The Semantic Logger: Supporting Service Building from Personal Context

Mischa M Tuffield
Intelligence, Agents, Multimedia Group
University of Southampton
Southampton, SO17 1BJ
mmt04r@ecs.soton.ac.uk

Antonis Loizou
Intelligence, Agents, Multimedia Group
University of Southampton
Southampton, SO17 1BJ
al05r@ecs.soton.ac.uk

David Dupplaw
Intelligence, Agents, Multimedia Group
University of Southampton
Southampton, SO17 1BJ
dpd@ecs.soton.ac.uk

ABSTRACT

The Semantic Logger [1] (SL) is presented as a system for the importing, housing, and exploiting of personal information. The system has been implemented using a number of Semantic Web enabling technologies, and attempts to store the information in a manner adhering to as many W3C recommendations as possible. The Semantic Logger's utility is grounded in two context-based applications, namely a recommender system, and a photo-annotation tool.

Categories and Subject Descriptors

H.3.1 [**Information Storage**]: Record Classification; H.3.3 [**Information Search and Retrieval**]: Query Formulation; H.3.4 [**Systems and Software**]: Information networks; I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods—*Relation systems, Semantics.*

General Terms

Design, Human-Factors, Standardization

Keywords

[Semantic Logging, Context, Lifelogs, Recommender Systems, Photo Annotation, Multimedia, Ontologies, Semantic Web, Memories for Life]

1. INTRODUCTION

In this paper we describe an auto-biographical metadata acquisition system, along with two services that run on top of the architecture, utilising the context assembled. The research aims to highlight readily available sources of information, and defines an approach for its surreptitious integration into a standard and web accessible form that also builds on the Semantic Web vision.

Our system builds on the ideas brought forward in the original *Scientific American* Semantic Web article [6], [32] with a particular focus on the notion of assembling, and integrating web accessible resources. At his keynote speech during the International Semantic Web Conference 2003 [4] Tim Berners-Lee identified the *'Killer App for the Semantic Web'*, not as a single application but the successful integration of information, or to use his blunt words, *'Its the integration, stupid!'*.

In an attempt to avoid sounding too evangelical we will present work that integrates a number of sources of information (identified in Section 5), to build up personal metadata. The Semantic Logger allows for users to select how much information they wish logged. Since this is a matter of preference it can not be dictated, even though it may directly affect the richness of the metadata assembled.

The Semantic Logger is intended as a contribution to the current research programme of *Memories for Life*[2] (M4L). M4L focuses on the use of technology to support human memory, and we believe that our system can be seen as a step in that direction. SL is a piece of social software, it allows people to share their metadata, and to reap the added value of exposing it to a community. The adoption of a *lingua franca*, in the form of widely used Resource Description Framework (RDF) [23] representations, allows for community based recommendations to be made from various sources.

The system builds upon a number of existing semantic web enabling technologies (see section 4), attempting to adhere to as many W3C[3] recommendations as possible. The Semantic Logger can be seen as a means to populate the Semantic Web with personal metadata, by exposing information in a structured form, i.e. by using RDF accessible through SPARQL endpoints [37]. The system uses a Universal Resource Identifier (URI) to point to a user's Friend of a Friend (FOAF) file. In the case that the user does not have a FOAF file the system will generate a basic one upon registration, allowing them to edit it as they see fit. Each users FOAF file serves as a unique identifier for their RDF data. The user's FOAF URI is employed to log the provenance of all the information asserted in the Semantic Logger.

[1] http://akt.ecs.soton.ac.uk:8080

[2] http://www.memoriesforlife.org

[3] World Wide Web Consortium (W3C) http://www.w3.org

CARPE'06, October 28, 2006, Santa Barbara, California, USA.

In the next section, we specify our motivations and visions for the Semantic Logger. Following that is a related work section where the differences between the Semantic Logger and other systems found in the literature are presented. The remainder of the paper sets out to describe the architecture and functionality of the Semantic Logger.

Section 6 highlights the utility of the services that exploit the infrastructure of the Semantic Logger. A photo annotation system, photocopain [35] has been adapted to work off the Semantic Logger, and a recommender system [21] is also realised using the Semantic Logger's capabilities.

2. MOTIVATION

In an attempt to realise some of promises roadmapped by the Semantic Web community: the seemless integration of heterogeneous data, and that of services exploiting existing machine-accessible knowledge [32], a decision to create an easy to use system architecture that allows users to store, update, and query their own knowledge base(s) through the web, seemed a pragmatic course of investigation.

Upon registration of a Semantic Log, a user is presented with tools that allow for the surreptitious recording of personal information. The presented list of information sources is far from an exhaustive one, and is not intended to limit the functionality of the system. The Semantic Logger has been designed in a manner to allow information, in various forms of RDF to be posted to the knowledge base (KB). The sources of information we have identified and implemented are rationalised by the nature of the services currently provided by the system, and are merely presented as inspiration for future development.

Given the abundance of personal information being posted to the web, backed up by the current trend of publishing to social software sites like, del.icio.us [4], flickr [5], Last.fm [6], Plazes [7], etc, and the number of people adopting shared vocabularies to document certain phenomena is increasing by the day. This apparent willingness to post personal information on the web, was a key driver in the development of the Semantic Logger.

The Semantic Logger aims to aggregate as much available personal information into a central knowledge base allowing for context-based systems [13] [10] [1] to exploit as needed. This notion of aggregation is grounded in two services that utilise this heterogeneous knowledge base (see section 6). It comes without saying that such a system can never be omniscient, our aim is to identify how much knowledge we can generate through the integration of as many sources of information. It is important to stress that a user is not required to expose all of the personal information presented below.

