Ontologies and the Brain: Using Spreading Activation through Ontologies to Support Personal Interaction

Akrivi Katifori

Department of Informatics and Telecommunications, University of Athens, Panepistimiopolis,15784, Athens, Greece

Costas Vassilakis

Department of Computer Science and Technology, University of Peloponnese, Terma Karaiskaki 22100, Tripoli, Greece

Alan Dix

Computing Department, Lancaster University, InfoLab21, South Drive, Lancaster, LA1 4WA, UK

Abstract

Ontologies, as knowledge engineering tools, allow information to be modelled in ways resembling to those used by the human brain, and may be very useful in the context of personal information management (PIM) and Task Information Management (TIM). This work proposes the use of ontologies as a long-term knowledge store for PIM-related information, and the use of spreading activation over ontologies in order to provide context inference to tools that support TIM. Details on the ontology creation and content are provided, along with a full description of the spreading activation algorithm and its preliminary evaluation.

Key words: Spreading activation, Personal interaction, Ontology, Personal ontology, User tasks

1 Introduction

As a direct result of the rapid technological progress of the last few decades, personal computers are becoming an important part both of our professional and personal life. Due to the advantages they offer for information storage, they have become repositories for numerous and diverse data collections including company information and scientific data, documents, electronic mail as

Email addresses: vivi@di.uoa.gr (Akrivi Katifori), costas@uop.gr (Costas Vassilakis), alan@hcibook.com (Alan Dix)

well as personal collections of media, like photographs, video or music. These repositories offer the possibility of recording and storing a huge amount of information, an amount unthinkable in the past when only the human brain and conventional means like paper were available. However, in order to take advantage of this potential memory supplement, computer users 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, user interaction is focused on functionally defined applications (word processing, address management, internet browsing); in this context storage, organization and retrieval of information in files or other databases is 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 performing the tasks this information is meant 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, i.e. towards the management of personal interaction (Catarci, Dix, Katifori, Lepouras, & Poggi, 2007; Dix, Katifori, Poggi, Catarci, Ioannidis, Lepouras, & Mora, 2007). 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 on ontologies in order to create a context inference model to be used in any ontology-based PIM/TIM prototype system. The subsequent section briefly presents the human memory theories which constitute the basis of this work. The following one outlines the use of an ontology in PIM and TIM systems and describes a personal ontology to be the basis of such a system. Next, the spreading activation over the personal ontology algorithm is described in detail, followed by a brief presentation of related work on spreading activation. Afterwards, the results of a preliminary evaluation of the spreading activation module are discussed. The last section presents the conclusions and outlines future work.

2 The human memory and spreading activation

The human brain is a very powerful information storage, computational and reasoning mechanism. Its exact structure and function is still under scrutiny by scientists in order to decode its secrets and probably mimic its functionality in artificial constructs. This section briefly outlines existing theories about the mechanism of the human memory.

2.1 *Different timescales of human memory*

The human memory operates on multiple timescales. According to the model that Atkinson and Shiffrin proposed in 1968 (Atkinson & Shiffrin, 1968), there are two distinct memory stores: short-term memory, and long-term memory.

- **Short-term or working memory** – the things we are currently thinking about. This is short lived (10-30 secs) (Peterson & Peterson, 1959) unless it is constantly rehearsed. It is also limited to hold only 5-9 chunks of information (seven plus or minus two) (Miller,1956) where a chunk is any meaningful unit. A chunk could refer to digits, words, chess positions, people's faces etc. The concept of chunking and the limited capacity of short-term memory became a basic element of all subsequent theories of memory.

- **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. Long-term memory appears to have an almost limitless capacity to retain information, but it could never be measured as it would take too long. This information seems to be encoded mainly in terms of meaning (semantic memory) but also retains procedural skills and imagery.

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 working 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; the main exception being work on *long-term working memory* focused on the ongoing establishment of context during text comprehension (Ericsson & Kintsch, 1995), also in the human factors literature there is substantial work on *situation awareness* in command and control situations. However, the mechanisms for these forms of memory are unclear as they are too "fast" for neuron growth. It may be in part due to more sustained electrical states or chemical changes in neurons called long-term potentiation or LTP (Lømo, 2003), which are known to last for anything from seconds to hours.

Long-term potentiation (LTP) is an increase in the strength of a chemical synapse that lasts from minutes to several days. LTP was discovered in the mammalian hippocampus by Terje Lømo (Lømo, 2003) and has remained a popular subject of neuroscientific research since. It is widely considered one of the major mechanisms by which memories are formed and stored in the brain.

LTP has been observed both in experimental preparations in vitro and in living animals (in vivo). Under experimental conditions, applying a series of short, high-frequency electric stimuli to a synapse can strengthen, or potentiate, the synapse for minutes to hours. In living cells, LTP occurs naturally and can last from hours to days, months, and years. Neurons connected by a synapse that has undergone LTP have a tendency to be active simultaneously: after a synapse has undergone LTP, subsequent stimuli applied to one cell are more likely to elicit action potentials in the cells to which it is connected.

