Using Spreading Activation through Ontologies to Support Personal Information Management

Akrivi Katifori
Dept. of Informatics & Telecommunications, University of Athens
Panepistimioupolis , Ilissia, 15784, Athens, Greece
vivi@di.uoa.gr

Costas Vassilakis
Dept. of Computer Science and Technology, University of Peloponnese
Terma Karaiskaki, 22100, Tripoli, Greece
costas@uop.gr

Alan Dix
Computing Department, Lancaster University,
LA1 4WA UK,
a.dix@comp.lancs.ac.uk

ABSTRACT

Recent research in the domain of Personal Information Management has recognized the need for a paradigm shift towards a more activity-oriented system. Ontologies, as semantic networks with a structure not dissimilar to the one used by the human brain for storing long-term knowledge, may be very useful as the basis of such a system. This work proposes the use of spreading activation over ontologies in order to provide to a task-based system and its associated tools with methods to record semantics related to documents and tasks and to support user context inference.

Author Keywords

Personal ontology, spreading activation, context inference.

ACM Classification Keywords

H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous, H3.0 Information Storage and Retrieval: General.

INTRODUCTION

As a direct result of the rapid technological progress of the last few decades, personal computers have become repositories for company information and scientific data, documents, electronic mail as well as personal collections of media, like photographs, video or music. Computer users, however, in order to take advantage of this memory complement offered to them, have to invest more and more time into managing and organizing their collections and repositories because, if they don't, retrieving information from them when necessary will be nearly impossible.

Furthermore, in current computer systems the user interaction paradigm is based on functionally defined applications (word processing, address management, internet browsing) and on the storage, organisation and retrieval of information in files or databases, the content types and structure of which are determined by the units of operation of the applications. However, real activity, whether for work or leisure, crosses application boundaries, may involve portions of files, and interlinks fragments of both. Users should not have to focus on managing their information but rather on performing the tasks this information is to be used for.

Recent research in the domain of Personal Information Management (PIM) and Task-centered Information Management (TIM) has recognized the need for a paradigm shift towards more task- and activity-oriented systems [21]. Ontologies, as semantic networks with a structure very similar to the one used by the human brain for storing long term knowledge, may be very useful as the basis of such a system. They offer a flexible and expressive layer of abstraction, very useful for capturing the semantics of information repositories and facilitating their retrieval either by the user or by the system to support user tasks. To this end, if combined with appropriate "intelligent" mechanisms, they may become useful tools to record semantics related to documents and tasks and function as an extension to the user's own memory, available both for the user and the system.

This work explores the application of the spreading activation theory of the human memory [1] [2] on ontologies in order to create a context inference model for an ontology-based PIM/TIM prototype system. The following section briefly outlines the architecture of this system, focusing on the ontology and context inference module, whereas the next describes briefly the human memory theories that have been the basis of this work, along with an example of spreading activation. The following section describes the creation of a personal ontology for the user domain, followed by the description of the spreading activation algorithm. The next section discusses briefly the results of a preliminary evaluation of the spreading activation module. Finally, after a brief outline of related work, the last section presents the conclusions and outlines future work.

TOWARDS A PERSONAL INTERACTION MANAGEMENT SYSTEM

The motivation of our work on personal ontologies and spreading activation has been the vision of more activity-centric computing and the general aim of moving from systems focusing on the management of personal information (i.e. PIM) to systems focusing on the management of personal interaction. We define Personal Interaction Management System (PIMS) to be a system that supports the user in executing tasks in an interactive and efficient way, providing at the same time effective and transparent mechanisms for maintaining the user's personal document collection.

In order for a PIMS to be effective, it should provide effective mechanisms for user profiling, semantic storage of documents and context inference. Figure 1 shows a sketch view of the main components a PIMS must include to support this functionality. The information side (documents, emails etc.) is linked to the computation side (actions) through two main components:

1. A *recogniser* finding suitable fragments of the raw information that are semantically meaningful and that can be used to initiate or feed into actions

2. A *personal ontology* that contains knowledge specific to the user (people, projects, etc.).

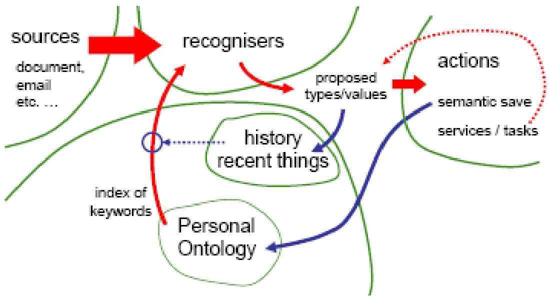

Figure 1. Basic outline of a PIMS system

These two feed into one another. The various terms, names, emails, etc, in a personal ontology can yield keywords to be matched against text or semi-structured sources. So an increasingly rich personal ontology will lead to better identification of suitable loci for action. Furthermore, as users perform actions the way in which they use information, the results of their activities can be used to enrich the ontology. For example, if a piece of text is used to search in a gazetteer it suggests that (i) it is a place name - that is we know more about its type and (ii) it is a place name that is important to the user - so will be suggested to be added into the personal ontology.