The sensitive nature of this metadata-chronology being accessible, along with the current trend in using web-based social software, implied that the Semantic Logger had to allow users to decide whether any information logged was to be posted for public consumption or not. This guided the design such that each user is provided with two different knowledge bases: a public, and a private persona. A user's private knowledge base is presented as a means to enrich their own media.

One of the Semantic Logger's long-term visions is to provide a solid platform for evaluating the approaches of the authors' respective long-term research interests, that of autobiographical metadata and recommender systems. However, we argue that by virtue of knowledge integration alone, added value emerges.

The principal support for this argument stems from the power of enabling the application of SPARQL queries on the available information, to answer questions that would be unfeasible under representations of singular domains, and also the added inferential capabilities that are enabled. For example:

- Queries

  How many users of the system attended the same events as me between time X and Y?

  This can be achieved by first selecting all events attended by the user between X and Y, using the iCal data, and then selecting all users with similar entries. If geo data is also available, it can be used extend and target the query.

  How many hyperlinks did I receive in email correspondence that I have yet to visit?

  A single query can be used to tackle this, by querying the email and browser history representations.

  What document was I reading on the way to event X?

  What was the name of the band I discovered while on holiday in Y?

  This can be seen as the first step in the development of a queryable personal memory store.

- Inference

  The system does not require the user to produce hand crafted annotations. The existence of various domains in the knowledge base supports the automatic creation of such metadata. For example, iCal entries referring to the same time period as GPS location data can be used to provide suggestions for the name of the place with the specified coordinates, with respect to the current user.

The importance of exposing the system as SPARQL endpoints, allows for applications to exploit the knowledge in unforeseen ways. Our attempts to comply with as many W3C recommendations as possible and this promotion of interoperability is proposed as a contribution to the Semantic Web vision.

3. RELATED WORK

MyLifeBits [14] and SemanticLIFE [2] can be regarded as the modern seminal systems in this area building on the

[4]http://del.icio.us/
[5]http://www.flickr.com
[6]http://www.last.fm
[7]http://www.plazes.com

ideas put forward by Vannevar Bush[7] in his Memex device. While these have proven to be a valuable source of inspiration for this project and numerous others [8, 17, 18, 27, 28, 38], the domain of interest of such systems is limited to the publishing, browsing and sharing of information in prespecified formats.

SemanticLIFE is preoccupied with allowing users to set-up an information repository to provide enhanced querying capabilities, while MyLifeBits introduces the notion of automatically producing annotations by exploiting co-occurent events. In this section we set out to identify the principal differences between the Semantic Logger and such previously developed systems. For a full overview of the state of the art in developments in the field the interested reader is pointed to [30].

Such systems have engineered over-ranging knowledge representations to support the functionality they provide. The Semantic Logger makes no attempt to homogenise data that is heterogeneous by nature; this is left for applications that will use the system as a platform, as per their requirements. The rationale is that different mappings will be appropriate for different applications, as exemplified in the remainder of this paper.

Another development worth mentioning is the NEPOMUK project [8], a EU FP6 funded collaboration of industrial and academic partners and industrial end-users. The project brings together various previous semantic desktop implementations, and focuses on knowledge integration in shared peer-to-peer environments, supporting automated community recognition. Detailed information on this has not been made available, however it seems that the focus is put once again in providing a solid platform for such sharing, rather than the ease of adding services to the system.

4. INFRASTRUCTURE

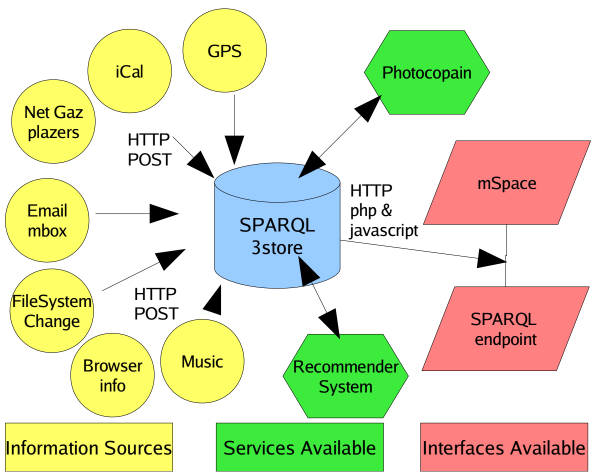

Figure 1: Overview of the Semantic Logger architecture

The Semantic Logger system has a service-based architecture, as shown in Figure 1, and has been designed so that new services may join on an ad-hoc basis. The interactions between Web Services have been implemented using HTTP requests, while the interactions with the central RDF triplestore make use of the SPARQL RDF query language [37].

At the heart of the system is the AKT Project's [9] SPARQL-compliant RDF triplestore 3store [16]. The key role of the triplestore is to act as a persistent store for the system, and to mediate the interactions between the other system components. The main requirements in selecting an appropriate RDF Knowledge Base implementation were efficiency and consistency. 3store is a system benchmarked against other RDF storage and query engines such as Jena [25], Sesame [5] and Parka [33] and shown to outperform in terms of both efficiency and scalability [34] [20].

The cornerstone in designing this architecture has been to develop a very open system, so that third parties can exploit the knowledge stored. We have chosen to expose two distinct methods of interacting with the system, namely in a public and private fashion. A number of distinct knowledge bases are maintained: A system-wide shared one, and one for each user, created automatically upon registration. When information is imported into the system, users are able to specify whether or not it should be publicly accessible. If this is the case, the information is added to both the shared and private ontologies. Both are exposed through web-based user interfaces to allow SPARQL queries on the data and the import of new knowledge. Furthermore, user interfaces have been designed to support automated query construction, as we cannot expect users of the system to be fluent in SPARQL.

