From Information to Interaction: in Pursuit of Task-centred Information Management

Alan Dix
Computing Department,
Lancaster University,
Lancaster, LA1 4YR,
United Kingdom
alan@hcibook.com

Akrivi Katifori
Dept. of Informatics and Telecommunications,
University of Athens
Panepistimiopolis, 157 84,
Athens, Greece
vivi@di.uoa.gr

Antonella Poggi
Dipartimento di Informatica e Sistemistica “A. Ruberti”,
Università di Roma “La Sapienza”,
Via Ariosto 25, 00185,
Rome, Italy
poggi@dis.uniroma1.it

Tiziana Catarci
Dipartimento di Informatica e Sistemistica “A. Ruberti”,
Università di Roma “La Sapienza”,
Via Ariosto 25, 00185, ,
Rome, Italy
catarci@dis.uniroma1.it

Yannis Ioannidis
Dept. Of Informatics and Telecommunications,
University of Athens
Panepistimioupolis, 157 84,
Athens, Greece
yannis@di.uoa.gr

Giorgios Lepouras
Dept. of Computer Science and Technology,
University of Peloponnese
Terma Karaiskaki, 22100,
Tripolis, Greece
gl@uop.gr

Miguel Mora
Escuela PolitÈcnica Superior
Universidad Autonoma de Madrid
28049 Madrid, Spain
Miguel.Mora@uam.es

Abstract

In this paper we describe two automatic inference mechanisms constructed around a personal ontology with the aim of providing human like assistance for form-filling. This is part of the wider vision of DELOS Task 4.8 Task-centered Information Management, which aims to create a more activity or task-focused interaction environment for users, taking their individual library of digital information and turning it into a resource for automated support. The two techniques described in this paper are (i) a spreading activation algorithm operating over the user's personal ontology which enables a form of temporal context or memory and (ii) a rule inference algorithm that takes examples of completed forms and infers how the fields map to generic sub-gaps of the ontology.

Categories and Subject Descriptors

H.2 [Database Managment]: H.2.1 Logical design; H.3 [Information Storage and Retrieval]: H.3.7 Digital Libraries; H.5 [INFORMATION INTERFACES AND PRESENTATION]: H.5.2 User Interfaces

General Terms

Design, Algorithms

Keywords

Personal Information Management, Ontology Management, Task Management

1 Introduction

Users have a considerable body of information at their hands, both shared resources such as digital liberties, but also their own personal digital resources such as calendars, bookmarks and address books. The process and tools for storing, organising and accessing these disparate individual resources is known as personal information management (PIM). Sometimes information is wanted purely for itself, for example to read a book. However, more often digital resources, both shared and personal, are being accessed for a purpose. The real focus of the users is on some activity, goal or task not on the information itself.

The vision of DELOS Task 4.8 Task-centered Information Management is to create software that directly supports the users' tasks not simply their information. We wish to transform the user's passive personal digital library into an active digital assistant.

In previous papers [6][8][15][23][27] we have outlined this vision and architectural structure of our prototype Task-centered Information Management system. In this paper we will focus on a particular aspect of this, the automatic completion of forms. This is partly because it is an important activity in its own right; as more and more interactions are through primarily web-based forms. However it is also because the techniques and algorithms used for form-filling are the ones which will be used more broadly in the longer-term TIM vision. In particular these are aspects that include significant levels of 'intelligence' and inferencing.

While our focus and vision is on supporting tasks not the information per se, this does not mean the information is not important and in fact at the core of our TIM architecture is the user's personal ontology that gathers in one (possibly virtual) place information about everything from the user's date of birth to the telephone numbers of colleagues and meta-information of recently read papers.

The paper is organized as follows. In Section 2 we discuss related work followed by a discussion of our own previous work in Section 3. In Section 4, a scenario is introduced illustrating the main issues that we focus on in this paper, i.e. context and task inference. Section 5 introduces our approach to context inference using spreading activation over the personal ontology. In Section 6, we tackle task inference by presenting a rule-based inferencing mechanism.

2 Related work

The issues faced in DELOS Task 4.8 belong to several different areas of research. In what follows we discuss related work in each of these main areas.

User modelling and ontologies. Using an ontology to model semantics related to the user personal domain has already been proposed for various applications like web search [17][32]. Most of these approaches use ontologies only as concept hierarchies, like hierarchies of user interests, without particular semantic complexity. The value of ontologies for personal information management has also been recognized and there is on-going research on incorporating them in PIM systems like OntoPIM [23], GNOWSIS [34] and the semantic desktop search environment proposed in [9].

Task management. Recently, research efforts on the problem of managing the user’s tasks have lead to the prototype Activity-Centered Task Assistant (ACTA), implemented as a Microsoft Outlook add-in[4]. In ACTA, a user’s task, named “ACTA activity”, is represented as a pre-structured container, which can be created inside the email folder hierarchy. It is a task-specific collection containing structured predefined elements called “components”, that embody common resources of the task and appear as activity sub-folders. Thus, for example, by creating an ACTA activity to represent a meeting, and by inserting the component “contacts”, the user aims at relating the content of this sub-folder, which will essentially be a list of names, with that particular meeting. Moreover, the population of an activity is done semi-automatically, by allowing the user just to drag into the appropriate activity component, the document containing the relevant data, which is afterward automatically extracted and stored. Even though ACTA activities are built relying on user’s personal data, their approach is not comparable to ours, since they do not consider tasks as a workflow of actions (e.g. filling and sending the meeting invitation email), which can be inferred and semi-automatically executed.