Figure 1 also shows a history sub-component related to the personal ontology. We need to record what is done in order to both (a) establish a sense of context and (b) be able to allow the system to gain some understanding of the user's ongoing activities.

Both of these require inference mechanisms which sit outside this picture, using the information from the personal ontology and history and then feeding this in to modify the recognition and action selection. For task inference we are using a bottom up approach described in more detail in [23] and prototyped over actions on web forms. For context inference we are using spreading activation over the Personal Ontology, which is the focus of this work. The PIMS system architecture and individual components are discussed in more detail in [21] and [22], whereas this work focuses on spreading activation for context inference.

ONTOLOGIES AND PERSONAL INFORMATION MANAGEMENT

According to [5], an ontology is an explicit specification of a conceptualization. The term "conceptualization" is defined as an abstract, simplified view of the world that needs to be represented for some purpose. It contains the concepts (classes) and their instantiations (instances) that are presumed to exist in some area of interest and their properties and relations that link them (slots). The term "ontology" is borrowed from philosophy, where an ontology is a systematic account of Existence. This section presents the creation of a personal ontology to be the basis of the intelligent context inference mechanism of our PIMS system.

Ontologies in PIM systems

Using an ontology to model semantics related to the user personal domain has already been proposed for various applications like web search [12], [13]. 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 [14], GNOWSIS [15] and the semantic desktop search environment proposed in [16].

In the context of our proposed PIMS system the personal ontology has a very important part to play. On one hand, it may constitute a useful repository of information related to many aspects of the user's personal and professional life. There the user will be able to store and access information on contacts (friends, colleagues, etc), activities (like a research project or a hobby), events (like project meetings, conferences, etc), documents (collected books and research papers, etc) and tasks. With the appropriate interface, the ontology may become an easily customizable repository of information that may serve as a memory complement for the user. On the other hand, coupled with intelligent mechanisms, the ontology may become invaluable for context inference in the process of supporting the user tasks through task inference.

To this end, we have created an ontology for the user's personal collection domain. This ontology has been created taking into account existing profile models in applications as well as related research in the area of profiling.

Figure 2 Overview of the Personal Ontology with the upper levels expanded

Ontology Creation

Creating a personal ontology, either automatically, manually or semi-automatically is not an easy task. In order for such an ontology to be truly personal, it should be able to reflect the user individuality, but, it should do so in the context of a specific general model that will enable exchange of information between users and will be usable by computers. This is the main reason why the personal ontology model we propose encompasses a basic core of general concepts that may be enriched to accommodate several user stereotypes or individual profiles. The addition of new classes may be accomplished both at the ontology designer and the end user level.

Details on the creation of the personal ontology may be found in [6]. The version of the personal ontology used in this work is an extension of the one in [6], 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 [8]. Figure 2 presents an overview of the upper levels of the class hierarchy.

The personal ontology attempts to encompass a wide range of user characteristics, including personal information as well as relations to other people, preferences and interests. The ontology may be extended through inheritance and the addition of more classes, as well as concept instantiation according to the needs of user stereotypes or individuals.

The personal ontology classes are divided in two main groups, which comprise the two upper levels of the ontology, "Value class" and "Thing" (Figure 2).

"Value Class" contains a description of information items that are more complex than simple data types but are not self-contained enough to be included in the ontology as separate entities. These among others include dates, URLs, telephone numbers, zip codes and names and may be used as slot types for the concepts of the "Thing" sub-hierarchy. Instances of these concepts may serve as information items identified automatically by a PIM or TIM application.

Class "Thing" (Figure 2) contains both abstract and tangible things, which may be objects, living organisms and concepts. Classes "Interest Type" and "Preference Type" model interest and preference hierarchies as the ones suggested in [12] and [13]. Class "Self" (highlighted in Fig. 2), a direct subclass of "Person", models the profiled user.

The ontology has been modeled using Protégé [7], a widely used open source ontology management tool with a well-defined API for creating plug-ins. To this end it was selected for the implementation and testing of the spreading activation algorithm over the personal ontology. The following section describes human memory theories on which our work has been based, along with a brief example of spreading activation.