When data is represented in an RDF graph, by virtue of the representation there exist multiple dimensions in which the data may indexed and viewed. The mSpace interface [22] has the ability to organise such data, in multipane browsers. In addition, the edges of the graph are allowed to be reordered, using dimensional sorting independent of the hierarchical nature of the representation, allowing for a number of such trees to be visualised and browsed.

The software has been considered a good opportunity for visualising the data gathered by the Semantic Logger system, as it is currently being released with a comprehensive API with the sole requirement of a SPARQL interface.

mSpace requires the definition of a *default column* and a *target column* along with the path, through ontological relationships (edges in the graph), between them to create a multi-columned re-arrangeable browser. While in the current implementation these have to be made explicit by the system engineer, we are currently in the process of automating the procedure, and allowing users to choose these columns. Furthermore, it is important to stress that this browsing ability is greater than that achieved through representations of singular domains, since the all the information logged by the system will be interconnected in automatically inferred or hard-coded ontologies.

It is crucial for the Semantic Logger to impose the mini-

[8]http://nepomuk.semanticdesktop.org

[9]http://www.aktors.org/

mum burden on a user joining the system. Focus has been placed on allowing the import of knowledge described in heterogeneous, widely used vocabularies, to avoid the need for prior semantic agreement. The lack of an overall representation however, introduces the need for alternative means of knowledge integration. Where it is possible, this is to be achieved via automated means, such as the S-MATCH algorithm, developed by the University of Trento [15]. Alternatively, where disagreement is too complex to be resolved in an automated fashion, mappings will be hard-coded into applications that use the Semantic Logger as a knowledge source, in *ad-hoc* fashion as per their requirements.

The richness of the metadata acquired, enables the system to be used as a platform for community of practice identification. For example, named entity recognition can be applied to email correspondence to identify closely related groups while co-authorship and co-reference of scholarly articles can be analysed as shown in [3]. Co-location at various events can be inferred from geo-data and calendar entries, while the latter, in combination with the analysis of locally stored multimedia files (e.g. music and video files) can aid in identifying common interests.

A final feature of the Semantic Logger worth mentioning is the way the logger makes use of the FOAF model. A user's FOAF file, is used to allow a user to publish data about themselves, using a URI, allowing for the user's data to be referred to from any dataset, or from within any context. Another advantage of the adoption of personal FOAF files is the ability for a user to define his/her friends, allowing for further connections to be made when using the system to identify communities of practice.

5. KNOWLEDGE ACQUISITION

The Semantic Squirrel Special Interest Group (SSSIG) [10] is a group of researchers based at the University of Southampton who aim to automate the process of logging available raw data, (or *'nuts'*), that can describe aspects of one's personal experience. A number of squirrels have been developed in this process, and an ethos of the group is to preserve this raw data in order to retain any unforeseen potentials for exploitation and transcend issues pertaining to platform and application restrictions. The SSSIG is also focusing on identifying novel systems using the collected data.

This raw data forms the basis of the knowledge acquisition phase for the Semantic Logger and is parsed into RDF representations. Effort has been put in selecting appropriate representations: they have been taken from proposed standards at the W3C or other standard making bodies, or have been selected due to current uptake on the web. Where such standards have not been available, we constructed local ontologies which describe the given phenomenon[11], while simplicity and generality maintenance have been paramount. The intent is to use raw data about people in order to build the context of a particular event at a particular time. By virtue of the fact that each event logged by the system is time-stamped and related to a FOAF URI, we are able to choose variable levels of granularity to describe its context.

[10]http://www.semantic-squirrel.org/

[11]http://akt.ecs.soton.ac.uk:8080/downloads.php

We collect, and propagate the following types of *'nuts'* into RDF representations:

- Calendar entries

  We have adopted the W3C recommendation in representing calendar entries in RDF [12]. A client-side application is available for download from the Semantic Logger site to automate the export of iCal [11] files (commonly used and platform independent) into this representation. In addition to querying capabilities as before, calendar entries can serve as context indicators for geographical locations (described below), enabling to an extend the resolution of co-location.

- Geo-data

  In an attempt to build up a log of a user's geographical data, we take a two pronged approach. For research purposes we have been carrying around GPS units to log our data. The information is extracted and parsed into an RDF representation, taken from *http://www.hackdiary.com/*. The RDF model builds ontop of the dublin core namespace[13], and W3C's recommendation for geographical data[14].

  GPS information is being used to track a user's change of location, but is not always a suitable method of tracking, for tall buildings, and movement between buildings within close proximity is hard to track, so a decision was taken to start employing a network gazetteer. The network gazetteer Plazes is currently being employed by the Semantic Logger. Plazes supplies the end user with client side applications that pick up a laptop's current network connection and provides information about the location if information has been entered for that wifi hotspot. Plazes provides a comprehensive API, and RSS 1.0 feeds, that export parsable RDF, of a users activity. We have taken the decision to adopt their namespace for the purposes of logging network activity.

  The combination of the GPS information, a user's network gazetteer (given that the user has a laptop computer), and a his/her iCal file, along with the Getty Geographical Name gazetteer, allow us to infer a user's geographical context.

