AM	10:00 AM	10:00 AM
11:00 AM	11:00 AM	11:00 AM	11:00 AM	11:00 AM	11:00 AM	11:00 AM
12:00 PM	12:00 PM	12:00 PM	12:00 PM	12:00 PM	12:00 PM	12:00 PM
1:00 PM	1:00 PM	1:00 PM	1:00 PM	1:00 PM	1:00 PM	1:00 PM
2:00 PM	2:00 PM	2:00 PM	2:00 PM	2:00 PM	2:00 PM	2:00 PM
3:00 PM	3:00 PM	3:00 PM	3:00 PM	3:00 PM	3:00 PM	3:00 PM
4:00 PM	4:00 PM	4:00 PM	4:00 PM	4:00 PM	4:00 PM	4:00 PM
5:00 PM	5:00 PM	5:00 PM	5:00 PM	5:00 PM	5:00 PM	5:00 PM
6:00 PM	6:00 PM	6:00 PM	6:00 PM	6:00 PM	6:00 PM	6:00 PM
7:00 PM	7:00 PM	7:00 PM	7:00 PM	7:00 PM	7:00 PM	7:00 PM
8:00 PM	8:00 PM	8:00 PM	8:00 PM	8:00 PM	8:00 PM	8:00 PM
9:00 PM	9:00 PM	9:00 PM	9:00 PM	9:00 PM	9:00 PM	9:00 PM
10:00 PM	10:00 PM	10:00 PM	10:00 PM	10:00 PM	10:00 PM	10:00 PM
11:00 PM	11:00 PM	11:00 PM	11:00 PM	11:00 PM	11:00 PM	11:00 PM

word-cloud graphics courtesy of d3-cloud

Figure 6: "Rhythm" Visualization

named entities and topics, all of which can be analyzed over time, audience, etc. Industry-standard information extraction techniques such as autoslog [12] could be applied to discover meeting arrangements, events that occur to named entities or topics mentioned in the emails, and so on. Sentiment analysis could be computed and exposed as another temporal stream, to attempt to model the user's disposition as a function of time. Additional third-party applications are planned, such as a tool for determining points of inflection in the descriptive statistics computed on the language model, and a tool to easily correlate other time-based data against the statistics streams.

6. REFERENCES

[1] S. Bird. Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, pages 69–72. Association for Computational Linguistics, 2006.

[2] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. Gate: an architecture for development of robust hlt applications. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 168–175. Association for Computational Linguistics, 2002.

[3] D. Ferrucci and A. Lally. Uima: an architectural approach to unstructured information processing in the corporate research environment. *Natural Language Engineering*, 10(3-4):327–348, 2004.

[4] D. Hammer-Lahav and D. Hardt. The oauth2. 0 authorization protocol. 2011. Technical report, IETF Internet Draft, 2011.

[5] M. E. Ireland, R. B. Slatcher, P. W. Eastwick, L. E. Scissors, E. J. Finkel, and J. W. Pennebaker. Language style matching predicts relationship initiation and stability. *Psychological Science*, 22(1):39–44, 2011.

[6] A. Mallet. Java-based packed for statistical nlp toolkit. *Available at (accessed 26.01. 10)*, 2010.

[7] Mit media lab's immersion project. `https://immersion.media.mit.edu/`. Accessed: 2014-02-04.

[8] K. G. Niederhoffer and J. W. Pennebaker. Linguistic style matching in social interaction. *Journal of Language and Social Psychology*, 21(4):337–360, 2002.

[9] Ohmage dsu 1.0 specification. `https://github.com/openmhealth/developer/wiki/DSU-1.0-Specification`. Accessed: 2014-02-04.

[10] F. Peng, D. Schuurmans, V. Keselj, and S. Wang. Automated authorship attribution with character level language models. In *10th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2003)*, pages 267–274, 2003.

[11] J. W. Pennebaker, M. R. Mehl, and K. G. Niederhoffer. Psychological aspects of natural language use: Our words, our selves. *Annual review of psychology*, 54(1):547–577, 2003.

[12] E. Riloff and W. Phillips. An introduction to the sundance and autoslog systems. Technical report, Technical Report UUCS-04-015, School of Computing, University of Utah, 2004.

[13] M. Swan. The quantified self: Fundamental disruption in big data science and biological discovery. *Big Data*, 1(2):85–99, 2013.