Task inference. There has been a long history of research into task detection, inference and prediction in human-computer interaction, with a substantial activity in the early 1990s including Alan Cypher's work on Eager [11] and several collections [3]. The line of work has continued (e.g., [12],[28]), but with less intensity than the early 1990s. Forms of task inference can be found in widely used systems, for example the detection of lists etc. in Microsoft Office or web browsers that auto-fill forms. The first example clearly demonstrates how important it is that the interaction is embedded within an appropriate framework, and how annoying it can be when inference does not do what you want! Some of this work lies under the area of “programming by demonstration” or “programming by example”, where the user is often expected to be aware of the inferences being made and actively modify their actions to aid the system. This is the case of [19] where authors present a learning system, called PLIANT, that helps users anticipating their goal, by learning their preferences and adaptively assisting them in a particular long-running application such as a calendar assistant. Other work falls more under user modeling, intelligent help, automatic adaptation or context-aware interfaces where the user may not be explicitly aware that the system is doing any form of inference [5]. Our work lies with the former as we do expect that users will be aware of the inferences being made and work symbiotically with the system in order to create a fluid working environment.

3 Summary of previous contributions

In this section we briefly present the DELOS Task 4.8 previous contributions, namely OntoPIM and the task specification language.

OntoPIM. OntoPIM [23] is a module that allows to manage the whole collection of heterogeneous personal data usually maintained in a personal computer (e.g. contacts, documents, emails), and to access them through a unified, integrated, virtual and yet user-tailored view of her data. This view is called Personal Ontology (PO), since it reflects the user’s view of her own domain of interest. It is therefore specific to each user. As for the language to specify the PO, we use the Description Logic called DL-Lite$_A$ [30][31], since besides allowing to express the most commonly used modeling constructs, it allows to answer expressive queries, i.e. conjunctive queries, in polynomial time with respect to the size of the data. This is clearly a distinguishing and desirable feature of such a language, in a context like ours, since the amount of data is typically huge in one’s personal computer.

In order to achieve the above mentioned result, OntoPIM proceeds as follows. First it extracts, by means of appropriate wrappers, pieces of relevant data from the actual personal data contained in the personal computer. Then it exploits the so-called Semantic Save module, which (i) stores such data in a DBMS, maintaining also its provenance, and (ii) stores the relationship existing between the data and the PO, as (implicitly) specified by the user. Note that the latter relationship reflects indeed the data semantics according to the user.

Task Specification. In order to be able to semi-automatically execute user tasks, we defined a task specification language [8] having two main features. First, the language is at the same time expressive enough for actually being helpful to the user, and simple enough for being effectively “usable” and “learnable” by the system. Second, the language allows to specify as a part of the task definition, the input/output data mappings, i.e. the relationships existing between the task and the PO. Specifically, the input data mappings specify the query to be posed over the PO in order to obtain the task input, whereas the output data mapping specify the task output as an update (possibly empty) to be computed over the personal data, according to the semantics of both the task execution and the PO. As we will see, the specification of task input/output data mappings is crucial for task inference/detection/suggestion.

Furthermore, we have explored task inference top-down approaches, where the user specifies aspects of the task using forms of declarative scripting. Our proposal was based on the idea of combining task decomposition and a plan language to describe for each complex task, the execution plan of its subtasks. On one hand, a complex task is decomposed into a set of subtasks. This allowed for a comprehensible view of the task. On the other hand, we have proposed a plan language limited to sequence, alternatives and repetition.

Personal Ontology and User Profile. As previously noted, the personal ontology is at the heart of the TIM vision. This personal ontology may be gathered from existing resources such as an email address book and/or produced explicitly by the user. It is the place that gathers both personal profile information such as the user's address and also wider information relating to the user such as friends, or projects. The personal ontology we envisage is not simply populated with instances on an individual basis, but also includes user specific classes and relations.

In order to create a simple yet comprehensive set of upper level concepts for the personal ontology, the standard classes, profile information models maintained by various applications, like instant messengers (e.g., ICQ[1] and community websites (e.g. Facebook[2], Myspace[3]), and proposed by researchers, like [17][37][36], were examined and general ontologies like the ones presented in [29] were taken into account along with the MIME directory profile vCard[4].

Details on the creation of the personal ontology may be found in [21]. The version of the personal ontology used in this work is an extension of the one in [21], as it has been enriched with more user-related classes for the user stereotype of "Researcher" in order to be used for the fine tuning and evaluation of the spreading activation algorithm. The ontology, along with example instances may be found in [25]. Figure 1 presents an overview of the upper levels of the class hierarchy.

Figure 1: Overview of the Personal Ontology with the upper levels expanded

---

[1] http://www.icq.com

[2] http://www.facebook.com/

[3] http://www.myspace.com/

[4] http://www.w3.org/TR/vcard-rdf

Ontology management tools such as Protégé[5] are currently designed to be used by ontology experts. In order to allow users to update their personal ontology easily, a web-based ontology browsing and editing tool has been proposed. This tool hides some of the complexity of the class hierarchy and allows custom views to be defined.