- Music playcount statistics

  Audioscrobbler [15], is a music search engine based on a large collection of music profiles. These profiles are built through allowing the users to download and install plugins to their respective media players that propagate the information to the system. The representation used to describe artists is Musicbrainz [16], a freely accessible knowledge base for the music domain, that publishes the data in their ontology in an attempt to provide a comprehensive music information service. These systems are currently in the process of developing their metadata vocabulary to be published in an

[12]http://www.w3.org/2002/12/cal/ical

[13]http://dublincore.org/documents/dces

[14]http://www.w3.org/2003/01/geo/

[15]http://www.audioscrobbler.net

[16]http://musicbrainz.org/mm/mm-2.1

ontology. In the interim phase, we developed a local version to describe the data made available through their web-service API.

- Firebox bookmarks, downloads and navigation history

  By virtue of its cross-platform nature, we have selected the Mozilla Firefox as our web-browser of choice. Firefox exposes the download information in RDF form [17] and thus can be easily imported to the system. Scripts have been developed to parse the bookmarks and history data into RDF. The RDF model uses two namespaces taken from the mozilla developers centre [18].

- Email

  A simple ontology has been constructed to describe email correspondence [19] as one of satisfactory quality has not been found to be readily available. We have developed a client-side application to parse and convert the widely used MBOX representation into the local format. The intended use of this information, in addition to the ability to query one's records, is to support the identification of communities of practice, under a predefined temporal context.

- File System Information

  Beagle[20] search indexes every file found on a user's computer. This is achieved by combining specialised analysis tools for extracting content from different file types. This creates a personal information space describing a computer at the file-system level. The information is parsed into a simple ontology and can be loaded into a user's SL. This enables services to detect the presence and usage of files, giving an indication to a user's interests.

6. SERVICES

The following section presents two services that integrate the information stored in the Semantic Logger. Firstly, a recommender system is presented that generates recommendations by utilising any relevant context stored in the Semantic Logger. Secondly, we elaborate on a system, that uses flickr as a datasource, and that combines content and context based information to propose annotations of one's personal photos.

6.1 Recommender System

As the breadth and depth of freely available content on the Web is ever expanding, the problem of information overload, as identified in [26] and numerous others, is a critical one. The problem of efficient search and retrieval of required information is a research area in its own right, and great effort has been applied into identifying the user's intent, via extrapolating from the supplied textual query strings [19]. However, to assume that people's awareness of their individual information needs extends to a degree at which they are able to formulate *contentual* queries to retrieve it is optimistic. This is demonstrated by considering the way information is exchanged between humans in conversation. This

[17]http://home.netscape.com/NC-rdf#

[18]http://developer.mozilla.org/en/docs/XUL_Tutorial:-RDF_Datasources

[19]http://semanticlogger.ecs.soton.ac.uk/email/#

[20]http://beagle-project.org

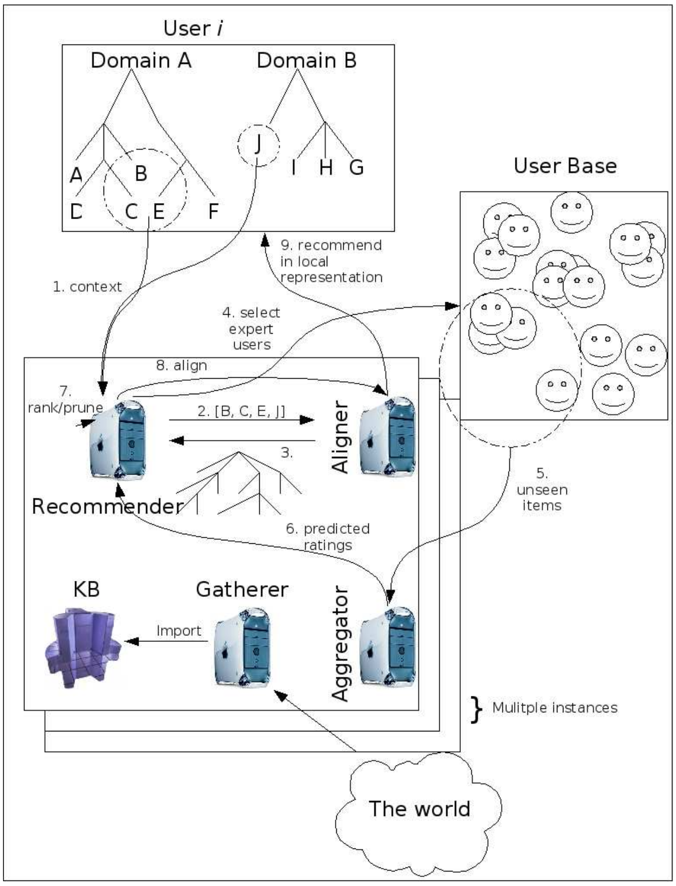

Figure 2: Overview of the recommender system architecture

is seldomly achieved by a single, targeted question and its subsequent answer, but rather via communicating interests through a series of statements, to enable the answering party to apply their expert knowledge to suggest possible answers.

We argue that through the knowledge published via the Semantic Logger, a system would be able to gain deep knowledge of users and their needs and apply this knowledge to assess the utility of resources to users. Such a system resides in the research domain of Recommender Systems (RS), since it would rely on accurately predicting user reactions to unseen and unsought items. However, the system we intend to deploy on top of the infrastructure described in this paper has significant differences with conventional RS implementations.

With the recent explosion in e-commerce applications and the appearance of meta-sites that aggregate product catalogues from multiple sources, the need for technologies to effectively emulate the way human sales assistants facilitate the sales process has become critical [9]. It is important to state that since the object of interest here is the prediction of human preferences, the concept of 'ground truth' is highly variable and subjective. The feature space representation of artifacts and users, however, is usually static in conventional RS implementations. We believe this to be a cause for a number of weaknesses in current implementations. The main source of motivation in developing such systems, at least on a commercial level, has been to push products to appropriate customers, based on their previous purchase behaviour within the system. Here, we intend to learn user preferences from any information they choose to share and recommend arbitrary *resources* from the systems ever expanding knowledge base.