Because changes in synaptic strength are thought to underlie memory formation, LTP is believed to play a critical role in behavioral learning. In fact, most neuroscientific learning theories regard long-term potentiation and its opposing process, long-term depression, as the cellular bases of learning and memory.

Based on the aforementioned theories and studies on human memory, for the needs of our research we distinguish three timescales of human memory (Dix, 2006), summarized in Table 1.

Table 1. Different timescales of human memory

human memory	timescale	mechanism	brain effect
long term	indefinite	physical	synapse growth
short term (Miller's 7±2)	10-30 seconds	electrical	neuron firing
mezzanine (no "proper" name)	minutes to hours	chemical	long-term potentiation LTP

2.2 The Spreading Activation Theory of Memory

The spreading activation theory (Anderson, 1983) has been proven to provide a model with a high degree of explanatory power in cognitive psychology (Sharifian & Samani, 1997). 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 (Sharifian & Samani, 1997).

The strength of the connection and the distance between the nodes 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 (Gazzaniga, Ivry, & Mangun, 1998). This spread of activation serves to make related areas of the memory network more available for further cognitive processing. Speed and probability of accessing a memory is determined by its level of activation, which in turn is determined by how frequently and how recently it has been used (Anderson, 1995), as expressed by the strength of nodes and connections. This strength decays over time (Anderson, 1983).

Empirical evidence has shown that the amount of activation of a concept node is a function of the strength of the associative pathway between the node and the source of activation (Lock, 1982). Moreover, the amount of activation spreading from a given node along a pathway is a function of the strength of that pathway relative to the sum of the strengths of all paths emanating from that node (Reder & Anderson, 1980). The theory of spreading activation also models the *fan effect*. This refers to the distribution of the activation emanating from one node between all its associated nodes, diminishing thus the amount of activation they receive (Anderson, 1983).

3 Ontologies and personal information management (PIM)

Taking into account the aforementioned theories on the functionality of the human memory, this section presents the creation of a personal ontology which will serve as a basis of an intelligent mechanism to support PIM.

Ontologies have proven to be useful tools for representing semantics both in the web and in the personal document collection domains. Their strength lies in the fact that they offer a layer of abstraction that at the same time may be interpreted by both humans and machines. Furthermore, the conceptual structure they offer seems to be close to the human brain semantic network memory model and, as a result, to be very useful and meaningful for the representation of concepts related to the user domain of interest.

Using an ontology to model information related to the user personal domain has already been proposed for various applications like web search (Crestani, 1997a; Crestani, 1997b; Gauch, Chaffee, & Pretschner, 2003; Trajkova & Gauch 2004). Most of these approaches use ontologies only as concept hierarchies (e.g. hierarchies of user interests) without particular semantic complexity. The value of ontologies for PIM has also been recognized and there is on-going research on incorporating them in PIM/TIM systems like OntoPIM (Katifori, Poggi, Scannapieco, Catarci, & Ioannidis, 2005; Catarci, Dix, Katifori, Lepouras & Poggi, 2007; Lepouras, Dix, Katifori, Catarci, Habegger, Poggi, & Ioannidis, 2006; Catarci, Habegger, Poggi, Dix, Ioannidis, Katifori, & Lepouras, 2006), GNOWSIS (Sauermann, 2005) and the semantic desktop search environment proposed in (Chirita, Gavriloai, Ghita, Nejdl, & Paiu, 2005).

In the context of any PIM/TIM 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 (such as 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 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. The following sections provide an ontology definition and introduce the personal ontology developed by our group and used for testing the spreading activation algorithm.

3.1 Ontology definition

According to (Gruber, 1993b), 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 objects, concepts and other entities that are presumed to exist in some area of interest and the relations that hold them. The term "ontology" is borrowed from philosophy, where an ontology is a systematic account of Existence.

Therefore, as defined in (Noy & McGuiness, 2001), an ontology is a formal explicit description of concepts, or classes in a domain of discourse. Properties or slots of each class describe various features and attributes of the class, and restrictions on slots (called *facets* or *role descriptions*) state conditions that must always hold to guarantee the semantic integrity of the ontology. Each slot has a type and could have a restricted number of allowed values. Allowed classes for slots of type Instance are often called the range of a slot. An ontology along with a set of individual instances of classes constitutes a knowledge base.

A more mathematical definition can be found in (Amann & Fundulaki, 1999) and may be adapted to the terminology used throughout this work as follows:

An ontology is a triple O = (C, S, isa) where:

(1) $C = \{c_1, c_2,\ldots, c_m\}$ is a set of classes, where each class c_i models a set of real-world objects (class instances),

(2) $P = \{p_1, p_2,\ldots, p_n\}$ is a set of properties (slots), where each property p_i is either:

- a simple-typed property of a class, taking values from a domain such as "Integer" or "String" or
- a binary typed role, representing a relationship between classes.

(3) isa ={isa1, isa2, …, isap} is a set of inheritance relationships defined between classes. Inheritance relationships carry subset semantics and define a partial order over classes, organizing classes into one or more tree structures (multiple tree structures can occur if more than one classes have no parent; if multiple inheritance is allowed, classes are organized in directed acyclic graphs instead of trees).