[14] Y. Zhao, J. Zobel, and P. Vines. Using relative entropy for authorship attribution. In *Information Retrieval Technology*, pages 92–105. Springer, 2006.

A Multilevel-Model Driven Social Network for Healthcare Information Exchange

Timothy Wayne Cook
National Institute of Science and Technology -
Medicine Assisted by Scientific Computing
Petrópolis, Brazil
+5521994711995
tim@mlhim.org

Luciana Tricai Cavalini
Department of Health Information Tecnology
Rio de Janeiro State University
Rio de Janeiro, Brazil
+552128688378
lutricav@lampada.uerj.br

ABSTRACT

The management of Big Data in healthcare is challenging due to of the evolutionary nature of healthcare information systems. Information quality issues are caused by top-down enforced data models not fitted to each point-of-care clinical requirements as well as an overall focus on reimbursement. Therefore, healthcare Big Data is a disjointed collection of semantically confused and incomplete data. This paper presents MedWeb, a multilevel model-driven, social network architecture implementation of the Multilevel Healthcare Information Modeling (MLHIM) specifications. MedWeb profiles are patient and provider-specific, semantically rich computational artifacts called Concept Constraint Definitions (CCDs). The set of XML instances produced and validated according to the MedWeb profiles produce Hyperdata, overcoming of the concept of Big Data. Hyperdata is defined as syntactically coherent and semantically interoperable data that can be exchanged between MedWeb applications and legacy systems without ambiguity. The process of creating, validating and querying MedWeb Hyperdata is presented.

Categories and Subject Descriptors

I.2.4 [**Computing Methodologies**]: Knowledge Representation Formalisms and Methods – *representation languages, semantic networks.*

General Terms

Management, Design, Standardization, Languages.

Keywords

Semantic interoperability; healthcare information exchange; Big Data.

1. INTRODUCTION

The health status of any population is the fundamental, common denominator to all other aspects of life. Without good health, a population will not thrive. Proper information management is key to good decision making at all levels of the healthcare system, from the point of care to the national policy making [1]. A given healthcare provider can have access to many sources of Big Data in healthcare and still not have access to meaningful clinical information. Having accurate, timely and semantically meaningful healthcare information is key to protecting the public in healthcare emergencies and in the day-to-day decision making in allocating scarce healthcare resources [2]. Therefore, it is important to ensure that the information related to each individual healthcare event is recorded at the moment and the place where the event happened, which is the most realistic representation of a given healthcare event. When the healthcare provider or the individual (the two most important components of the decision intelligence chain in healthcare) have control over the way this information is structured and how semantics is persisted, the realism of the knowledge representation is maximized [3].

The effectiveness of healthcare systems can be measured by their adequate response to the demographic and epidemiological profile of their target population. Over the last decades, these profiles have shown fast and complex changes due to globalization, as it can be seen during the occurrence of epidemics and pandemics, as well as in the daily overcrowding of emergency services [4]. The incorporation of Information Technology (IT) in healthcare has been proposed as a strategy to overcome the current situation, but there are obstacles for the accomplishment of this promise, which are derived from the significant complexities of health information in the dimensions of space, time and ontology.

In addition, in the typical healthcare provider spectrum, each provider has different information needs. Therefore, the applications or at least the views into applications need to be very specific in order to improve usability [5]. Large standardized systems are usually slow to change and adapt to the rapid rate of change dictated by the adoption of new emerging medical technologies [6]. The end result of the presence of such complexity in healthcare information systems is that they are usually not interoperable and have high maintenance costs. These issues have a significant impact on the low level of adoption of information technology by healthcare systems worldwide, in particular when compared to other sectors of the global economy [7].

The complex scenario of global health informatics has been studied over the last half of the 20th century and into the 21st century along with the explosion of information technology. Many different (and very costly) solutions have been proposed to the interoperability and maintenance problems of healthcare applications, with limited results [8]. In the past two decades, a different approach has been proposed for the development of healthcare information systems. This approach is generically defined as the Multilevel Model-Driven (MMD) approach and its main feature is the separation between the data persistence mechanisms and the knowledge modeling [9].

There are three MMD specifications available: the dual-model proposed by the openEHR Foundation [10], the ISO 13606 Standard [11], both of them adopting the object-oriented approach, and the Multilevel Healthcare Information Modeling (MLHIM) specifications [12], implemented in eXtensible Markup Language (XML) technologies. MedWeb is the implementation of the MLHIM specifications using many concepts of a social network application.