SPREADING ACTIVATION IN THE HUMAN BRAIN AND IN ONTOLOGIES

Different Timescales of Human Memory

The human memory operates on multiple timescales. According to the model that Atkinson and Shiffrin proposed in 1968 [4], there are two distinct memory stores:

- **Short term memory**– the things we are currently thinking about. This is short lived (10-30 secs) unless it is constantly rehearsed.
- **Long term memory** – the things we have learnt and stay with us for years (possibly forever), but may be more or less easy to retrieve.

Short term memory is held in patterns of electrical activity whereas long term memories are formed by actual synapse growth. 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. These include the context of "what am I doing now" as well as recent episodic memory of "what has happened in the last few minutes". This in-between or 'mezzanine' memory is not well dealt with in the literature as it is too 'fast' for neuron growth. It 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.

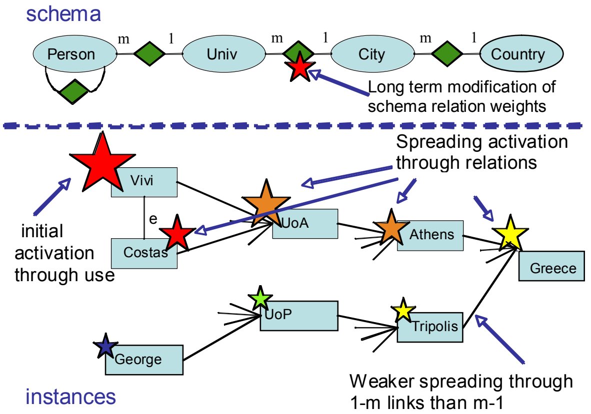

Figure 3. An example of spreading activation over the personal ontology

The spreading activation theory [1] has proven to provide a model with a high degree of explanatory power in cognitive psychology [2]. The main advantage of this model is that it captures both the way knowledge is represented and also the way it is processed. According to this theory, 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 [2].

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 [3].

An Example of Spreading Activation through the Personal Ontology

To provide a clearer view of the use of spreading activation over the personal ontology, consider Figure 3. It presents an ontology schema with classes for Persons, Universities, Cities and Countries and a number of relationships (unlabelled, but all obvious except that the Person to Person relationship is PhD supervisor).

In the lower part of the figure are a number of instances on the classes and their relationships: from the class Person the instances Vivi, Costas and George have been added to the personal ontology. Costas supervises Vivi's PhD. Costas and Vivi both work for UoA which is in Athens, Greece and George works for UoP which is in Tripolis, Greece.

Suppose that there is some sort of initial reason for focusing on Vivi, perhaps she has been selected by the user for interaction or has appeared in a recent email. Vivi is therefore given a high initial activation level. The entities directly connected to Vivi (in this case Costas and UoA) are then also given activation, less than Vivi's though. Because UoA receives activation entities connected with it, in particular the City it is in, Athens, are given a small share of its activation. This then further spreads to Greece, the country in which the City of Athens is situated.

This activation will also flow backwards through relations so that Tripolis becomes activated because Greece is, and so on. When activation spreads through a 1–m relation it will be weaker than through an m–1 relation, as there are many things to which it is spreading. Also note that some entities may receive activation through multiple routes. For example, UoA receives activation because it is the place Costas works as well as because of its direct link to Vivi. This may then lead to cycles of activation, hence the importance of 'throttling' the spread of activation where there is large fan out (as in 1–m relations).

Some relations may be deemed more important than others and hence given a weighting in the schema, which can then be used when computing the activation spread. In addition, over time these could be adjusted so that where entities in either side of a relation are often active together the relation may grow in weight. The amount of activation an entity receives is also related to (1) if the entity was activated by an external factor (i.e. being detected in an e-mail), (2) if the entity was recently active (i.e. detected very recently in the user's tasks), (3) if the entity has long term importance to the user.

The result of the spreading activation will be a set of entities that have been found relevant to the user's current task and therefore given priority during the task inference stage. The context inference module, based on spreading activation, has been inspired by the human memory spreading activation model, which is briefly presented in the following section.

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 exceeds some value then the medium-term activation 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 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 following section presents the algorithm and implementation inspired by the human memory model through spreading activation on the personal ontology

Spreading Activation Algorithm

The spreading activation through the personal ontology algorithm assumes, to avoid repetition in the formulae, that the inverse of each relation is explicitly recorded in the ontology schema. For a real schema this means that all qualifiers would have to range over relations and their inverses. Also the weight (strength) of a relation is directional, so that there is a difference between the weights depending on which direction the relation is traversed. Again to simplify the formulae, property values will be ignored.