The next section illustrates our vision on automatic form-filling through a real-world scenario.

4 The Problem: an illustrative scenario

Consider the partially completed (paper) form:

		Form KZ1
Family Name:	Dix	
First Name:		
Email:		
City/Town:		

Given the form is in this paper and one of the authors is Alan Dix it is fairly obvious that the first field should be "Alan" and the email is "alan@hcibook.com". Obviously most forms do not appear in the middle of academic papers, but they do have a context.

Imagine that Antonella has a (human) assistant called Henry and that it is Antonella who has partially completed the form. Henry would know that Alan Dix is a colleague of Antonella and complete the information accordingly. This was easy given Dix is an uncommon name and there is probably only one "Dix" in Antonella's address book.

If it only the "First Name" field had been filled in then there might have been several possibilities as "Alan" is a more common name than "Dix". In such cases Henry might have had to ask her which Alan this meant from a small number of possibilities. However, if Antonella's recent activity had been about the TIM project then Henry might have suggested Alan Dix as top of the list as he is connected with the TIM project.

The field that says "City/Town" is a little more problematic as it could be Alan's place of work that is being referred to (Lancaster) or his place of residence (Kendal) or maybe even his place of birth (Cardiff). Here again Henry could narrow this down to a few possibilities, but in the end would have to ask. Let us suppose that Antonella chose "Lancaster" as in this context she knows it means place of work.

Now imagine Antonella starts to fill out a second (very simple) form that says:

	Form XQ3
University:	

If this form appeared completely on its own then Henry would suggest " Università di Roma" as this is Antonella's own institution. However, if this immediately follows the previous form there may be some connection, so Henry would also suggest "Lancaster University" as the last form had been about Alan Dix. Again Henry would be able to offer likely choices, but Antonella would have to choose which. Let us suppose she selects "Lancaster University".

[5] http://protege.stanford.edu/

So the completed forms look like:

Form KZ1

Family Name: Dix
First Name: Alan
Email: alan@hcibook.com
City/Town: Lancaster

Form XQ3

University: Lancaster University

At this point Henry falls ill and her new assistant Enrico understands only Italian. Happily the address book software and other information about Antonella's contacts and projects is all internationalised so Enrico has access to all Antonella's personal information. Unfortunately the forms are only in English so that Enrico can look at previous forms that Antonella has completed (with Henry's aid), but has no understanding of terms like "Family Name".

Antonella then starts to fill out a new form KZ1 and enters the name "Katifori ". Although Enrico does not understand any English, he is able to look at the previous completed form KZ1 for Alan Dix and form this work out what the different fields mean. He can then look up information about Akrivi Katifori and propose a completed version of the form (Enrico suggestions in italics):

Form KZ1

Family Name: Katifori
First Name: *Akrivi*
Email: *vivi@mm.di.uoa.gr*
City/Town: *Athens*

Antonella then gets a blank form XQ3. Enrico can see that the last time XQ3 was completed immediately after a KZ1 it referred to the same person and was their university name, so he completes the form.

Form XQ3

University: *University of Athens*

Finally imagine Antonella does not keep email addresses in her address book. So before filling in the original form KZ1 she had to go to an online system. She fills in the name "Alan Dix" and gets "alan@hcibook.com". As Henry was efficient, he was careful to keep the record of this along with the paper forms that Antonella uses. So, when Enrico takes over the job he is able to see that the output of the online system was used as the input of the next form and aid Antonella just like the input–input relationships in the paper forms.

From this scenario we can see several things:

(a) *use of personal information* – The human assistants know about Antonella herself (e.g. her university is "Università di Roma ") and Antonella's contacts, projects etc.

(b) *use of temporal context* – The human assistants know about things that Antonella has been doing recently and are therefore likely to be related to current activity. For example, if there are several people called "Alan" in her address book, then the one related to the project she is currently working on is likely the right one.

(c) inference of form types – Even though Enrico does not understand English he can work out what the fields mean by looking at previous examples. For example, he can work out what "Family name" and "First name" mean because he can see that "Dix" and "Alan" have been used to complete them.

(d) inference of form relationships – The human assistants knows how fields in one form relate to those in subsequent forms based on previous examples. For example, the field "University" in form XQ3 means the university of the person referred to in the preceding form KZ1.

(e) uncertainty and human choice – While the human assistants could make educated guesses of many of the fields, there is often some uncertainty and they had to either offer choices to Antonella, or if the appropriate value is sufficiently clear present it as a suggestion.

We wish to be able to provide automatic support that shares some of the features of human support illustrated in the above scenario. In particular the support that Enrico was able to offer even though he did not understand the language written on the forms. Of course, some understanding of the text on the forms would be useful as the label "email" tells you what the next field will be. However, this is already used successfully in many web browsers and studied by others (e.g., [10] [33]), so we are focusing our efforts on novel technologies to address different aspects of human assistance.