This paper presents the technical background for the implementation of MedWeb, including the definition of 'hyperdata', in dialectic relationship to the concept of Big Data, as well as the description of the technological solutions adopted in MedWeb for the process of generating, validating and querying hyperdata instances.

2. METHOD

MedWeb is a MLHIM-based meta-application, with a workflow structure set up as a social network, also providing the interface with independently developed MLHIM-based applications and other legacy systems. The MLHIM specifications are published (https://github.com/mlhim) as a suite of open source tools and documentation for the development of electronic health records and other types of healthcare applications, according to the MMD principles. The specifications are structured in two Models: the Reference Model and the Domain Model.

The abstract MLHIM Reference Model is composed of a set of classes (and their respective attributes) that allow the development of any type of healthcare application, from hospital-based electronic medical records to small purpose-specific applications that collect data on mobile devices. This was achieved by minimizing the number and the residual semantics of the Reference Model classes, when compared to the openEHR specifications. The remaining classes and semantics were regarded as necessary and sufficient to allow any modality of structured data persistence. Therefore, the MLHIM Reference Model approach is minimalistic, but not as abstract as a programming language [9].

In the MLHIM Reference Model implemented in XML Schema 1.1, each of the classes from the abstract Reference Model are expressed as a complexType definition, arranged as 'xs:extension'. For each complexType there are also 'element' definitions. These elements are arranged into substitution groups in order to facilitate the concept of class inheritance defined in the abstract Reference Model.

The MLHIM Domain Models are defined by the Concept Constraint Definitions (CCDs), also implemented in XML Schema 1.1, being conceptually similar to the openEHR and ISO 13606 archetypes. Each CCD defines the combination and restriction of Pluggable complexTypes (PcTs) and their elements of the (generic and stable) MLHIM Reference Model implementation in XML Schema 1.1 that are necessary and sufficient to properly represent any given clinical concept. In general, CCDs are set to allow wide reuse, but there is no limitation for the number of CCDs allowed for a single concept in the MLHIM eco-system, since each CCD is identified by a Type 4 Universal Unique Identifier (UUID) [12]. This provides permanence to the concept definition for all time, thus creating a stable foundation for instance data established in the temporal, spatial and ontological contexts of the point of recording.

The MLHIM implementation uses XML Schema 1.1 in an innovative way. Modeling each PcT in a CCD by defining further restrictions on the Reference Model (RM) types as the xs:base in an xs:restriction. Giving the fact that the majority of medical concepts are multivariate, for the majority of CCDs, a n (n > 0) number of PcTs will be included. For instance, since it is likely to have a CCD with more than one PcT, each one of them will be nmed with a Type 4 UUID [12]. This allows the existence of multiple PcTs of the same RM complexType (e.g., ClusterType, DvAdapterType, DvStringType, DvCountType) in the same CCD without conflict. This approach also enables data query, since it creates a universally unique path statement to any specific MLHIM based data. This query approach holds true even when PcTs are reused in multiple CCDs.

Figure 1 shows the conceptual view of the sections of a CCD. Notice that the CCD is composed of two sections: the Metadata (white box) and the Definition (green oval). Primarily the definition is the structural component and the metadata is the ontological component of the concept. These are the overall separations between the two sections. Though it can be argued that the definition does carry some semantics as well as structural information about a concept; the metadata section is where the semantics for the entire CCD concept is defined and is therefore available for any healthcare application to discover about instance data. The blue circles represented XML Schema complexType definitions as restrictions of the MLHIM Reference Model complexTypes.

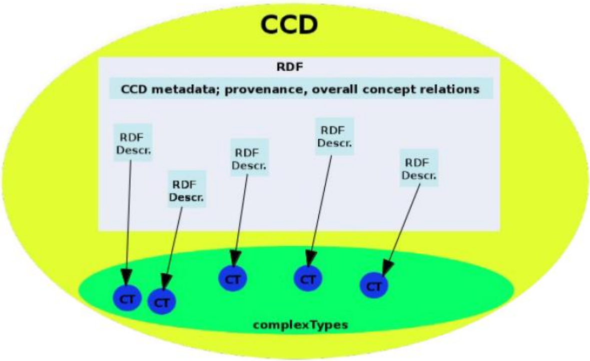


Figure 1. Structure of a MLHIM Concept Constraint Definition (CCD).