Given this we have a set of relations L, of entities (instances) E and instances of relationships (statements) S. Every statement is a relation between specific entities:

$$S = L \times E \times E$$

So a typical statement is of the form r(e1,e2), where r is a relation and e1 and e2 are instances.

The current state of the ontology is then simply a set of statements:

$$\text{OntologyState} \equiv OS \subset S$$

An activation state over such an ontology is then an activation level (real number) assigned to each entity:

$$\text{ActivationState}: E \rightarrow R$$

The set of all possible activation states over an entity set E will be denoted as AS(E). We will refer to the three time scales of system activation as STA, MTA and LTA. **STA** (Short Term Activation) refers to things that are currently active, **MTA** (Medium Term Activation) to things that have been recently active (and most probably still are), whereas **LTA** (Long Term Activation) to things that are important to the user in the long term. There is also a 'trigger' activation, **IA** (Immediate Activation), corresponding to the things that are in some way important directly due to the current task/interaction; for example, the things that are in the currently viewed e-mail or web page.

$$STA, MTA, LTA, IA \in AS(E)$$

Also we assume that each relation, r, has a long-term weight LTW(r) that is initialized according to the cardinality of the relationship (1-1, 1-m, m-1, m-n).

The basic steps of the algorithm may be summarized as follows:

1. Initialize appropriate weights and activations
2. Create a set with the currently active entities (entities e with **IA(e)** > 0), Active Set
3. Repeat:

   Compute STA(e) for the entities in the Active Set as well as their related ones

   For the related entities whose STA exceeds a threshold, place them in the Active Set

   Until <condition>

4. Update MTA and LTA activation weights if appropriate

We envision that the spreading activation algorithm will be triggered after each "event" in a PIMS system. With the term "event" in this case we refer to a user action that has resulted in the identification of ontology entities related to the current action. For example, the user opens an e-mail, and in it the sender name has been detected as well as the name of a research project the user currently participates in. Bearing this in mind, the following sections present in details the algorithm steps.

Updating Short Term Activation

Given a particular state of the STA, each entity e has an incoming activation **IN** given by

$$IN(e) = \sum LTW'(r) \times STA(e'),\ r \in R \wedge \exists\, e' \in E: r(e, e') \in \text{OntologyState}$$

The value of LTW'(r) is in fact the relation LTW value divided by the number of entities e' is related with through this relation, i.e. the "fan out" of the relation.

The formula for the STA then becomes:

$$STA(e) = f\,(IA(e), IN(e), MTA(e), LTA(e))$$

The function will typically count IA strongly, and only take into account MTA and LTA where either IA or IN(e) is non-zero. For example, a possible function might be:

$$f(ia, in, mta, lta) = (A \times ia + B \times in) * (1 + (C \times mta + D \times lta))$$

The non-linear term means that long- or medium-term activation are not in themselves sufficient to cause short-term activation, but do strengthen the effect of STA.

The result of the STA update function is passed through a sigmoid function ($S(sta) = \frac{1}{1+e^{-sta}}$) to emphasise the difference between large and small activations and to cap the largest.

The equation for STA is recursive and is applied on the set of activated entities of each step.

Spreading Activation Termination Conditions

For the number of iterations during the spreading activation algorithm step for STA computation, two options have been considered:

1. **Full Spreading of Activation**: Repeat spreading computations for the whole ontology, until it reaches a stable state.
2. **Constrained Spreading of Activation**: Repeat for a specific number of iterations.

The first option was not selected for a number of reasons Firstly, bearing in mind that a personal ontology serving as memory aid for a user may contain thousands of instances, applying spreading activation on the whole ontology would not be very efficient, especially in applications like task information management where access to the ontology is very frequent.

Furthermore, the existence of cyclic paths in the ontology graph means that the spreading activation process won't end because of the loops. A way to go around this would be to detect already visited entities and avoid loops by not spreading activation to them again. However, this could be a problem as well, as an entity may be related to more than one active entities. For example, an entity e1 may receive activation from the directly connected to it e2 during the first iteration step and from the entity e3 that is related to e1 through e4 during the second iteration step (e1→ e2 and e1→ e4 → e3). So excluding already visited entities has also been rejected.

As a result, for the needs of our implementation of spreading activation we opted for constrained spreading activation, also suggested in [10]. We feel that this variation of the algorithm is closer to the human memory spreading activation process where the finite rate of firing and tome available means that only a limited number of 'steps' are taken. To this end, we apply spreading activation on the ontology for a specific number of iterations. Futrhermore, if the personal ontology is a 'small world' then it may be that activation can in principle spread across the network in a very few iterations, before feedback loops become too powerful. The optimum number of iterations is still an issue for experimentation and it is directly related to the needs of the specific application as well as the ontology weights and parameters.