As shown in [1], by expanding the feature space representation to include contextual dimensions, allows for relatively dense subspaces (or *'slices'*) to be extracted each time a recommendation is made, enabling more targeted recommendations. We are developing a system to extend on that idea, described in [21] along with preliminary evidence demonstrating the merits of this approach. This allows the use of an arbitrary number of such dimensions in the recommendation process, in an opportunistic fashion. The research aims to address the following problems:

- The 'cold start' problems

  As most users are not inclined to rate previously seen items, only a few items will receive ratings. This limited data – the 'cold start' problem [31] – renders similarity metrics not sensitive enough to distinguish between users, particularly new ones introduced to the system. Hence, the most highly rated items from anyone are recommended, independent of the user. Our approach addresses the cold start problem by defining a more strict process in identifying appropriate groups of experts. This is the case since the recommendation context is obtained through the users local representation which is likely to differ between users. As such only users with ontologies similar enough to be mapped onto each other are considered, and the effect of such issues is expected to be milder. In addition, new users do not enter the system with empty profiles, since multiple domains of interest are considered and the knowledge acquisition takes into account local information already present on the user's node.

- The most similar items are not always good recommendations

  Another drawback is that items interchangeable with the ones rated highly by users can be recommended, by virtue of some systems' focus on items' features, ignoring potential user requirements. As such, for a system to be able to avoid such issues, equivalence (or subsumption) between items, under particular contexts, needs to be evaluated. If equivalence between items can be assessed during the translation phase required to make recommendations, the system will be able to avoid making interchangeable recommendations of no merit to the user.

- Shifts and temporal cycles of user interests

  Most conventional RS architectures do not encode for shifts of the user's interest over time, since all ratings provided by a user have an equal bearing on the recommendation selection. To clarify this point consider the following conceptualisation: A user X has provided high ratings only for items in some set A, however s/he is now only interested in items from another set, B. A conventional RS will not be able to recommend items from set B until enough ratings are provided for items in B, in order for them to dominate in the clustering and selection processes. It may even be the case that X is only interested in items from A during the weekend, while only items from B are of interest during the week. This means that a system should not become stable, and that the classification of the same items to different classes, at different times, may be deemed correct, something that would be unacceptable in most machine learning contexts. To account for this requirement of preference time dependence, conventional architectures recompute their user clusters periodically, effectively choosing a different training set every time. This can aggravate problems caused by data sparsity, and important modelling decisions about transitions between user needs have to be addressed. We address this problem in two different ways: By either efficiently identifying the change in the recommendation context and configuring the system appropriately, or by allowing the user to form a specialised query that explicitly restricts the recommendation space.

- Feature weights are assigned independently of context

  While it is apparent that an artifact's features have a bearing on whether it appears interesting or not, users may not be able to identify its desirable characteristics at the outset. For instance, someone who wants to buy a new car might only specify 'I want a black car' to begin with. Instead of buying the first black car available, s/he might look at a variety of black cars and as their knowledge of cars grows in the process, discover other possible features of interest, or even come across an unusual opportunity and end up buying a different coloured car. This would suggest that for a RS to be successful, it needs to be able to identify which of an item's features may potentially be of interest to the user, against a variety of possible modes of generalisation. This is remedied via hard-coding feature weights for specific recommendations in specialised components, or alternatively through inference on the number of experts who record such features.

- Only items described in one pre-specified representation are considered

  Since the focus in RS applications has been to enable organisations to suggest appropriate items from their catalogue to customers, not much effort has been put into learning user preferences based on the items they already have in their possession, regardless of their origin. However, a good sales assistant in a clothing shop will first look at what the customer is wearing before making suggestions. Conversely, by enabling arbitrary items to be imported in user profiles to make them more representative of the user, a system also gains the ability to assess such artifacts for recommendation.

It is assumed that the human selection process is better modelled through a dynamic function that operates on some weighted subset of an artifact's physical and contextual attributes. Defining this subset statically at the outset is expected to have a negative effect on the recommendation quality. To rectify this, the RS architecture employs a variety of components, each capable of performing a subroutine of the recommendation process. These are then combined at runtime to produce recommendations. Their performance is logged and determines the conditions under which the component may be used in the future.

- Clustering algorithms

  Clustering algorithms will be used to partition the dataset into groups of similar items and users. For users, this is achieved through exploiting subsets of the information available in their profile, while in the case of items the clustering is carried out by considering a subset of the descriptive features available for them. A wide variety of such algorithms is needed to facilitate the architecture and instances are chosen based on their past performance under similar contexts, as logged by the system. The Semantic Logger infrastructure allows for a number of novel approaches to such clusterings, such named entity recognition in email correspondence, co-authorship, co-location inferred from GPS data, event attendance from calendar entries, file system similarity and so forth.

- Recommenders

  These are the components responsible for evaluating the context of a recommendation need and for selecting the components that will be used to produce that recommendation. Recommenders will also receive predicted ratings computed by aggregators and augment them according to the recommendation context. Different recommenders may use other component selection, and ranking strategies to improve performance in specific contexts. The bias in choosing a particular instance is again determined by its past performance.

- Aggregators

  Aggregators are components able to combine user ratings for particular items. It is expected that under different contexts the degree to which the system can allow disagreement between the domain experts may vary. As such, multiple aggregators are required and the selection of a particular one is dependent on how well it has performed in the past, under a similar context.