The *personal ontology* is central to this as it provides the personal information (a) that is used throughout. The temporal context (b) is provided by the use of *spreading activation* over the personal ontology (Section 5). This means that things that have recently been part of the user's activities or are related to them can be flagged as 'hot'. To address points (c) and (d) we have a *rule inference engine* (Section 6) that takes examples of forms and infers relationships between their fields, again making use of the personal ontology, and as part of this also create suggested types for them.

Just like a human assistant, these algorithms do not always give definitive results and they are used mostly to tune suggestions and choices. This use of potentially fallible heuristics within a supportive user interaction has been called *appropriate intelligence* [14], and central to this is the need to make sure that failures in inference are virtually costless.

5 Spreading Activation over the Personal Ontology

In order to make our computer as helpful as the human assistant it would be useful to attempt to develop for the computer the appropriate levels of “intelligence” needed in order to assist us as efficiently as the human one. To this end, we took advantage of the existing studies on the mechanisms of the human memory and, based on them, we propose an appropriate “memory” mechanism through a personal ontology. The following section briefly presents existing theories on human memory whereas the next two their application for our “assistant”.

Different Timescales of the Human Memory

The human memory operates on multiple timescales. According to the model that Atkinson and Shiffrin proposed in 1968 [2], there are two distinct memory stores: short term memory, which contains the things we are currently thinking about, and long term memory, which contains the things we have learnt and stay with us for years (possibly forever), but may be more or less easy to retrieve. However, there are things that stay around longer than the 10-30 seconds of the short term memory, but are related to the current moment and task. This in-between or 'mezzanine' memory may be in part due to more maintained electrical states or chemical changes in neurons called long term potentiation or LTP, which are known to last for anything from seconds to hours.

According to the spreading activation theory [1], knowledge in the long term memory is represented in terms of nodes and associative pathways between nodes, which form a semantic network of concepts. A hierarchical structure is also present in this network, classifying concepts in more generic and more specific ones [35]. Connection strength and node distance are determined by the semantic relations or associative relations between the conceptual nodes. This model assumes that activation spreads from one conceptual node to those around it, with greater emphasis to the closer ones [18].

An ontology is a knowledge representation means very close to the structure of the semantic network of the human long term memory. To this end, it was selected as the storage mechanism for the "memories" of our assistant (cf. Section 3). Having the personal ontology to store memories, the next step is to create the appropriate mechanisms to access and use these memories when performing a task, like filling a form. Again, in this case, the human memory spreading activation theory may serve as the basis for these mechanisms. The following section presents a spreading activation framework for accessing the information in the personal ontology and inferring thus the user context.

Different Timescales for Task-based Interaction

In user interaction with a system multiple timescales can be noted, which roughly correspond to the ones apparent in the human memory model. First, there are the contents of the personal ontology and the available information sources that roughly correspond to human long-term memory. Not all things in this long-term system memory are equally important and it should be recorded that some things (such as the user's own address) are more important than others (the address of the plumber). Corresponding to the short/working memory are the things the system has to store regarding the current user task – for example, the contents of the email the user has just opened, the text the user has just selected, the web page just visited, or the form field being completed. Finally, there are the things the user has been recently doing (other pages visited, documents seen, etc.) that roughly correspond to the mezzanine memory. This recent history is important as, for example, if the user has recently viewed a web site about an upcoming event and then goes to a travel website it is likely that the place to be visited is that of the event.

These different levels could be dealt with in a spreading activation framework by simply fading memories over time so that entities frequently encountered become increasingly highly "activated". However, with a single mechanism it is hard to create a balance between having recent things be more active (the place just mentioned in an email) than important general things (the user's address), whilst on the other hand not having them crowd out the longer-term things.

Because of this it seems more appropriate to explicitly code these different levels using multiple activations with 'rules' for passing activation between short-term to longer-term memories. The simplest such rule would be to define thresholds so that if the short-term activation (STA) exceeds some value then the medium-term activation (MTA) is incremented and similarly if the medium term memory exceeds its own threshold (signalling that something has been repeatedly of high relevance), then the long-term activation (LTA) grows. In addition, certain events (e.g. explicitly interacting with an entity) may be regarded as sufficiently important to increase the long-term memory directly (just as significant events are easily remembered).

The ActiveOnto Protégé Plug-in

In order to test the spreading activation algorithm, which has been implemented in Java, a Protégé plug-in has been created[6]. The ActiveOnto plug-in allows the initialization and setting of all the algorithm parameters and allows the user to simulate the functionality of the algorithm in a PIM/TIM system.

In the plug-in the user may select instances as “Immediately Active”, simulating thus their appearance in an e-mail, document or web page. Then, by pressing the “update” button, the STA, MTA and LTA activations are computed and the user may view the instances that received an STA value greater than a specific user-defined threshold (cf. Figure 2).