Updating Medium and Long Term Activation

At the end of the spreading activation cycle, MTA and LTA are updated.

We simply increment MTA if the STA exceeds a value:

$$\text{if } (STA(e) > threshold_{STA})\ MTA'(e) = MTA(e) + \delta_{MTA}$$

And similarly for LTA:

$$\text{if } (MTA(e) > threshold_{MTA})\ LTA'(e) = LTA(e) + \delta_{LTA}$$

However, there are several issues to consider here. One is the exact values of δ_{MTA} and δ_{LTA}. Furthermore, each entity MTA and LTA apart from being incremented when active, it should also be decayed when inactive. This gradual decay should reflect the fact that memories tend to fade or become less readily accessible when they fall into disuse for a sufficient period of time. Due to the differences of medium and long term activations, however, their update mechanisms should be examined separately.

MTA Increase and Decay

MTA expresses the number of things that are recently and currently "active" and may involve the user's current tasks. This suggests that the total amount of MTA weights in the ontology should remain relatively steady, in order to reflect the fact that the user's divided attention among many tasks (and, subsequently, entities), results in less attention paid to each particular task/entity.

To this end, we define a constant, MaxMTATotal, which represents the maximum value for the sum of all MTA weights in the ontology.

The decay of MTA is accomplished with the following process:

Every T steps:

1. The total amount of MTA increase over the T steps, s_{MTA}, is recorded
2. We set $\lambda_{MTA} = s_{MTA}$ / MaxMTATotal as the decay factor

3. For every entity e, the new MTA is computed: $MTA'(e) = (1 - \lambda_{MTA}) * MTA(e)$

The frequency of the MTA decay as well as the maximum total of MTA weights should be adjusted according to the needs of the specific application. For a TIM system MTA probably should be updated after each “event”.

In special cases when IA on its own exceeds some value or was caused by some specific effect, MTA could be increased directly.

LTA Increase and Decay

LTA reflects the long term importance of entities: it represents the fact that some things have been important to the user several times in the past. Even if currently or in the recent past they may not have been active, they most probably will be again in the future. Entities like the user’s address or parents can never be entirely forgotten.

As a result, when decaying LTA weights, it should be made sure that the decay does not result in important things having their LTA weight value gradually returning to zero. A way to accomplish this is to make sure that the LTA of an entity never decays to less than a percentage (n%) of its maximum value.

We define as maxLTA(e) the maximum LTA value an entity e has ever received Furthermore, we define two constants, λ_{LTA} as the decay constant that depends on the time interval between each decay and minPerc as the minimum percentage of the entity maxLTA value that the LTA of an entity may reach when decayed. The LTA decay is computed using the following process:

At the designated time points, for every entity e:

```
if (LTA(e) > maxLTA(e)) {maxLTA(e) = LTA(e)}
minLTA_e = minPerc * maxLTA(e);
if (LTA(e) > minLTA_e ) {
        delta_e = λLTA * (LTA(e) - minLTA_e)
        LTA’(e) = LTA(e) - delta_e
}
```

An issue here is the definition of the time interval between consecutive decays. For the moment, events are considered as a time unit in order to measure the passage of time. The LTA decay time intervals in a TIM application should take into account other factors like the real time elapsed and the computer usage time elapsed.

LTW and Relation Weights

Relation weights are a very important issue in the spreading activation framework.

Three levels of relation weights may be distinguished:

(1) The relation as a whole, which is expressed by the relation Long Term Weight – LTW.

(2) Weights on the precise instance of a relation, that is for a specific e_1, e_2 with a relation r between them, we could assign a weight dependent on:

   (a) Whether the relation was important in spreading activation

   (b) Whether both e_1 and e_2 have received high activation

(3) Weights on the relation for an individual entity, that is given an entity e1 for the specific instance of the relation r in e1, the LTW’ is computed as the relation LTW/k, where k is the relation fan-out for the specific entity, i.e. the number of entities with which e1 is connected through the specific relation r.

For the moment, the spreading activation algorithm has been implemented with the third option for LTW weights. As an example, if we look at the class-students relationship, then if a particular class has many students we may want to reduce the spread accordingly, closer to an activation budget model where if a node has so much activation it spreads some of it to other nodes, but has to share amongst the ones connected to it. A model of this form could penalise well-connected entities (which are likely to be central and generally important ones), but without some bias of this form such entities might just become 'fixations' of the ontology.