- Classifiers

  Having defined a recommendation context, the need to assess which items are appropriate under that context arises. This is a binary classification task and since the items in question may originate from different domains, a collection of such classifiers is required. Where there is more than one classifier suitable for the task, the current context and its past performance guide the selection process.

- Ontology aligners

  Heterogeneity exists between the representations of different types of resources. In order to assess similarity the system will need to acquire the relevant partial translations from those representations to a temporary shared one, which will be discarded after the process is facilitated. Since it would be unfeasible to define a representation to which any user-defined ontology can be translated to, a variety of such components will be implemented to enable different modes of generalisation or specialisation.

The space of recommendable items is identified with a linear space, with a descriptive feature labelling each dimension, and the semantics of putative similarity expressed as some metric in this space. The subspaces relevant to an item of interest in the current context are identified by clustering techniques based on such metrics. A metric is evaluated as follows: Each unique feature is assigned a weight based on how strong the relationship between items with the same feature value is by applying Ontology Network Analysis (ONA) [3, 26], on the RDF graph published by each user. ONA is a set of graph edge expansion heuristics to assess distance between instances in an ontology by evaluating how well connected they are [3]. We intend to use such methods to identify which other concepts are closer to the recommendable ones with respect to each feature. The features which express indirect relationships between items in the recommendation subspace are preferred and receive higher weights.

The recommendation context is determined through the user-'s recent behaviour as logged by the system, inferred restrictions from long-term observations of user preferences, additional restrictions provided explicitly by the user, and global trends. This provides added leverage in identifying shifts of interest or locating specific domains from which recommendations are to be drawn. A suitable user cluster is then identified by selecting users with experience of items that fit this context and also with sufficient overlaps between their profiles and that of the active user, in order to meaningfully assess similarity between them. These users are viewed as the group of domain experts who are able to communicate best (in terms of their personal ontologies) with the active user.

6.2 Photocopain

In Tuffield *et al* [35] the Photocopain photo annotation system is presented as a stand alone system that utilises context and content based methods to generate metadata to enrich one's personal photo-collection. The semi-automatic nature of the service is stressed, identifying the need to allow a user to author any proposed annotations, highlighting the *'Gold Standard'* of any manual annotations. The integration of a number of sources of highly heterogeneous data, along with the combination of low-level content based feature vectors, allows us to suggest annotation to the user. Ultimately, our plan is to discover relationships between photos, based on the meta-data assembled by photocopain and stored in the SL to generate an autobiographical narrative.

Personal photographs are seen as important digital additions to the human memory store. Given the pleasure surrounding the browsing of one's photo-collections, the enrichment based on the photographer's context is presented as a method of enhancing search and retrieval. Furthermore, as digital technology has dramatically increased the numbers of photographs taken (it has been estimated that up to 375 petabytes of information is created in photographic form annually), there are obvious problems with determining the context of individual photographs, especially due to the fact that people do not tend to have time to manually annotate all of them.

The remainder of the discussion focused around the photocopain system, will include a list of information sources utilised, and will be followed by an insight into the advances made to the system since its last report. As with the Semantic Logger, photocopain performs best when presented with many sources of information. The utility of photocopain running off a user's Semantic Log will be proportionate to how much knowledge is stored in the personal KB.

The following sources of information are harnessed by Photocopain:

- Camera metadata

  Exchangeable Image File (EXIF) [12] metadata records camera parameters at the time that the photograph was taken. These parameters include: aperture setting; focal length of the lens; exposure time; time of photo; flash information; camera orientation (portrait/landscape); and focal distance. We can also derive other information from these metadata, such as the average scene brightness of an image. The EXIF is extracted from the images, presented to the photocopain system, and then uploaded to the 3store, in a RDF representation.

- Global positioning data

  GPS data can be recorded in EXIF if the camera is equipped with the required hardware, or alternatively a GPS tracklog matched with a photos timestamp can be used to determine location accurately. This is primarily of use when the camera is used outdoors. As described in *'Geo-Data'*, section 4 the GPS, the Network Gazetteer, the Getty Gazetteer, and the iCal information can be used to piece together a geographical log of a given user. All of aforementioned sources of information can be found in a user's Semantic Log.

- Network Gazetteer

  Client-side software is available for download from the Plazes site, which is executed everytime a user's laptop moves to a new network. Given the existence of any GPS data, the latitude and longitude can be associated with the network place, as well as a place name taken from the Getty gazetteer. This information is also taken from a user's Semantic Log, to provide extra locational context.

- Calendar data

  Calendar information in RDFiCal is utilised by photocopain, in order to annotate personal photos. These are being used to provide more context for a photograph, by recording where the user planned to be when the image was taken. Although iCal provides information about the timing of events, the descriptions of the events are in free text; we perform simple named entity extraction on the location string to identify place names with which the image is then annotated. The fact that people are not always where their calendar states they are is more evidence for why any proposed annotation require human approval.

- Image analysis, Classification, and Flickr

  A selection of image analysis techniques, such as the CIELab colour-map, *Hue, Intensity, Texture (HIT) Map* [24], and the edge direction coherence vector [36] have been used to propose annotations for image content. A number of classifiers have been trained using flickr's image pool as our source of training data, these are elaborated upon in [35]. Flickr users may associate images with a number of free text tags (e.g. Tim BL, WWW2006, Edinburgh); we use the photographs associated with certain tags as training sets for our image analysis algorithms. For example one hundred and fifty images of the tag 'flower' where taken from flickr via its API[21], and then images that are not flowers were also downloaded from flickr, in order to train a classifier.