The light blue boxes represent Resource Description Framework (RDF) semantic links to definitions or descriptions of those complexTypes. RDF is a way to describe resources in a way that both humans and computers can interpret their meaning. RDF is a foundational component of the XML family for describing resources via URIs, specifically on the WWW. However, the concepts easily transfer to other environments and the technologies are well known. There are multiple syntaxes for presenting RDF. In MLHIM the RDF/XML syntax was adopted, to provide computability with the reference implementation.

The entire RDF section in a CCD is enclosed in an XML annotation by a starting, <rdf:RDF> and an ending </rdf:RDF> tag. This is the structural approach of all XML documents. A CCD is a special XML document called an XML Schema. An XML Schema defines the constraints to be placed on instance document of data contained in XML markup. Some examples of these constraints are: minimum or maximum value of a DvQuantityType, or string length of a DvStringType. It can also be a restriction on certain choices such as an enumerated list of strings of a DvStringType.

The CCD Metadata section describes the concept and provenance information for the CCD. It is located between the rdf:Description tags. It can be noticed that the tags all have two parts separated by a colon. The left side of the colon is referred to as a namespace. That can be thought of as the name of a vocabulary or a set of specifications. The right side is the element name.

It is also important to emphasize that every element name is unique within its namespace. This means that the same element name may be used in many different namespaces and still have different meanings.

In the CCD Metadata section there are tags that have a namespace 'dc:'. This is the Dublin Core namespace. The Dublin Core Metadata Initiative maintains an industry standard set of metadata definitions used across all industries. Therefore, any person or any application familiar with the DCMI standard will be capable of interpreting what is meant by the metadata entries in a CCD. Following and using industry standards is a foundation policy of MLHIM.

The two rdf:Description tags on the CCD display how the semantics of a PcT are improved. The rdf:about tag points to a PcT ID in the CCD, declaring 'what' is being described in this structure, and that description is 'about' this specific PcT. On the next line there is a rdfs:isDefinedBy tag, meaning that; in the RDF Schema namespace, there is an element that will be used to declare that this PcT is defined at this location or by this vocabulary and code. The rdf:resource tag is used to point to the resource for the definition. The description for this PcT is finally closed by the end tag. This structure appears consistently for all CCDs openly available at the Concept Constraint Definition Generator Library (www.ccdgen.com/ccdlib).

It is important to note that there can be several elements within a single rdf:Description tag set. This can alleviate the issues surrounding controlled vocabulary harmonization and mapping. By being performed at a single concept point, there is no doubt what is meant by the concept. In attempts at general mapping, it is often a matter of coarseness of the vocabularies as to whether or not the meanings actually correlate.

In MLHIM, the CCD knowledge modeler decides whether or not terms from different vocabularies represent what they intend to model. Thus, the MLHIM specifications help removing ambiguity in semantics. This is essential in healthcare, because it is not possible to achieve global consensus on all (or any) healthcare concept models [13]. In order to avoid semantic conflicts but at the same time that different medical cultures, schools and models are respected, the MLHIM eco-system allows for many different CCDs that model the same concept, even in slightly different ways.

Given the fact that MLHIM provides a common information framework against which any type of application can be built by independent developers, the type of syntactically coherent and semantically rich data generated by MLHIM-based applications can be regarded as 'hyperdata' [14]. The term 'hyperdata' is here proposed as an overcoming of the concept of Big Data, since the latter is based on conventional software and has created much more confusion and impossibilities than solid analytics in healthcare [15].

Big Data can be defined as a huge set of databases [16]. In healthcare, the level of complexity and heterogeneity of the distributed databases is such that querying the Big Data is not cost-effective and often inaccurate, since there are semantics missing and inconsistent structures across all of the databases included in any given Big Data set [17]. On the other hand, 'hyperdata' is a huge set consistently structured data, coming from any type of MMD-based healthcare applications.

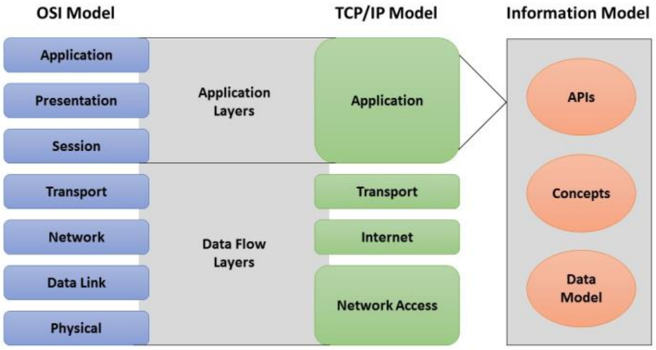


Figure 2. Analogy among the OSI, TCP/IP and Information Models.