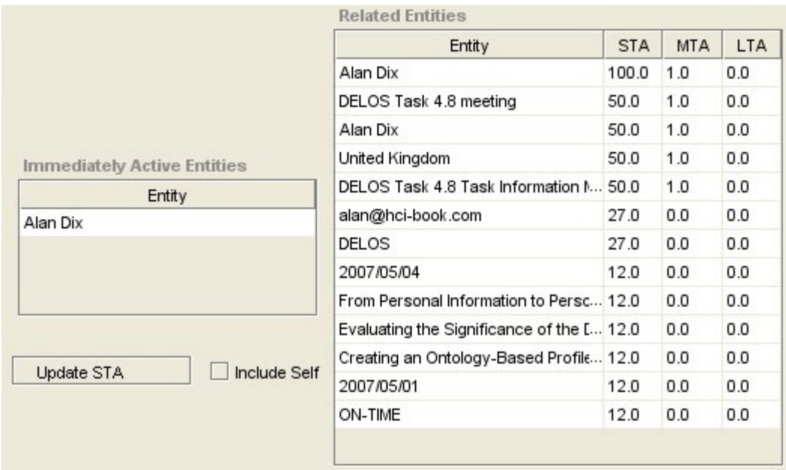
In order for the plug-in to function, an ontology with specific characteristics must be used, as slots representing the activation weights are needed. More specifically, the ontology to be used with the plug-in should have the following characteristics:

1. All classes should conform to a meta-class having the slots IA, IN, STA, MTA, LTA and MAXLTA of type String.

2. All instances should have the slots IA, IN, STA, MTA, LTA and MAXLTA of type String.

3. All slots should conform to a meta-slot with an LTW slot of type String.

An example of personal ontology conforming to the above characteristics can be found in [25].



Entity	STA	MTA	LTA
Alan Dix	100.0	1.0	0.0
DELOS Task 4.8 meeting	50.0	1.0	0.0
Alan Dix	50.0	1.0	0.0
United Kingdom	50.0	1.0	0.0
DELOS Task 4.8 Task Information I...	50.0	1.0	0.0
alan@hci-book.com	27.0	0.0	0.0
DELOS	27.0	0.0	0.0
2007/05/04	12.0	0.0	0.0
From Personal Information to Perso...	12.0	0.0	0.0
Evaluating the Significance of the [...	12.0	0.0	0.0
Creating an Ontology-Based Profile...	12.0	0.0	0.0
2007/05/01	12.0	0.0	0.0
ON-TIME	12.0	0.0	0.0

Figure 2. Part of the plug-in window showing the STA, MTA and LTA values for the entities that received STA activation value greater than 12, when entity “Alan Dix” was activated.

The ActiveOnto plug-in is currently being used to test the spreading activation algorithm. After a thorough evaluation, the next step will be the integration with the form-filling module.

6 Task inference

If we assume a well-populated personal ontology (possibly also drawing on standard sources such as gazetteers) it is possible to encode the relationships between form fields such as those found in Section 4. Figure 3 shows the two forms used in the Section 4 examples annotated with ontology relationships. In fact there are three levels of such annotation:

(i) For a particular *instance* of a filled out form, we can find a particular person, Alan Dix, a particular university and particular location the various fields are the properties of the respective

[6] Available from http://oceanis.mm.di.uoa.gr/pened/?c=pub - plugins

instances.

(ii) In general, the field labelled "Family Name" is in fact a surname of a Person (where surname is a property of the class Person)

(iii) In general the fields "Family Name", "First Name" and "Email" refer to the respective fields of the *same* person. This needs not be the case. For example, Figure 4 shows an alternative potential set of relationships between fields showing that the same typing of fields (ii) can correspond to different patterns of instantiation.

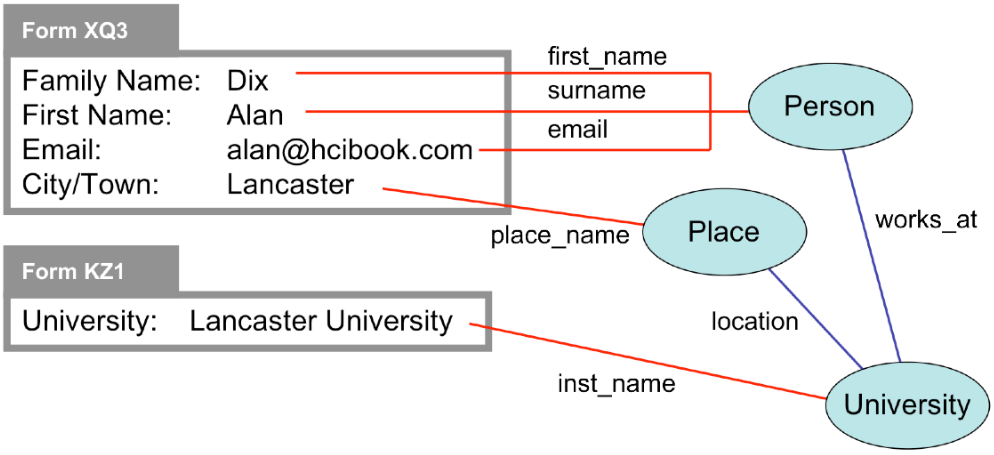


Figure 3. Forms annotated with ontology relationships.