In order to accommodate the individual instances (entities), this definition can be extended with a fourth element $E = \{e_1, e_2,\ldots, e_q\}$, where every e_w is an instance of some class $c_x \in C$. Each instance i_w includes a concrete value for every property p_y associated with c_x or its ancestors (as defined by the *isa* set). Ontologies can be represented as directed graphs where nodes correspond to classes and instances, and links to roles and isa relationships.

Ontologies may be enriched with axioms and production rules (Corcho & Gómez – Pérez, 2000). Axioms model sentences that are always true. They are included in an ontology for several purposes, such as constraining its information, verifying its correctness or deducing new information. Production rules follow the structure *If... Then...* and are used to express sets of actions and heuristics which can be represented independently from the way they will be used.

3.2 A personal ontology to model the user domain

Ontologies may become a very useful tool in personal information management as they offer the possibility for rich and flexible formalism of concepts and tangible things related to the user domain. Hoqwever, creating a personal ontology, either automatically, manually or semi-automatically is not an easy task (Maedche & Staab, 2002). In order for such an ontology to be truly personal, it should be able to reflect the user individuality, but, on the other hand, 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 is based on a basic core of general concepts that may be enriched to accommodate several user stereotypes or individual profiles. 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. Besides storing information describing real-world concepts and things, the ontology complements each entity with data modeling its importance for the different timescales of human memory; the spreading activation algorithm for computing and exploiting this data is presented in the next section.

For the creation of the ontology we adopted a top-to-down approach. Throughout the whole process, Gruber's design criteria (Gruber, 1993a) (clarity, coherence, extensibility, minimal encoding bias, minimal ontological commitment) were taken into account. The ontology was modelled manually as automatic or semi-automatic methods were not applicable at the level of user information we were interested in.

In order to create a simple yet comprehensive set of upper level concepts for the personal ontology, profile information models maintained by various applications, like instant messengers (ICQ, 2008) and community websites (Facebook, 2008; MySpace, 2008), and proposed by researchers, like (Tazari, Grimm, & Finke, 2003; Kobsa, 1993; Trajkova & Gauch, 2004; Gauch, Chaffee, & Pretschner, 2003; Gruber, 1993a) were examined and general ontologies like the ones presented in (Miller, 1990) were taken into account along with the MIME directory profile vCard (Renato, 2001).

Details on the creation of the personal ontology may be found in (Golemati, Katifori, Vassilakis, Lepouras, & Halatsis, 2007). The version of the personal ontology used in this work is an extension of the one in (Golemati, Katifori, Vassilakis, Lepouras, & Halatsis, 2007), 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, to be presented in the next section. The ontology, along with example instances may be found in (Katifori, Vassilakis, Dix, Daradimos, & Lepouras, 2007). Figure 1 presents an overview of the class hierarchy. The personal ontology classes are divided in two main groups, which comprise the two upper levels of the ontology, "Value Class" and "Thing".

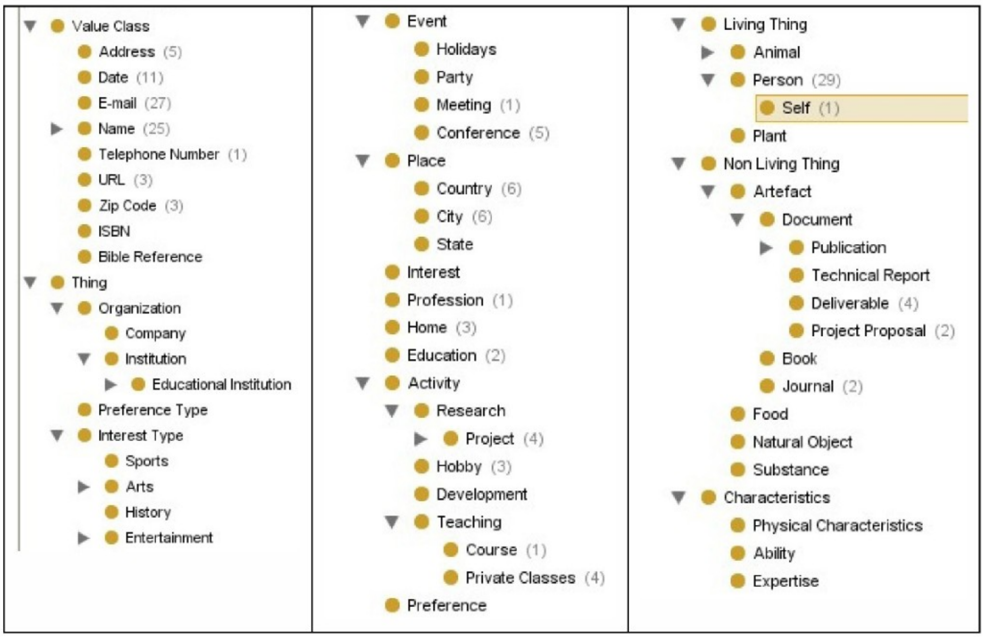


Figure 1 - Overview of the class hierarchy

The "Value Class" subhierarchy contains a description of information items that are more complex than simple data types (e.g. integers or strings) because their values adhere to specific syntax rules and/or comprise of multiple parts, but on the other hand it would not be useful to place them in the main 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 classes of the "Thing" sub-hierarchy. Instances of these classes may serve as information items identified automatically by a PIM or TIM application.