For better clarification, Figure 2 displays the analogy among the OSI Model, the TCP/IP Model and the Information Model. It can be seen that the TCP/IP model aggregates the levels above the transport layer in one application level. On the other hand, the OSI model is more detailed on the communication layers. However, inside the application level there are the conceptual models that transforms the data into meaningful information. There are three components of the information model to take into consideration:

Data Model – The application data models, which is healthcare present extreme variability, Built upon ISO standardized datatypes, it allows machine processing and calculating.

Concepts – The conceptual models are needed in order to transform data into information. In human engineered domains these are typically well defined and semantics can be assumed even on a global basis, in many cases. In any of the sciences where evolution is involved in the engineering the approach goes from as simple, efficient and stable as possible (human engineered) to as complex and changing as necessary for survival. In the biosciences area, same or similar named concepts are actually interpreted differently and at varying levels of detail across different sub-domains and, often, in different cultures and even in different schools of training. Therefore these concepts must be well defined for the specific use intended and then be made available to every end-user of the data so that they can make the decision as to whether that data actually represents the information they need.

Application Programming Interfaces (APIs) – Consistent with any modern data exchange operation, there is a need for standardized APIs that can provide serializations, usually in JSON and XML formats.

The actual key to interoperability that is missing in todays' information system design is the ability to transfer the semantics of the concepts between applications. MedWeb has this capability through the use of the MLHIM technologies. This allows for machine based decision support and analysis vertically across individual records as well as horizontally across large datasets.

3. RESULTS

The MedWeb implementation is composed of the following structures: (1) the MLHIM Reference Model implementation in XML Schema 1.1; (2) the Patient and Provider profiles, modeled as CCDs; (3) a MarkLogic 7 database that provides data persistence and query built-in services.

The MarkLogic database stores data instances validated according to the correspondent CCD. The CCDs Schemas are valid according to the MLHIM Reference Model Schema, which is valid according to the W3C XML Schema 1.1 and XML Language specifications. Thus, as any other MLHIM-based application [9] MedWeb has a complete backward validation chain from data instance to the W3C specifications, provided by independent third-party tools such as the Xerxes and Saxon XML parser/validators. The proof of semantic interoperability achieved by the MLHIM specifications is demonstrated with simulated data automatically generated from a set of CCDs using oXygen and persisted into the an eXist database (https://github.com/mlhim/mlhim-emr) as a predecessor to MedWeb.

MedWeb applications that collect vital signs, using the Bluetooth$^{®}$ connected sensor on mobile devices, also capture contextual data, such as date and time, location, outside temperature. The data collected on these applications can be directly sent to MedWeb via a REST API, using a JavaScript Object Notation (JSON) representation instead of the XML. This is done to reduce the size of the message, which is feasible using ubiquitous XML technology, since it is a common development pattern to translate be-tween XML and JSON and back to XML, and there are open source tools readily available for this procedure. With the standard MedWeb REST API, it is possible to authenticate and authorize the user's connection, receive the JSON file, transform it to the XML representation, validate it against the CCD and return a status code that notifies the vital signs recording application that the data was received and added to the record.

Given the MMD level nature of the MLHIM specifications, the mobile application does not need to include the MLHIM Reference Model, the CCDs or XML data instances, producing valid JSON output directly instead. When the reference ranges or any other component of the information changes, or when the mobile device gets a new sensor array that also collects, for instance, humidity and air quality, the only requirement is to create a new CCD with the new syntax and semantics and generate a new format JSON file. When the MedWeb reports on these various data points across time it will know about the changes and report them all in their correct contexts. Fig. 3 shows the comparison of a portion of an XML instance with its transformation to the JSON equivalent.

Figure 3 displays the real configuration of MedWeb, operating with distributed XML databases in a cloud configuration. The MedWeb ecosystem is composed of Clients (patients, healthcare providers of all types, hospitals and clinics), which will access MedWeb via any of the front-end processes (a REST API, HTTP interface, SOAP XML message interface, authentication/authorization), also consisting of the external format to XML instance transformations. For instance, data in JSON format can be transformed back to the XML representation, validated against the CCD by the use of the MLHIM XML Instance Converter (MXIC) source code available at (https://github.com/mlhim/mxic) or any similar implementation. A status code is then returned to notify the application that the data was received and added to the record. Back-end processes have the primary functionality of data instance validation, as well as reporting, analysis and other preparation for presentation.