A well-connected entity bound to be a fixation in the ontology is the instance of “Self”, which represents the user in the ontology. As this is the user’s personal ontology, it is natural for it to be the best-connected one, a focal point related to almost all entities in the ontology. This special characteristic of the “Self” instance affects the spreading of activation, so it has been treated as a special case and we have experimenting both with its inclusion and exclusion during the execution of the spreading activation algorithm.

Working with weights on relation instances remains an open issue that requires further research, as it is not yet clear what would their interaction would be with LTW weights.

As a final point, LTW weights could also be adjusted to reflect the fact that if it appears that usually when an entity is active so are all those it is related to through a particular relationship r, then this would suggest that that relationship should be given a higher weight.

IF foreach $e \in dom(r)$,
  (i) $MTA(e) > threshold_{R1}$
  AND
  (ii) for most e': $r(e, e') \in OntologyState$, $MTA(e') > threshold_{R2} \times MTA(e)$
THEN
  increase LTW(r)

However, this needs to be applied with some care as it is a positive feedback loop – stronger LTW leads to stronger incoming activation and hence makes it more likely that related things are active together, further increasing the LTW of the relation. Until the exact implications of LTW update have been identified, it has not been included in the spreading activation algorithm.

LTA, STA and MTA Initialization

For the spreading activation algorithm to yield useful and meaningful results, there are two very important factors. The first is a rich personal ontology and the second the correct weight and parameter adjustment and initialization. For testing the algorithm and after preliminary experimentation, we concluded at a set of default values for these parameters and weights. These are set as default values in the Protégé plug-in for the evaluation of the algorithm, described in the following section. It is obvious that a different set could be used according to the needs of the application that would use the algorithm.

PRELIMINARY EVALUATION

In order to fully evaluate the spreading activation algorithm, it should be integrated in a TIM prototype and observe its effectiveness in the working environment of the user. For the moment, in order to achieve the fine-tuning of the algorithm parameters and locate problems and flaws, a testing platform has been created in Java in the form of a plug-in for the Protégé ontology editor [7].

This section presents the evaluation platform as well as the results of the preliminary evaluation.

The ActiveOnto Protégé Plug-in

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 (Fig. 4).

The plug-in may be found in [24] along with its installation instructions. 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.

As an example, the personal ontology in [8] may be used.

The plug-in offers the possibility to include or exclude the "Self" instance at will in the execution of the spreading activation algorithm, by appropriately setting or clearing a relevant checkbox.

Preliminary Evaluation

As a first step, we asked a researcher of our group to aid us in evaluating the plug-in. The researcher was asked to populate his personal ontology with instances relevant to his work and computer-related activities in general for the past six months.

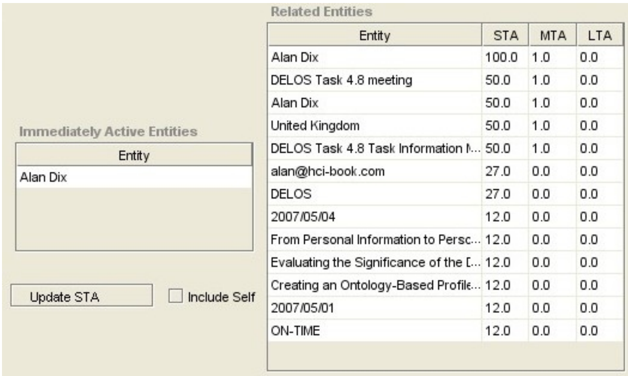

Figure 4. Part of the plug-in window showing the STA, MTA and LTA values for the entities that received STA activation value greater than 12, when entity "Alan Dix" got IA = 1.

Then, we asked him to go through his e-mail for the same duration and for each e-mail to set in the plug-in the IA activation of the instances that appeared in the e-mail and update STA, MTA and LTA weights.

Although the STA update seemed to generally produce relevant concepts and with meaningful activation values, the update of the MTA and LTA weights showed that using the e-mail in this way did not produce interesting results concerning these two weights. This is to be expected, as the e-mails constitute a series of, most of the time, irrelevant "events". Consequently, entities unrelated to one another followed in succession, resulting in constantly increasing and decreasing MTA values that never surpassed the appropriate threshold for increasing the LTA weight.

These results have lead us to construct more realistic usage scenarios than going through a continuous series of e-mails. We have decided to proceed with another non-formal evaluation that would be more task-oriented. This evaluation procedure is currently being designed.

RELATED WORK

Spreading activation is not a new concept in semantic networks related research. There is a number of proposed applications of spreading activation, especially in the area of information retrieval [11].