  This process was automated by first identifying what words have been clustered together inside flickr (getRelated function via flickr api), and then listing the words have been clustered in conjunction with the list of flower's related tags. This list was then combined with the terms related to flower extracted from Wikipedia's Categories[22], and was used as a filter when randomly collecting a set of one hundred and fifty images we assume to be the class of 'not flower'.

  We selected a handful of flickr's most popular tags to be our initial content-based annotations (vocabulary). These include: landscape, cityscape, portrait, groupphoto, architecture, seascape, and flower. The decision to use this dataset has ensured that any proposed annotations are grounded within flickr's shared conceptualisation of these terms. For example, if Photocopain proposes the annotation 'landscape', what it actually means is 'this image is similar to images tagged landscape by the flickr community' as opposed to the developers' understanding of the word.

  The web service based architecture developed for photocopain, allows easy integration of new image analysis algorithms, and/or new clustering algorithms as needed.

  Work is currently underway to build automatic classifiers for commonly occurring human annotations. If a user persistently annotates his/her photos with a tag not recognised, the system will attempt to collect some test and training data from flickr, and automatically try different combinations of the different image analysis algorithms, along side the different classifiers. Another initiative on this front is the notion of having the 'gold standard' annotations to feedback into the system, retraining the classifiers in an incremental fashion.

[21] http://www.flickr.com/services/api/
[22] http://en.wikipedia.org/wiki/Wikipedia:Browse

- Community Annotations from the Semantic Logger

  Given that photocopain has been repurposed to work with the Semantic Logger, a piece of social software, the scope now allows for annotation's to be shared within communities. Friend information can be exploited in order for annotations to be shared with communities. Given that a friend of your took a picture at the same time and place will allow for annotations to be proposed by the system. This notion of using other peoples annotations was inspired by the undertakings of the ZoneTag project [23] at Yahoo!

Photocopain shows how the information found inside the Semantic Logger can be used to enrich one's personal media library. Photocopain uses *content*, *context*, and *community* based knowledge in order to generate as much metadata as possible. Photocopain presents the user with a number of annotations for each image submitted, while these are in turn corrected by the user, and then uploaded back to the users Semantic Log. This process of importing this photo-specific information back to the Semantic Logger, adds another dimension to be exploited by the aforementioned recommender system.

7. CONCLUSIONS AND FUTURE WORK

This paper presents the Semantic Logger a system for logging personal information. The design and resulting infrastructure have been grounded in two systems. The web service aspect of the architecture has been stressed through the paper, and is the basis for future work. It goes without saying that the Semantic Logger will require users to supply data for it to be a success, the system is about to be released to our department for some initial testing, before being released to the public.

OpenKnowledge (OK) [24] is an open peer to peer semantics based system, that accommodates knowledge sharing through interaction. Peers are able to carry out tasks and collaborate through executing interaction models associated with the task, provided they have downloaded appropriate software capable of satisfying any constraints posed in such models. These interaction models are expressed in the Lightweight Coordination Calculus [29] and peers assume the roles described in them and carry out message passing to accomplish a successful execution. We intend to define the interaction between the components of the Semantic Logger in such models, such that peers that join the OK system are able to automatically use the functionality made available by the Semantic Logger.

We indent to set up a Wiki, to allow people to present new sources of information, or new services as they see fit. We can envisage software that will log a user's video viewing habits, to aid the recommendation process. Integration with the Google Maps API [25] will allow for the information to be displayed and browsed by geographical data.

8. ACKNOWLEDGMENTS

This work is supported under the AKT IRC, sponsored by EPSRC, UK, Grant number GR/N15764/01, and under the OpenKnowledge and HealthAgents EU Framework 6 STREP projects under Grant numbers IST-FP6-027253 and IST-FP6-027213.

9. ADDITIONAL AUTHORS

Srinandan Dasmahapatra
SENSE Group, Southampton, sd@ecs.soton.ac.uk

Paul H. Lewis
IAM Group, Southampton, phl@ecs.soton.ac.uk,

David E Millard
IAM Group, Southampton, dem@ecs.soton.ac.uk,

Nigel R. Shadbolt
IAM Group, Southampton, nrs@ecs.soton.ac.uk.

10. REFERENCES

[1] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin. Incorporating contextual information in recommender systems using a multidimensional approach. *j-TOIS*, 23(1):103–145, Jan. 2005.

[2] M. Ahmed, H. H. Hanh, S. Karim, S. Khusro, M. Lanzeberger, K. Latif, M. Elke, and Kha. Semanticlife: A framework for managing information integration and web-based applications and services. In *Proceedings of the 6th International Workshop on Information Integration and Web-based Applications and Services (IIWAS 2004)*, 2004.

[3] H. Alani, S. Dasmahapatra, K. O'Hara, and N. R. Shadbolt. Identifying communities of practice through ontology network analysis. *IEEE Intelligent Systems*, 18(2):18–25, 2003.

[4] T. Berners-Lee. Keynote speech: Semantic web status and direction. In *International Semantic Web Conference 2003(IWSC-03)*. Springer, Berlin, Heidelberg, 2003.

[5] J. Broekstra, A. Kampman, and F. van Harmelen. Sesame: A generic architecture for storing and querying rdf and rdf schema, 2002.

[6] T. Burners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5), May 2001.

[7] V. Bush. As we may think. pages 17–34, 1988.

[8] S. Cayzer and P. Castagna. How to build a snippet manager. In *Proc. of Semantic Desktop Workshop at the ISWC*, 2005.

[9] L. Coyle and P. Cunningham. Improving recommendation ranking by learning personal feature weights. *Lecture Notes in Computer Science*, 3155:560–572, 2004.