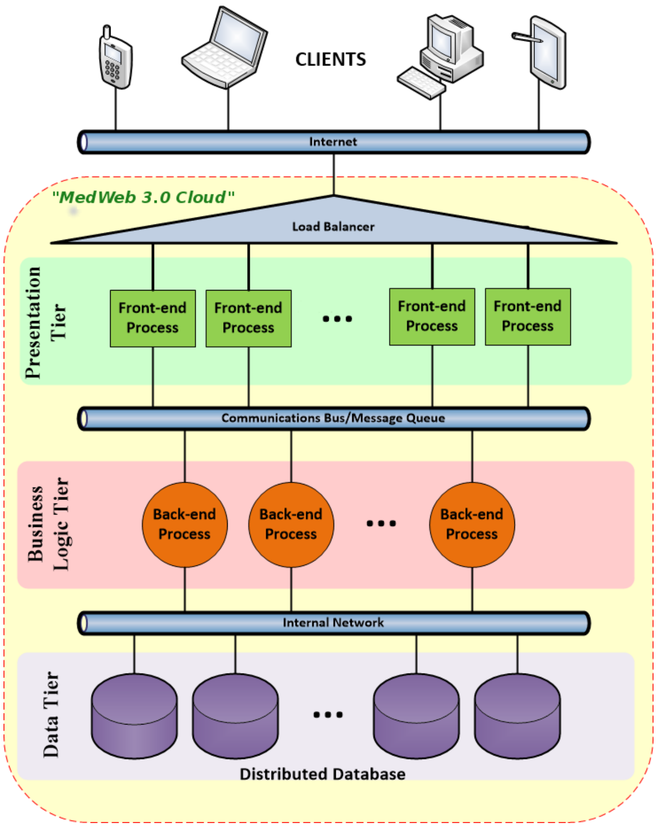


Figure 3. Schematic representation of the MedWeb ecosystem.

There are many user roles in this scenario and each role has *information to contribute* and *needs to be met*. These are not contrived for the purpose of MedWeb; those needs are currently expressed by the healthcare informatics community today. From this perspective, the actual role of MedWeb is to act as a barter mediator in this information exchange domain. Thus, it is relevant to define in an explicit way the roles, needs and contributions of each category of healthcare information user. Table 1 is a synthetic representation of such categories, associated to the correspondent solution proposed by MedWeb, in terms of technologies adopted for its implementation.

Table 2. Major categories of MedWeb users: roles, needs, contributions and solutions.

Role	Needs	Contributions	MedWeb Solution
Patients and Parents	Not to repeat form entry at every clinic To have each care giver know what the others are doing To have access to their own (or theirs child's) information	Can easily keep personal information up to date Can manage where all points of care are taking place	The patient is the center point of their information management
Healthcare Providers	To have access to their patients' data from any location To record the patient-related data according to their own expertise and clinical workflow	Can enter unbiased data about their patients Can improve scheduling and procedure management	The Domain Models underneath the professional profiles are MLHIM CCDs
Healthcare Institutions	To have opportune access to unbiased data collected at the point of care	Can create interfaces to the MedWeb for institutional use Can improve scheduling and procedure management	Access to anonimized data from REST APIs on MedWeb can be built for specific purposes
Researchers	To promote effective translational research based on biomedical research data coming from different sources	Can enter unbiased data about their research subjects Can make their anonimized data publicly available	MedWeb produces automatic UUIDs for each patient/research subject as well as maintains the data in an easy to anonymize infrastructure

4. DISCUSSION AND CONCLUSIONS

MMD is a solution for semantic interoperability of healthcare information systems, and it has been proven valid in software by independent researchers. The specifications adopted for the implementation of MedWeb present an industry standard, easily implementable, manageable way to develop semantically interoperable healthcare applications of any size.

Mobile health (mHealth) has been proposed as the solution of current healthcare IT shortcomings, which are (only apparently) related to the hardware support and the unfriendly user interface of Electronic Medical Records [18]. The current development of the mHealth technologies however, are showing that the same underlying problem is persisting, since the mHealth applications are unable to share data and their semantics are not transferrable from the original applications [19].

mHealth applications have the potential of giving the control of the information back to the patients, but it is essential to make this information shareable to the healthcare providers [20]. In order to achieve that goal, it is necessary to find a proper user interface that promotes sharing, and the social media architecture is fitted for that, since it has a wide acceptance by the general population [21]. Due to its features, the application of the social media approach to mHealth has been recently regarded as an important innovation with the potential to scale-up the compliance to mHealth [22] [23].