Crestani [19] proposes the use of spreading activation on automatically constructed hypertext networks in order to support web browsing. In this case, constrained spreading activation is used in order to avoid spreading through the whole network, as is the case with our implementation. Liu et al [18] use spreading activation on a semantic network of automatically extracted concepts in order to identify suitable candidates for expanding a specific domain ontology. Xue et al [19] propose a mining algorithm to improve web search performance by utilizing the user click-through data. Weighted relations between user queries and selected web pages are created and spreading activation is performed on the resulting network in order to re-rank the search results of a specific query.

Hasan in [20] proposes an indexing structure and navigational interface which integrates an ontology-driven knowledge-base with statistically derived indexing parameters, and the experts' feedback into a single spreading activation framework to harness knowledge from heterogeneous knowledge assets.

Neural networks and in particular Hopfield Networks [9] attempt to approach and simulate the associative memory again by using weighted nodes but at a different level. In this case, the individual network nodes are not separate concepts by themselves, but rather, in their whole, are used to represented memory states. This approach corresponds to the neuron functions of the human brain, whereas ours attempts to simulate the human memory conceptual network functions.

CONCLUSIONS AND FUTURE WORK

This work outlines a spreading activation over a personal ontology framework to be used in the context of a Personal Interaction Management System. The human brain and the theories related to the different levels of human memory and spreading activation have been the incentive of this work.

The proposed personal ontology model along with the mechanism that implements the spreading activation will be incorporated in the PIMS prototype currently under development to provide context inference to support user actions, as well as act as a memory supplement for the user.

Very important for the algorithm effectiveness in identifying "active" entities that are relevant to the ones appearing in the user's current task are the parameters for updating the weights. These parameters have been fine tuned to an extent through a process of preliminary testing, but there is still work to be done in this direction.

There is also a number of issues to be further investigated:

Weights on relation instances. To this end, an extension for the Protégé ontology model has been created, allowing the existence of weighted relations to be defined as slot types [26]. At the moment, the incorporation of these weights in the algorithm is being investigated, in order to decide if they offer some added value to the algorithm effectiveness

LTW update. The LTW relation weights are at the moment static. Their update according to occurring events and/or connected entities' STA, MTA and LTA variations, is being investigated.

Automatic tuning of spreading activation parameters, e.g. automatic alteration for the number of iterations.

Dealing with topics/contexts – People often do two interleaved –but not relevant to each other– tasks nearly simultaneously, e.g. someone works on a project and opens a window to see the latest football news. The task inference mechanism should be extended to recognize such cases and produce two distinct and specific tasks, instead of a single task consisting of irrelevant activities

Results of the preliminary, informal evaluation of the algorithm have shown it to be effective in inferring the context of user tasks. A more effective and thorough task-based evaluation is being designed in order to evaluate the update of MTA and LTA weights. However, in order to fully evaluate the algorithm, it should be incorporated in the PIMS prototype under development. There are various issues relevant to this incorporation, such as:

User interaction with the weighted ontology. Bearing in mind that the ontology will be a simplification of the user's semantic network on some aspects of his/her life, his/her contribution on defining the ontology entities and relations, as well as fine-tuning the weights will be invaluable. Although for an experienced user doing this directly on an ontology editor like Protégé would not be difficult, non-expert users would have trouble coping with such an editor interface, as well as the concept of the ontology itself. Furthermore, editing the ontology would add to the user's work a substantial overhead. To this end, semi-automatic methods for visualizing, updating and personalizing the ontology along with the weights are being investigated [25].

Representation of tasks/activities. Should 'types' of tasks and actual instances of things done be represented within the ontology as concepts, just like a friend's name, or should they be placed in some parallel but linked representation?

Scaling – the spreading activation so far has been created and tested for a personal ontology, but the personal ontology may well include links to external ontologies, even the whole web. Should we and how do we do this form of reasoning over very large ontologies?

REFERENCES

1. Anderson, J. R.. A spreading activation theory of memory. Journal of Verbal Learning and Verbal Behaviour, 22 (1983) 261-295