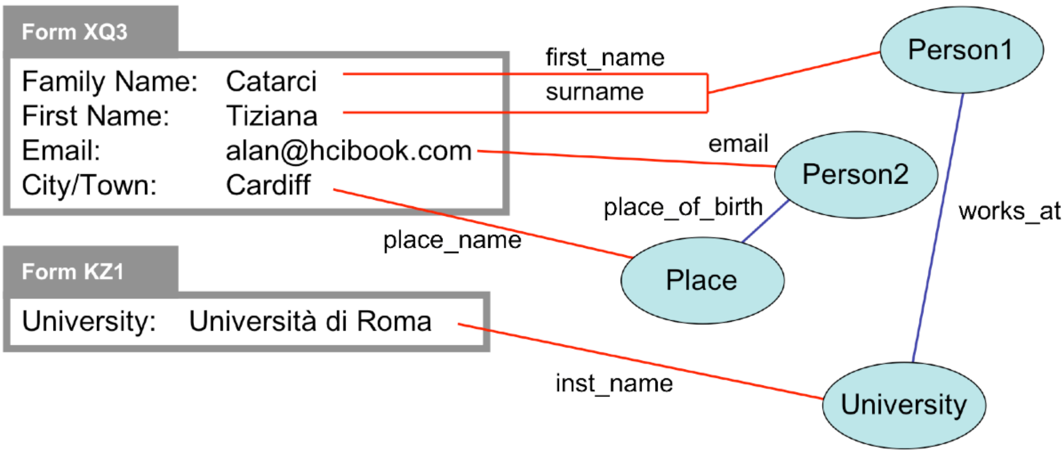


Figure 4. Alternative possible relationship of fields.

A human analyst looking at the examples of the form would in many cases be able to decide what entities were referred to by the various fields (level i), what the types are (level ii) and from these the general relationship (level iii). Some of this would use the analyst's world knowledge beyond what is formally stored in an ontology. However, we can infer much of the same information automatically.

Our rule inference engine does precisely this job. It does not create the full graph but instead works out individual rules for each field showing how it could be inferred from other fields (cf. Figure 6). Each rule has a destination field and a path from it to source fields where the path starts with a property link, then follows a series of class-class relationships and finally has one or two property links to the source fields. For example, the City.Town field in Figure 3 might have path:

place_name – Place – location – University – works_at – Person – <<binding>>

where binding is{ first_name: "Family Name", surname: "Family Name", email: "Email" }

In general such rules may not be unique. For example, the form data for “Akrivi Katifori” would be ambiguous of she both lived and worked in Athens. While it may not be possible to obtain a single rule, some rules are 'better' than others, so the rules each give a weight to show how likely they are to be correct. For example, there would be a relationship between the person Alan Dix and the place Lancaster through the path:

Person (Alan Dix) – lives_in – Place (Kendal) – in_country – Counry (UK)
– inv(in_country) – Place (Lancaster)

In a sufficiently well connected ontology there will be an indeterminate number of such nonsense paths and these need to be either avoided or given a low weighting.

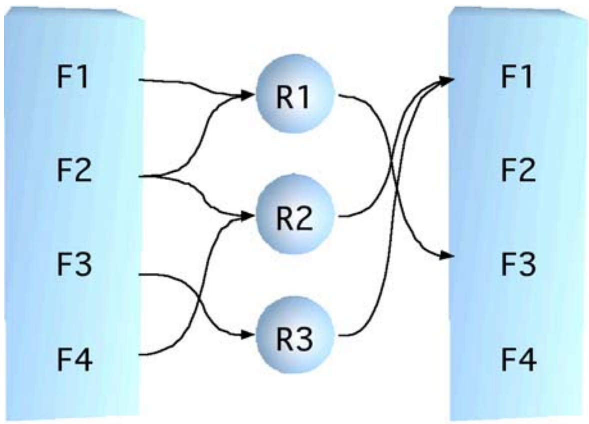


Figure 5. Each field has rules showing how it can be derived from others.

The rule inference algorithm first of all looks up all filled in form fields against property values in the ontology. This yields candidate start points. It then effectively works out 'shortest' paths from a destination field to a single source field or group of fields using a variant of Dykstra's algorithm [13]. Here shortest takes into account the number of relationships traversed and also the arity of the relationships: if a path from source to destination traverses a 1-m relationship it is less good than one that traverses a m–1 relationship as the latter is more predictive. The best paths are retained and given a weight with better paths having a higher weight.

This process is followed for a single instance of a form giving suggested form filling after the very first exposure. However, over time, when new examples are seen, the set of potential rules should reduce giving more and more precise predictions.

When the user starts to fill out a new form each field is matched against the stored rules to see if it can be given a value. If several rules could be used then the rules are ordered based on a dynamic weight. The dynamic weight combines the weight of the rule and the precision of the various source fields (e.g. "Dix" as a surname is more precise than "Smith").

The current implementation uses a Thunderbird plug-in to monitor live web activity (see Figure 6). The plug-in is responsible for scanning the web page DOM to find forms and fields. As the user starts to complete the form, the plug-in interrogates the rules selection and execution components in order to attempt to fill in remaining fields. When the form has been completed the data is passed back to the rule inference algorithm so that it can be minded for further rules or to refine previous rules for the same form. The rule inference, selection and execution components are all implemented in Java connecting to Protégé or an alternative ontology store. The system has sub-second responses for both rule inference from a new form and form filling.