The current eHealth and mHealth scenario, where the challenge of achieving semantic interoperability among all the distributed applications recording data from patients following individual care pathways is the motivation for the development of MedWeb. For that to be accomplished, it was necessary to look at the standardized approaches to recording, storing and exchanging data and then improve the semantics of that data so that enough information is exchanged. Thus, the information receiver understands the same spatial, temporal and ontological concepts that were present at the moment the information was recorded.

While the information infrastructure of MedWeb, the MLHIM Reference Model, is a general-purpose model designed to be implementable in any programming language, the reference implementation adopted the constraints of the W3C XML specifications to insure the widest possible implementability, and XML Schema 1.1 was chosen to provide concrete evidence of functionality.

MedWeb can be regarded as the MLHIM-based application development framework for mHealth. At this point, there are development projects of purpose-specific applications for epidemics control and emergency case management that can also generate data extracts to be consumed by legacy systems, since it is possible to include data already persisted in conventional software to the MLHIM eco-system through MXIC and the MLHIM Application Platform & Learning Environment (https://github.com/mlhim/MAPLE). It is expected that those initiatives will expand the acceptance of the MMD principles by some new and innovative segment of the medical software industry, where conventional one-level 'data silos' [6] are still hegemonic.

It is expected that in the future, the best CCDs will be re-used and a large repository of publicly vetted CCDs would then emerge. However, MLHIM always allows the new models to be created as science changes, while the existing CCDs will be forever valid for any data instances created against them along with their specific RM version.

However, some issues are outside the control of the MedWeb eco-system. When knowledge modelers points to a controlled vocabulary or other resource as a semantic link for a CCD, they should choose the best quality resources available. Especially in the cases of controlled vocabularies (e.g., terminologies, ontologies, classifications), if the vocabulary is not well managed and versioned properly then the definition may disappear; or worse, be modified to change the meaning. If the vocabulary development organization does not provide version information and reuses codes with a different meaning this can cause semantic conflict. Thus, best practices for knowledge modeling of CCDs are always encouraged.

In the process of implementing MMD-based solutions for healthcare IT, healthcare professionals and computer scientists increase the dialogic interface between their domains. In consequence, the wider adoption of MMD will produce a new hybrid expert, and then healthcare knowledge modeling will emerge as a new area of expertise for the both scientific fields involved in the development of MedWeb applications.

5. ACKNOWLEDGMENTS

Our thanks to the National Institute of Science and Technology – Medicine Assisted by Scientific Computing, for partial financial support.

6. REFERENCES

[1] De Leon, S., Connelly-Flores, A., Mostashari, F., and Shih, S. C. 2010. The business end of health information technology. Can a fully integrated electronic health record increase provider productivity in a large community practice? *J. Med. Pract. Manage.* 25 (May 2010), 342-349.

[2] Sittig, D. F., and Singh, H. 2010. A new sociotechnical model for studying health information technology in complex adaptive healthcare systems. *Qual. Saf. Health Care* suppl. 3 (October 2010), i68-i74. DOI= 10.1136/qshc.2010.042085

[3] Alsos, O. A., Das, A., and Svanæs D. 2012. Mobile health IT: the effect of user interface and form factor on doctor-patient communication. *Int. J. Med. Inform.* 81 (January 2012), 12-28. DOI= http://dx.doi.org/10.1016/j.ijmedinf.2011.09.004

[4] Maojo, V., and Kulikowski, C. 2006. Medical informatics and bioinformatics: integration or evolution through scientific crises? *Methods Inf. Med.* 45 (September-October 2006), 474-482.

[5] Hyman, W.A. 2010. When medical devices talk to each other: the promise and challenges of interoperability. *Biomed. Instrum. Technol.* suppl. (2010), 28-31.

[6] Raths, D. 2010. Shifting away from silos. The interoperability challenges that hospitals face pale in comparison to the headaches plaguing State Departments. *Healthc. Inform.* 27 (January 2010), 32-33.