2. Sharifian, F., Samani, R. Hierarchical Spreading of activation, in Proc. of the Conference on Language, Cognition and Interpretation, Isfahan (1997), 1-10
3. Gazzaniga, M. S., Ivry, R. B., Mangun, G. R., Cognitive Neuroscience, The Biology of the Mind, pp 289-294, W. W. Norton & Company, New Ed edition (1998)
4. Atkinson, R.C. & Shiffrin, R.M. 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 (1968)
5. Gruber, Thomas R., A Translation Approach to Portable Ontology Specifications, Knowledge Acquisition, Special issue: Current issues in knowledge modelling, Vol 5, Issue 2 (1993) 199-220
6. Golemati M., Katifori A., Vassilakis C., Lepouras G., Halatsis C.: Creating an Ontology for the User Profile: Method and Applications, Proceedings of the First RCIS Conference (2007) 407-412, April 23-26, Ouarzazate, Morocco
7. Protégé, http://protege.stanford.edu/
8. Katifori, A., Vassilakis, C., Dix, A., Daradimos, I., Lepouras, G., Spreading activation user profile ontology, Available from http://oceanis.mm.di.uoa.gr/pened/?category=pub#ontos
9. Hopfield, J. Neural networks and physical systems with emergent collective computational properties. Proceedings of the National Academy of Sciences of the USA, 79 (1982), 2554 - 2588
10. Crestani, F. Retrieving documents by constrained spreading activation on automatically constructed hypertexts. In Proceedings of EU- FIT 97- Fifth International Congress on Intelligent Techniques and Soft Computing, pages (1997), 1210-1214, Aachen, Germany
11. Crestani F. Application of spreading activation techniques in information retrieval. Artificial Intelligence Review, 11(6) (1997), 453–482
12. J. Trajkova, S. Gauch. Improving Ontology-based User Profiles, Proc. of RIAO 2004, University of Avignon (Vaucluse), France, April 26-28 (2004), 380-389
13. S. Gauch, J. Chaffee, A. Pretschner. Ontology-Based User Profiles for Search and Browsing, User Modeling and User-Adapted Interaction: The Journal of Personalization Research", Special Issue on User Modeling for Web and Hypermedia Information Retrieval (2003)
14. V. Katifori, A. Poggi,. M. Scannapieco, T. Catarci, & Y. Ioannidis. OntoPIM: how to rely on a personal ontology for Personal Information Management, In Proc. of the 1st Workshop on The Semantic Desktop (2005)
15. Leo Sauermann. The Gnowsis Semantic Desktop for Information Integration, Proceedings of the 3rd Conference Professional Knowledge Management (2005)
16. Paul – Alexandru Chirita, Rita Gavriloai, Stefania Ghita, Wolfgang Nejdl, Raluca Paiu, Activity Based Metadata for Semantic Desktop Search, Proceedings of the 2nd European Semantic Web Conference (2005)
17. A. Maedche, S. Staab, "Mining Ontologies from Text", Proceedings of EKAW 2000 (2000) 189-202
18. W. Liu, A. Weichselbraun, A. Scharl, E. Chang. Semi-Automatic Ontology Extension Using Spreading Activation, Journal of Universal Knowledge Management, vol. 0, no. 1 (2005), 50 - 58
19. Gui-Rong Xue, Hua-Jun Zeng, Zheng Chen, Wei-Ying Ma, Wensi Xi, Weiguo Fan, Yong Yu. Optimizing Web Search Using Web Click-through Data. (2004): 118-126
20. Hasan, Md Maruf (2003) A Spreading Activation Framework for Ontology-enhanced Adaptive Information Access within Organisations. In Proceedings of the Spring Symposium on Agent Mediated Knowledge Management 2003 (2003) Stanford University, California, USA
21. T. Catarci, A. Dix, A. Katifori, G. Lepouras and A. Poggi. Task-Centered Information Management. In DELOS Conference 2007 on Working Notes, 13-14 February 2007, Tirrenia, Pisa (Italy), C. Thanos and F. Borri (eds.) (2007), 253-263
22. G. Lepouras, A. Dix, A. Katifori, T. Catarci, B. Habegger, A. Poggi, Y. Ioannidis (2006). OntoPIM: From Personal Information Management to Task Information Management, Personal Information Management, SIGIR 2006 workshop, August 10-11, (2006), Seattle, Washington
23. T. Catarci, B. Habegger, A. Poggi, A. Dix, Y. Ioannidis, A. Katifori, and G. Lepouras. Intelligent user task oriented systems. In In Proc. of the Second SIGIR Workshop on Personal Information Management (PIM 2006), 2006
24. Katifori, A., Dix, A., Vassilakis, C., Spreading activation Protégé plugin, available from http://oceanis.mm.di.uoa.gr/pened/?c=pub#plugins
25. A. Katifori , E. Torou, C. Halatsis, G. Lepouras and C. Vassilakis, A Comparative Study of Four Ontology Visualization Techniques in Protégé: Experiment Setup and Preliminary Results, Proceedings of the IV 06 Conference (2006)
26. C. Vassilakis, G. Lepouras, A. Katifori, wt-Protégé – An Extension for Protégé Supporting Temporal and Weighted Data, Technical Report TR-SSDBL-07-001, (2007), http://wt-protege.uop.gr