The class "Thing" contains both abstract and tangible things, which may be objects, living organisms and concepts. They may be:

- Classes belonging that represent person characteristics (Activity, Ability, Characteristics, Preference, Interest, etc)
- Classes that model things relevant to a person like events s/he was present at, his/her home, the university where s/he studied in, etc. This group of classes may be expanded according to the user's stereotype or particular interests and activities. For example, for a researcher, concepts like "Project", "Meeting" or "Conference" would be present in the ontology as well.

The classes "Interest Type" and "Preference Type" model interest and preference hierarchies as the ones suggested in (Gauch, Chaffee, & Pretschner, 2003) and (Maedche & Staab, 2002). Lastly, the classes "Living Thing" and "Non Living Thing" model real-world tangible objects. The class "Self" (highlighted in Figure 1), a direct subclass of "Person", models the profiled user.

In an ontology modeling the user domain, relations to other people, either personal or work-related, play a very important role. We used slot sub-classing to create a set of basic person relations slots, as shown in Figure 2, with the "acquaintance" slot sub-hierarchy. This can be extended or adapted according to the needs of each application.

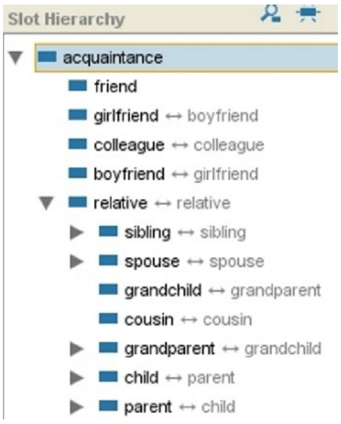


Figure 2 - "acquaintance" slot sub-hierarchy

The ontology has been modelled using the Protégé (Protégé, 2008) ontology management tool and the Protégé database project format. Protégé is a widely used open source ontology development 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 the algorithm in more detail.

4 Spreading activation through the personal ontology

Based on the personal ontology described in the pervious section, this section outlines a basic spreading activation framework in order to simulate the human memory mechanism and thus take advantage of the ontology to support PIM and TIM. The basic idea of the algorithm is, given an ontology class/instance that receives activation through an external event (its appearance in a document or e-mail for example), to provide a list of relevant ontology classes/instances to be used either for the classification of the specific document in a ontology-based PIM system or to support tasks in a TIM one.

The first section outlines the basic idea of the algorithm and its correspondence with the human memory model described in previous sections, followed by a simple example, whereas the next presents the algorithm in detail.

4.1 Different Timescales for Task-based Interaction

In the course of user interaction with a system, multiple timescales can be noted, which roughly correspond to the ones described in Table 1. 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 noted that some things (such as the user's own address) are more important than others (e.g. 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 recently encountered multiple times 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.

In order to address this issue, it seems appropriate to explicitly code the three different timescales in which the human brain operates using distinct activation levels 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 identified as "high relevance"), then the long-term activation grows. In addition, certain events (e.g. explicitly interacting with an entity) may be regarded on their own right 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 for simulating the aforementioned human memory model through spreading activation on the personal ontology.

4.2 Spreading Activation Algorithm

The spreading activation through the personal ontology algorithm described below assumes, to avoid repetition in the formulae, that the inverse of each relation is explicitly recorded in the ontology schema. For a real schema, as well as in our implementation, this means that all qualifiers would have to range over relations and their inverses. Also the weight (strength) of a relation is directional, allowing different 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 (note that according to the formal ontology definition given in section "Ontologies and Personal Information Management", the set of relations L is a subset of the set of properties P, including exactly those slots that model relationships) and a set of entities (instances) E. In terms of these sets, we may define the set of instances of relationships (statements) S:

$S = L \times E \times E$

Every statement is a particular relation between specific entities and set S includes all possible (as defined by set L) relationships between the entities of set E. A statement is thus a triple of the form (r, e1, e2), where r is a relation and e1 and e2 are instances; for readability purposes, statements will be denoted as r(e1,e2). In a real ontology only some of these statements will be valid with respect to the rules specifying which properties are valid for which classes of entities.

However, we assume that this validity checking is performed outside the spreading activation algorithm.

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

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

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

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

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 ontology entities (classes and instances) that are recognized 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, add them to 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 system supporting PIM/TIM. With the term "event" in this case we refer to a user action that has resulted in the identification of ontology entities (instances and classes) 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, we present in following sections in detail the algorithm steps.