[7] Kadry, B., Sanderson, I. C., and Macario, A. 2010. Challenges that limit meaningful use of health information technology. *Curr. Opin. Anaesthesiol.* 23 (April 2010), 184-192. DOI= http://dx.doi.org/10.1097/ACO.0b013e328336ea0e

[8] Blobel, B. 2011. Ontologies, knowledge representation, artificial intelligence: hype or prerequisites for international pHealth interoperability? *Stud. Health Technol. Inform.* 165 (2011), 11-20.

[9] Cavalini, L. T., and Cook, T. W. 2012. Knowledge engineering of healthcare applications based on minimalist multilevel models. *Proceedings of the IEEE 14th International Conference on e-Health Networks, Applications and Services* (Beijing, China, October 10 - 13, 2012). HealthCom 2012. IEEE, Piscataway, NJ, 431-434. DOI= http://dx.doi.org/10.1109/HealthCom.2012.6379454

[10] Kalra, D., Beale, T., and Heard, S. 2005. The openEHR Foundation. *Stud. Health Technol. Inform.* 115 (2005), 153-173.

[11] Marley, T. 2002. Standards supporting interoperability and EHCR communication: a CEN TC251 perspective. *Stud. Health Technol. Inform.* 87 (2002), 72-77.

[12] Cavalini, L. T., and Cook, T. W. 2014. Use of XML Schema Definition for the development of semantically interoperable healthcare applications. *Lect. Notes Comput. Sci.* 8315 (2014), 125-145. DOI= http://dx.doi.org/10.1007/978-3-642-53956-5_9

[13] Shalom, E., Shahar, Y., Taieb-Maimon, M. , Martins, S. B., Vaszar, L. T., Goldstein, M. K., Gutnik, L., and Lunenfeld, E. 2009. Ability of expert physicians to structure clinical guidelines: reality versus perception. *J. Eval. Clin. Pract.* 15 (December 2009), 1043-1053. DOI= http://dx.doi.org/10.1111/j.1365-2753.2009.01241.x

[14] Kopecky, J., Pedrinaci, C., and Duke, A. 2011. RESTful write-oriented API for hyperdata in custom RDF knowledge bases. *Proceedings of the 7th International Conference on Next Generation Web Services Practices* (Salamanca, Spain, October 19 - 21, 2011). NWeSP 2011. IEEE, Piscataway, NJ, 199-204. DOI= http://dx.doi.org/10.1109/NWeSP.2011.6088177

[15] Webster, P. C., and Kondro, W. 2011. Medical data debates: big is better? Small is beautiful? *Can. Med. Assoc. J.* 183 (March 2011), 539-540. DOI= http://dx.doi.org/10.1503%2Fcmaj.109-3799

[16] Jacobs, A. 2009. The pathologies of Big Data. *Queue – Data* 7 (July 2009), 1-10. DOI= http://dx.doi.org/10.1145/1563821.1563874

[17] Cheung, K. H., Prud'hommeaux, E., Wang, Y., and Stephens, S. 2009. Semantic Web for health care and life sciences: a review of the state of the art. *Brief. Bioinform.* 10

(March 2009), 111-113. DOI = http://dx.doi.org/10.1093/bib/bbp015

[18] Shaw, N. T., and Bainbridge, M. 2013. Computerisation in general practice: lessons for Canada from the UK and Australia. *Stud. Health Technol. Inform.* 183 (2013), 28-36. DOI= http://dx.doi.org/10.3233/978-1-61499-203-5-28

[19] Morrissey, J. 2014. Regulatory clouds part for mHealth apps, but barriers for full integration remain. *Hosp. Health. Netw.* 88 (February 2014), 22-23.

[20] Germanakos, P., Mourlas, C., and Samaras, G. A mobile agent approach for ubiquitous and personalized ehealth information systems. *Proceedings of the Workshop on 'Personalization for e-Health' of the 10th International Conference on User Modeling* (Edinburgh, Scotland, July 29, 2005). UM'05. Springer-Verlag, Berlin Heidelberg, 13-24.

[21] Duggan, M., and Smith, A. (2014). *Social Media Update 2013*. Technical Report. Pew Reaearch Center.

[22] Estrin, D., and Sim, I. 2010. Open mHealth architecture: an engine for health care innovation. *Science (Washington)* 330 (November 2010), 759-760.

[23] Tomlinson, M., Rotheram-Borus, M. J., Swartz, L., and Tsai, A. C. 2013. Scaling up mHealth: where is the evidence? *PLoS Med.* 10 (February 2013), e1001382. DOI= http://dx.dor.org/10.1371/journal.pmed.1001382