[10] M. Davis, M. Smith, J. Canny, N. Good, S. King, and R. Janakiraman. Towards context-aware face recognition. In *MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia*, pages 483–486, New York, NY, USA, 2005. ACM Press.

[23] http://zonetag.research.yahoo.com/zonetag/
[24] http://openk.org
[25] http://www.google.com/apis/maps/

[11] F. Dawson and D. Stenerson. Internet calendaring and scheduling core object speci- cation. request for comments 2445. Technical report, Internet Engineering Task Force, 1998.

[12] EXIF. Exchangebale image file format for digital still cameras: EXIF version 2.2. Technical report, Japan Electronics and Information Technology Industries Association, 2002.

[13] K. Falkovych and F. Nack. Context aware guidance for multimedia authoring: harmonizing domain and discourse knowledge. *Multimedia Syst.*, 11(3):226–235, 2006.

[14] J. Gemmel, G. Bell, R. Lueder, S. Drucker, and C. Wong. Mylifebits: fulfilling the memex vision. In *MULTIMEDIA '02: Proceedings of the 10th ACM international conference in Multimedia*, pages 235–238, 2002.

[15] F. Giunchiglia, P. Shvaiko, and M. Yatskevich. S-match: An algorithm and an implementation of semantic matching. In *European Semantic Web Symposium*, pages 61–75, 2004.

[16] S. Harris. SPARQL query processing with conventional relational database systems. In *WISE Workshops*, pages 235–244, 2005.

[17] T. Heath, E. Motta, and M. Dzbor. Context as a foundation for a semantic desktop. In *Proc. of Semantic Desktop Workshop at the ISWC*, 2005.

[18] T. Iofciu, C. Kohlshutter, W. Nejdl, and R. Paiu. Keywords and rdf fragments: Integrating metadat and destktop search in beagle++. In *Proc. of Semantic Desktop Workshop at the ISWC*, 2005.

[19] S. Lawrence. Context in web search. *IEEE Data Engineering Bulletin*, 23(3):25–32, 2000.

[20] R. Lee. Scalability report on triple store applications. Technical report, MIT, 2004.

[21] A. Loizou and S. Dasmahapatra. Recommender systems and ontology alignment in the semantic web. In *ECAI2006 Workshop on Recommender Systems*, Riva del Garda, Italy, 2006.

[22] m. c. schraefel, M. Karam, and S. Zhao. mSpace: interaction design for user-determined, adaptable domain exploration in hypermedia. In P. D. Bra, editor, *Proceedings of AH 2003: Workshop on Adaptive Hypermedia and Adaptive Web Based Systems.*, pages 217–235, Nottingham, UK, 2003. Springer.

[23] F. Manola and E. Miller. RDF primer: W3c recommendation. http://www.w3.org/TR/rdf-primer/, 2004.

[24] G. G. Mateos and C. V. Chicote. A unied approach to face detection, segmentation and location using hit maps. In *In Syposium Nacional de Re-conocimiento de Formas y Anlisis de Imgenes*, Benicasim, Castelln, 2001.

[25] B. McBride. Jena: Implementing the RDF model and syntax specification. In *Proceedings of the 2nd International Workshop on the Semantic Web, SemWeb*, 2001.

[26] S. E. Middleton, D. C. De Roure, and N. R. Shadbolt. Capturing knowledge of user preferences: ontologies in recommender systems. In *1st international conference on Knowledge Capture*, pages 100–107, New York, NY, USA, 2001. ACM Press.

[27] K. Molle and S. Decker. Harvesting dekstop data for semantic blogging. In *Proc. of Semantic Desktop Workshop at the ISWC*, 2005.

[28] J. Richter, M. Volkel, and H. Haller. Deepamenta - a semantic desktop. In *Proc. of Semantic Desktop Workshop at the ISWC*, 2005.

[29] D. Robertson. A lightweight coordination calculus for agent systems. *Lecture Notes in Computer Science*, 3476:183–197, 2005.

[30] L. Sauermann, A. Bernandi, and A. Dengel. Overview and outlook on the semantic desktop. In *Proc. of Semantic Desktop Workshop at the ISWC*, 2005.

[31] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock. Methods and metrics for cold-start recommendations. In *25th International ACM Conference on Research and Development in Information Retreival*, 2002.

[32] N. R. Shadbolt, W. Hall, and T. Berners-Lee. The semantic web: Revisted. *IEE-Intelligent System*, 21(3):96–101, May 2006.

[33] K. Stoffel, M. G. Taylor, and J. A. Hendler. Efficient management of very large ontologies. In *AAAI/IAAI*, pages 442–447, 1997.

[34] M. Streatfield. Report on Summer Internship Work For the AKT Project: Benchmarking RDF Triplestores. Technical report, University of Southampton, 2005.

[35] M. M. Tuffield, S. Harris, D. P. Dupplaw, A. Chakravarthy, C. Brewster, N. Gibbins, K. OHara, F. Ciravegna, D. Sleeman, N. R. Shadbolt, and Y. Wilks. Image annotation with photocopain. In *Proceedings of Semantic Web Annotation of Multimedia (SWAMM-06) Workshop at the World Wide Web Conference 06*. WWW, May 2006.

[36] A. Vailaya, A. Jain, and H. Zhang. On image classication: City images vs. landscapes. In *Pattern Recognition.*, page 19211935, 1998.

[37] World Wide Web Consortium. SPARQL query language for RDF, working draft. Technical report, World Wide Web Consortium, 2005.

[38] H. Xiao and I. F. Cruz. A multi-ontology approach for perssonal information management. In *Proc. of Semantic Desktop Workshop at the ISWC*, 2005.