4.3 Updating Short-Term Activation

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

$$IN(e) = \sum [LTW'(r) \times STA(e')],\ r \in L \wedge \exists\, e' \in E: r(e, e') \in OntologyState$$

The value of LTW'(r) is in fact the r relation's LTW [i.e. LTW(r)] value divided by the number of entities e' is related with through this relation, i.e. the "fan out" of the relation for entity e'. The formula for the STA then becomes:

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

The function f will typically count IA strongly, and only take into account MTA and LTA where either IA(e) or IN(e) is non-zero. For example, a possible function that we have used in our studies is:

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

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 (Mitchel, 1997) $S(sta) = \dfrac{1-e^{-sta}}{1+e^{-sta}}$ 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.

4.4 Spreading Activation Termination Conditions

The value of STA for each entity cannot be computed in a single step, because by nature the spreading activation algorithm is recursive. Consider for example the case that entity e_1 is connected to entity e_2, which is in turn connected to entity e_3. If e_1 receives incoming activation, some of it will be transferred to e_2; at this stage, however, some of e_2's activation should be transferred to e_3 and, to this end, a second iteration of the algorithm is needed. In the general case, *n*-iterations of the algorithm enable the propagation of the activation to entities connected to the initially activated ones via paths of length *n* or less.

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 entity. For example, an entity e1 may receive activation from a directly connected entity e2 during the first iteration step and from the entity e3 that is related to e1 through e4 during the second iteration step (e2→ e1 and e3→ e4 → e1. So excluding already visited entities has also been rejected. Another alternative would be to consider a state as "stable" (and thus terminate the algorithm) when all activation transfers are below a specified threshold th_{stable}. The threshold could be set as an absolute value (e.g. $S(sta) < 10^{-4}$) or as a ratio of the transferred activation divided by the current value of the receiving entity's STA (e.g. $(S(sta) / STA(e)) < 10^{-3}$), in both cases however experiments are needed to set the threshold to a reasonable value that guarantees efficiency and does not affect the accuracy of the algorithm.

As a result, for the needs of our implementation of spreading activation we opted for the constrained spreading activation, also suggested in (Lømo, 2003). We feel that this variation of the algorithm is closer to the human memory spreading activation process where the finite rate of firing and time 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. Furthermore, if the personal ontology is a 'small world' then it may be that activation can in principle spread across the network in very few iterations, before feedback loops become too powerful and start to affect significantly the STA weights. 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.

4.5 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 issue is the exact values of δ_{MTA} and δ_{LTA}. Furthermore, each entity's MTA and LTA values, apart from being incremented when active, they 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.

4.5.1 MTA Increase and Decay

A non-zero MTA value for an item expresses the fact that the item is recently and currently "active" and may be involved the user's current tasks. Higher MTA values imply more activations and/or activations of higher importance. Since any human's capacity for dealing with different subjects in a period of time is limited, the number of items having MTA > 0 should be also limited. Moreover, 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 performed 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 event, MTA could be increased directly.

4.5.2 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 they are not currently active or they may not have been active in the recent past, 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 = λ_{LTA} * (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.

4.6 LTW and Relation Weights

Relation weights are a very important issue in the spreading activation framework. Three levels of relation weights may be distinguished, which may be used as different options for regulating the spreading of activation between entities:

1. The relation as a whole, which is expressed by the relation's Long-Term Weight – LTW.

2. Weights on a particular instance of a relation, that is for a specific e1, e2 with a relation r between them, we could assign a weight dependent on:

   - Whether the relation was important in spreading activation

   - Whether both e1 and e2 have received high activation.

   In this case, the user could also specify a priori the weight of a particular relation. For instance, if there is a "friend" relationship, the user could assign higher weights to "better" friends.

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. Combinations of the 3 previously mentioned options could be envisaged, as, for example, option 1 and 2 could be combined. Working with weights on relation instances, however, remains an open issue that requires further research, as it is not yet clear what would the exact effects on activation spreading be.

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.

4.7 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.

Recall from section “Updating Short-Term Activation” that the formula for computing STA is

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

with A, B, C and D being constant values. At the moment before each new "event" STA is initialized to 0. A scale of [0, 100] is used for STA activation. With this scale, we decide that an IA entity should receive an activation of 80 and, as a result set A=80. The rest of the parameters B, C, and D may be set according to the needs of each application.

The MTA and LTA provide a non-linear factor in the STA computation function. In this case, they provide the factor multiplied to the sum of IA and IN.

For the needs of our implementation, we set C=0.5 and D=0.3 and use a scale [0, 20] for MTA and LTA. The increase step of MTA is set to 2, the STA threshold for MTA update is set to 50 and the maximum sum of MTA of all entities is set to 100. Finally, the number of steps T after which MTA decay is performed is set to 1 (we decay after every update). Regarding LTA, for the test implementation we used the user triggered events as a time unit, and have set the decay steps to 20.

LTW weights are initialized according to the cardinality of the relation (1-1, 1-m, etc). Appropriate weights are assigned in the range (0, 1], in order to simulate the fact that all or part of an entity activation spreads to a connected entity. LTW should be further fine tuned by the user or have various default values, according to the specific relation.

5 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 (Crestani, 1997b).