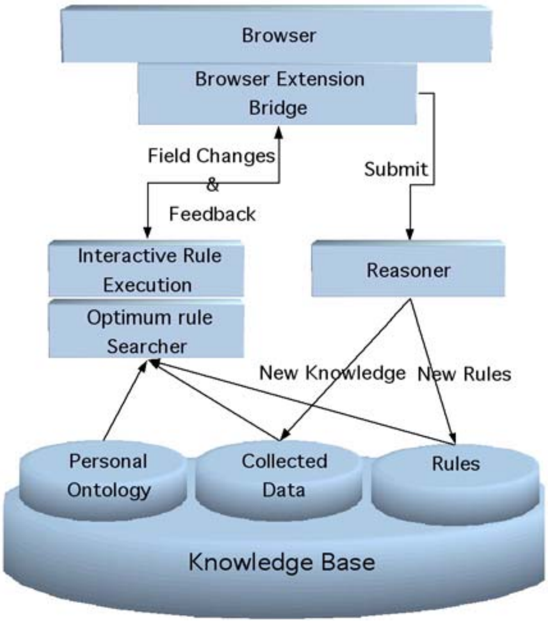


Figure 6. Task inference architecture

In a Semantic Web world with full meta-data for all information and services, some of the work of the rule inference algorithm would be un-necessary. However, such meta-data would only encode generic classes and relationships, but we would also want to deal with idiosyncratic relationships and classes, for example, the fact that a user only uses a particular form for looking friends' phone numbers and another for work colleagues while both forms would be marked up generically to refer to "Person". More pragmatically the techniques described here work for any web form whether or not it has any form of semantic mark-up, effectively inferring what the mark-up would be … and currently the vast majority of web-based material has no such mark-up.

7 Summary and on-going work

We have described two 'intelligent' mechanisms that together provide the necessary inferencing to enable form-filling support similar to a human assistant. The rule inference engine has been deployed on one of the authors' machines for a period of months gathering data so far on over 350 forms. This is beginning to give us an idea of how such a system would work in practice. This and the spreading activation algorithm are integrated to the extent that they both share the common personal ontology, but are not yet more deeply connected. We expect this to allow more efficient and more subtle suggestions.

These two mechanisms are components of a larger prototype TIM system many parts of which are now in place and some still in process. In particular, we have ongoing work on inference of task sequences and closer integration with desktop applications. We expect that the whole will definitely be greater than the sum of its parts. However, it is pleasing that even the components are already showing considerable promise.

8 References

[1]. Anderson, J. R. 1983. A spreading activation theory of memory. Journal of Verbal Learning and Verbal

Behaviour, 22, pp. 261-295.

[2]. Atkinson R. C. and Shiffrin R. M., 1968. Human memory: A proposed system and its control processes. In K.W. Spence and J.T. Spence (Eds.), The psychology of learning and motivation, vol. 8. London: Academic Press.

[3]. Beale R. and Finlay J., 1993. Neural networks and pattern recognition in human-computer interaction. SIGCHI Bull. 25 (2), pp. 25-35.

[4]. Bellotti V. and Thornton J., 2006. Managing Activities with TV-Acta: TaskVista and Activity-Centered Task Assistant. In *Proceedings of the Second SIGIR Workshop on Personal Information Management (PIM 2006)*.

[5]. Boone G., 1998. Concept features in Re: Agent, an intelligent Email agent. In K. P. Sycara and M. Wooldridge (eds), *Proceedings of the Second international Conference on Autonomous Agents. Minneapolis,* ACM Press, pp. 141-148.

[6]. Catarci T., Dix A., Katifori A., Lepouras G., and Poggi A, 2007. Task Centered Information Management. In *Proceedings of the DELOS Conference*, Springer-Verlag LNCS. To appear.

[7]. Catarci T., Dong X. L., Halevy A. and Poggi A., 2007. Structure Everything. In Jones W. and Teevan J. (eds), Personal Information Management. University of Washington Press UW Press. To Appear.

[8]. Catarci T., Habegger B., Poggi A., Dix A., Ioannidis Y., Katifori A. and Lepouras G., 2006. Intelligent User Task Oriented Systems. In *Proceedings of the Second SIGIR Workshop on Personal Information Management (PIM 2006)*.

[9]. Chirita P.-A., Gavriloai R., Ghita S., Nejdl W., and Paiu R, 2005. Activity Based Metadata for Semantic Desktop Search, Proceedings of the 2nd European Semantic Web Conference.

[10]. Chusho T., Fujiwara K., and Minamitaniet K., 2002. Automatic Filling in a Form by an Agent for Web Applications. In *Proceedings of the Asia-Pacific Software Engineering Conference 2002*, pp. 239-247.

[11]. Cypher A., 1991. Eager: Programming Repetitive Tasks by Example. In *Proceedings of the ACM Conference on Human Factors and Computing Systems (CHI'91)*.

[12]. Cypher A. (ed), 1993. Watch What I Do: Programming by Demonstration. MIT Press.

[13]. Dijkstra E. W., 1959. A note on two problems in connexion with graphs. In Numerische Mathematik, 1, pp. 269–271.