Crestani (1997a) 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, Weichselbraun, Scharl & Chang (2005) use spreading activation on a semantic network of automatically extracted concepts in order to identify suitable candidates for expanding a specific domain ontology. Xue, Zeng, Chen, Ma, Xi, Fan, & Yu (2004) 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 (2003) 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 (Hopfield, 1982) 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.

Recently, the spreading activation theory has been recognized as a candidate approach for supporting personal interaction with the system, in the newly emerging areas of personal information management (PIM) and Task Information Management (TIM). Katifori, Vassilakis & Dix (2008) discuss an approach for employing spreading activation to support management of personal information. The work presented in this paper extends the work of the previous paper, by refining activation processes, computing and tuning parameters related to how activation is spread among nodes, and addressing in detail the issue of initial value assignment for LTA, STA and MTA. These extensions have emerged through the process of evaluating the spreading activation methodology; this evaluation is another extension to the above mentioned work, and is described in the following section.

6 Preliminary evaluation

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 a 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 to evaluate the spreading activation algorithm in real conditions and fine tune its parameters, a fully functional PIMS is required. This system should provide effective mechanisms for user profiling, semantic storage of documents and context inference. Figure 3 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:

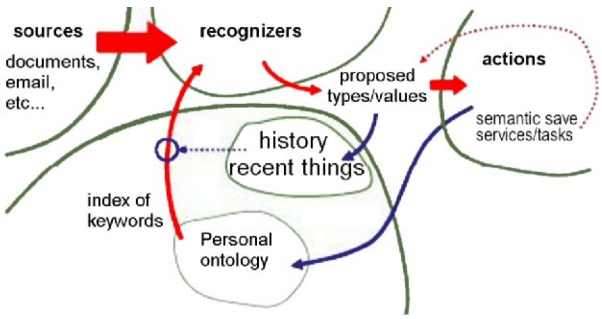


Figure 3 - Main components of a PIMS

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.).

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 3 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.

At the moment a prototype PIMS (Katifori, Poggi, Scannapieco, Catarci, & Ioannidis, 2005; Catarci, Dix, Katifori, Lepouras, & Poggi, 2007; Dix, Katifori, Poggi, Catarci, Ioannidis, Lepouras, & Mora, 2007) has been designed and is being developed, based mainly on web technologies. The first complete version of the prototype will allow the full evaluation of the spreading activation algorithm and allow us to 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 (Protégé, 2008).

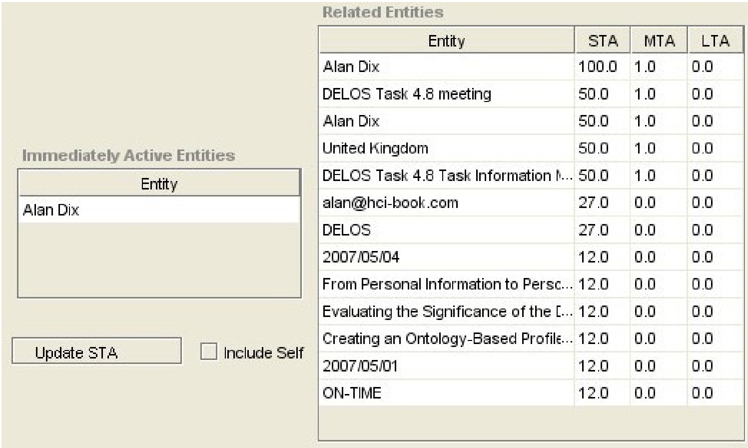
The following sections present the evaluation platform as well as the results of the preliminary evaluation.

6.1 The ActiveOnto Protégé Plug-in

The ActiveOnto plug-in (Katifori, Dix & Vassilakis, 2008) allows the initialization and setting of all the algorithm parameters and allows the user to simulate the functionality of the algorithm in a PIMS.

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). In a full-scale implementation, these entities could appear in a designated screen area (such as the Windows Vista sidebar (Microsoft Corporation, 2008)), allowing the user to select an entity and navigate to it, or perform actions on it through a context menu.

The plug-in may be found in (Katifori, Dix & Vassilakis, 2008) 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:



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 M...	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 Persc...	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 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.

1. All classes should conform to a meta-class having the slots IA, IN, STA, MTA, LTA and MAXLTA of type String. Note the spreading activation is allowed to include classes themselves through the *is-a* relation.

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 (Katifori, Vassilakis, Dix, Daradimos, & Lepouras, 2007) 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.

6.2 Evaluation Method and Results

As a first step of the evaluation, one of the authors populated her personal ontology with instances relevant to her work and computer-related activities in general for the past six months. These included colleagues and friends with whom she co-operated, papers submitted to journals and conferences, presentations prepared, conferences she attended, etc. This personal ontology (Katifori, Vassilakis, Dix, Daradimos, & Lepouras, 2007) during the evaluation included 90 classes, 211 instances and 196 slots types, of which about 60 were property ones and the rest relation type ones.