[14]. Dix A., Beale R., and Wood A., 2000. Architectures to make Simple Visualisations using Simple Systems. In *Proceedings of Advanced Visual Interfaces (AVI2000)*, ACM Press, pp. 51-60.

[15]. Dix A., Catarci T., Habegger B., Ioannidis Y., Kamaruddin A., Katifori A., Lepouras G., Poggi A., Ramduny-Ellis A., 2006. Intelligent context-sensitive interactions on desktop and the web. In *Proceedings of the International AVI'2006 Workshop on Context in Advanced Interface*.

[16]. Dix A. and Marshall J., 2003. At the right time: when to sort web history and bookmarks. In Volume 1 of the *Proceedings of HCI International 2003*, J. Jacko and C. Stephandis (ed.), Lawrence Erlbaum

Associates, pp. 758-762.

[17]. Gauch S., Chaffee J., Pretschner A., 2003. Ontology-Based User Profiles for Search and Browsing, User Modeling and User-Adapted Interaction. Journal of Personalization Research, Special Issue on User Modeling for Web and Hypermedia Information Retrieval.

[18]. Gazzaniga M. S., Ivry R. B., and Mangun G. R., 1998. Cognitive neuroscience: The biology of the mind. W. W. Norton & Company, New Edition, pp. 289-294.

[19]. Gervasio M. T., Moffitt M. D., Pollack M. E., Taylor J. M. and Uribe T. E., 2005. Active preference learning for personalized calendar scheduling assistance. In *Proceedings of the 10th international conference on Intelligent user interfaces (IUI '05)*, ACM Press, pp. 90-97.

[20]. Golemati M., Katifori A., Vassilakis C., Lepouras G., Halatsis C., 2007. Creating an Ontology for the User Profile: Method and Applications. In *Proceedings of the First International Conference on Research Challenges in Information Science (RCIS)*. To appear.

[21]. Golemati M., Katifori A., Vassilakis C., Lepouras G., Halatsis C., 2006. User Profile Ontology version 1, available at http://oceanis.mm.di.uoa.gr/pened/?category=publications

[22]. Katifori A., Daradimos I., Dix A., 2007. Web Methods for Ontology Retrieval, Technical Report, http://oceanis.mm.di.uoa.gr/pened/?c=pub#tr

[23]. Katifori A., Poggi A., Scannapieco M., Catarci T., and Ioannidis Y., 2005. OntoPIM: how to rely on a personal ontology for Personal Information Management. In *Proceedings of the 1st Workshop on The Semantic Desktop*.

[24]. Katifori A., Torou E., Halatsis C., Lepouras G., and Vassilakis C., 2006. A Comparative Study of Four Ontology Visualization Techniques in Protégé: Experiment Setup and Preliminary Results. In *Proceedings of the IV'06 Conference.*

[25]. Katifori A., Vassilakis C., Dix A., Daradimos I., and Lepouras G., 2007. Spreading activation user profile ontology. Available from http://oceanis.mm.di.uoa.gr/pened/?category=pub#ontos

[26]. Koutrika G. and Ioannidis Y., 2005. Personalized Queries under a Generalized Preference Model. In *Proceedings of the 21st International. Conference On Data Engineering (ICDE)*, pp. 841-852.

[27]. Lepouras G., Dix A., Katifori A., Catarci T., Habegger B., Poggi A., and Ioannidis Y., 2006. OntoPIM: From Personal Information Management to Task Information Management. In *Proceedings of the Second SIGIR Workshop on Personal Information Management (PIM 2006).*

[28]. Lieberman H. (ed), 2001. Your Wish is My Command: Programming By Example. Morgan Kaufmann.

[29]. Miller G. A., 1990. WORDNET: An On-Line Lexical Database. International Journal of Lexicography, 3–4, pp. 235–312.

[30]. Poggi A., 2006. Structured and Semi-Structured Data Integration. PhD thesis, Dipartimento di Informatica e Sistemistica, Universit`a di Roma "La Sapienza".

[31]. Poggi A., Lembo D., Calvanese D., De Giacomo G., Lenzerini M., and Rosati R., 2007. Linking Ontologies to Data. International Journal of Data Semantics. To appear.

[32]. Razmerita L., Angehrn A., and Maedche A., 2003. Ontology based user modeling for Knowledge Management Systems. In *Proceedings of the User Modeling Conference*, Springer Verlag, pp. 213-217.

[33]. Rukzio E., Schmidt A., and Humann H., 2004. Privacy-enhanced Intelligent Automatic Form Filling for Context-aware Services on Mobile Devices. In *Proceedings of AIMS*.

[34]. Sauermann L., 2005. The Gnowsis Semantic Desktop for Information Integration. In *Proceedings of the 3rd Conference Professional Knowledge Management*.

[35]. Sharifian F. and Samani R., 1997. Hierarchical Spreading of activation. In *Proceedings of the Conference on Language, Cognition and Interpretation*, pp. 1-10.

[36]. Tazari M. R., Grimm M., and Finke M., 2003. Modeling User Context. In *Proceedings of the 10th International Conference on Human-Computer Interaction (HCII2003)*.

[37]. Trajkova J. and Gauch S., 2004. Improving Ontology-based User Profiles. In *Proceedings of RIAO 2004*, pp. 380-389.