This preliminary evaluation would monitor changes in ontology weights while the user performed a series of tasks. These tasks included realistic usage scenarios compiled in cooperation with the user. Each task, for example "authoring a conference paper", was broken down to basic steps, like "the user is editing the paper", "the user e-mails the co-authors", "the user receives reply with attachment", etc. For each basic step, classes and instances that appear in the corresponding e-mails or documents were identified. Then, for each set of identified entities (classes and instances), corresponding to a basic step of the task, the entities were set as "selected" in the plug-in; afterwards, the "Update STA" button was pressed to invoke the STA, MTA and LTA computation algorithms. The user verified to what extent the entities identified as most relevant according to their STA score were in fact relevant and, also, to point out entities that did not receive a high STA score but are in fact relevant.

Several runs of the spreading activation algorithm were made, in order to fine-tune the algorithm parameters. Using the parameters of Table 2 to compute activation levels for 37 sub-tasks, the following results have been recorded:

- The mean percentage of the entities that were characterized as **relevant and useful** compared to the total number of entities with STA greater than 20 was **59%**

- The mean percentage of the entities that were characterized as **relevant but not useful** (this includes trivial classes like Person and E-mail) compared to the total number of entities with STA greater than 20 was **33.3%**

- The mean percentage of the entities that were characterized as **irrelevant** compared to the total number of entities with STA greater than 20 was **6.1%**

- In 14 of the sub-tasks, 1 entity identified by the user as *important* did not receive high STA score whereas in 4 sub-tasks 2 important entities did not receive a sufficiently high score. In the remaining 19 sub-tasks all important entities were correctly identified.

The following section presents an example of this process, highlighting the use and benefits of the personal ontology and spreading activation in a PIMS.

6.3 An example of the use of spreading activation in a PIMS

User Vivi is at the process of co-authoring a paper to submit to a PIM Workshop, along with researchers from other universities. The collaborative authoring is done through e-mail. We assume that there is already a first draft of the paper and in the ontology there is a "Conference Paper" instance of the paper entitled "Creating an ontology", linked to the five authors (Alan, Antonella, Vivi, Costas, Ilias) as well as an instance of "Conference", linked to the paper. For the portion of the authoring process described below, we will assume that only Vivi, Alan and Antonella take part in it.

We also assume that our PIMS has the appropriate recognizers to identify ontology classes and instances inside e-mails, Microsoft Word documents and web pages. As soon as a recognizer identifies a class or an instance within the scanned object, the spreading activation module is triggered and returns a set of entities (instances and classes) that are found to be of high relevance with the current document and task. These entities may be proposed to the user for the classification of the document (Katifori, Poggi, Scannapieco, Catarci, & Ioannidis, 2005) or used to infer the user context in order to support the current user task, like filling in a form automatically as described in (Dix, Katifori, Poggi, Catarci, Ioannidis, Lepouras, & Mora, 2007). For the purposes of the evaluation of the spreading activation algorithm, the triggering is emulated rather than driven by real external events as it is essential to understand the behavior of the algorithm before embedding it into a live system.

Instances of class "Person" (which actually correspond to real-world persons) are linked to the appropriate instances of the Values Classes "Person Name" and "E-mail". The recognizers identify person names and e-mails as instances of these two classes. In the evaluation example described below, the spreading activation algorithm has been tuned to not consider the "Self" instance, thus the "Self" instance does not appear in any result. Lastly, we assume that this is the first time the PIMS the spreading activation module is working, and consequently the STA, MTA and LTA weights in the personal ontology are all set to 0. In the following tables (tables 3-6) only the entities with STA above the threshold of 20 are presented in order to make the presentation more concise. A more detailed presentation of the following example may be found in (Katifori, 2008). The spreading activation parameters of Table 2 have been employed.

Table 2. Spreading Activation Algorithm Parameters

A	B	C	D	Iter.	STA Thres.	MTA Delta	MTA Thres	LTA delta
80	0.3	0.5	0.3	4	50	1	2	1

Step 1: Vivi opens the paper draft to edit it. The instances of *Person Name* "Alan", "Antonella", "Costas" and "Ilias", as well as the *Paper* instance "Creating an ontology" become activated.

Table 3. A test case of applying the spreading activation algorithm when the system has no prior memory

Entity	Entity Type	STA	MTA	LTA
Creating an ontology (conference paper)	Thing/Instance	98	0.94	0
Costas Vassilakis (person name)	Value Class/Instance	88	0.94	0
Alan Dix (person name)	Value Class/Instance	88	0.94	0
Ilias Daradimos (person name)	Value Class/Instance	88	0.94	0
Antonella Poggi (person name)	Value Class/Instance	88	0.94	0
Person Name	Value Class/Class	88	0.94	0
PIM Workshop (conference)	Thing/Instance	50	0.94	0
Costas Vassilakis (person)	Thing/Instance	27	0	0
Antonella Poggi (person)	Thing/Instance	27	0	0
Ilias Daradimos (person)	Thing/Instance	27	0	0
Alan Dix (person)	Thing/Instance	27	0	0
Conference paper	Thing/Class	27	0	0
Person	Thing/Class	27	0	0

Table 3 presents the STA, MTA and LTA weights after the first run of the algorithm. Column "Entity Type" indicates whether the related entity (listed in the first column) is a descendant of "Value Class" or "Thing" (recall from section 3.2 that the "Value Class" hierarchy contains essentially data items, while class the "Thing" hierarchy contains both abstract and tangible things, which may be objects, living organisms and concepts and effectively describes the universe of discourse) and whether it corresponds to a particular instance or a class. We may note that apart from the immediately activated instances, relevant entities like the "PIM Workshop" conference have received a high STA score. All were found relevant to the Microsoft

Word document containing the paper draft and would be chosen for its categorization by the user, except from “Person Name”, which is relevant but not useful.

Step 2: Vivi sends an e-mail having as subject “PIM Workshop”, to the two other authors that participate in this stage of authoring (Antonella and Alan). The instances “PIM Workshop” (Conference), alan@hci-book.com (E-mail) and antonella@hotmail.com (E-mail) become activated.

Table 4. Updating STA, MTA and LTA when the system has a memory of one task

Entity	Entity Type	STA	MTA	LTA
PIM Workshop (Conference)	Thing/Instance	100	1.68	0
alan@hci-book.com (E-mail)	Value Class/Instance	88	0.88	0
antonella@hotmail.com (E-mail)	Value Class/Instance	88	0.88	0
Creating an ontology (Conference paper)	Thing/Instance	73	1.68	0
E-mail	Value Class/Class	50	0.88	0
Antonella Poggi (Person)	Thing/Instance	50	0	0
Alan Dix (Person)	Thing/Instance	27	0	0
Conference	Thing/Class	27	0	0
5/4/2008 (Date)	Value Class/Instance	27	0	0
6/4/2008 (Date)	Value Class/Instance	27	0	0
15/6/2007 (Date)	Value Class/Instance	27	0	0

In this case, entities relevant to the e-mail such as the recipients’ addresses, the workshop, the workshop event dates and submission deadline and the paper itself receive high STA score. Note that other entities other entities may have received also activation through the spreading process, but the STA value they have amassed does not pass the threshold of 20 and are thus not included in table 4; one such example is the “DELOS Task 4.8 TIM” entity which is also relevant as the specific paper was created through collaboration in the context of this task, but its STA value is less than 20. The E-mail Value Class has been identified as relevant with a sufficiently high STA value too, but is not considered to be useful for interaction purposes.

After several similar e-mail exchanges and editing of the draft (steps 3-12 which are omitted here for brevity) Vivi receives the final confirmation from one of the authors to submit the paper.

Step 13: Vivi receives an e-mail having as subject “PIM Workshop”, from one of the two authors that participate at this stage of authoring, Antonella; the other author (Alan) appears in the recipient list of the e-mail. The instances “PIM Workshop” (Conference), alan@hci-book.com (E-mail) and antonella@hotmail.com (E-mail) become activated.

Table 5. Updating STA, MTA and LTA when the system has a memory of numerous tasks having the same focus

Entity	Entity Type	STA	MTA	LTA
alan@hci-book.com (E-mail)	Value Class/Instance	100	2.35	2
antonella@hotmail.com (E-mail)	Value Class/Instance	100	2.35	2
PIM Workshop (Conference)	Thing/Instance	100	2.77	8
Creating an ontology (Conference Paper)	Thing/Instance	100	2.77	8
Alan Dix (Person)	Thing/Instance	50	1.95	0
Conference Paper	Thing/Class	50	2.11	1
Person	Thing/Class	27	0.26	0
Antonella Poggi (Person)	Thing/Instance	27	0.26	0
5/4/2008 (Date)	Value Class/Instance	27	0	0
6/4/2008 (Date)	Value Class/Instance	27	0	0
15/6/2007 (Date)	Value Class/Instance	27	0	0

As before, all concepts are found relevant, except "Person", which seems redundant. We note that "PIM Workshop" and "Creating an ontology" have received high MTA and LTA scores, which is to be expected, as they have been the focus of this task. The paper co-authors, have received high MTA activation, but despite this they did not receive as high LTA one as the paper and workshop. This can be attributed to the fact that a considerable portion of the author's activation was withheld during the successive runs of the algorithm for each sub-task by two Value Class instances linked to the author, namely Person Name and E-mail. This suggests that Value Classes, necessary for the recognizer modules of the PIMS to work properly, should pass their whole activation through to the instances the values of which they represent. To this end, the LTW value of the relation from the Value Class instances to the corresponding Thing instances should be set to a high value. As a next step, the user visits the PIM Workshop website in order to submit the paper.

Step 14: In the submission web page, the user is presented with appropriate forms where she has to fill in, among other details, the paper title. As soon as the title is recognized and activated in the ontology, the paper authors are activated as well and their information may be automatically filled-in, in the appropriate forms. As a final step, the user receives an e-mail from one of the authors, Costas, asking if everything was OK with the paper submission.

Step 15: In the received e-mail the activated entities are the class "Paper", mentioned in the e-mail and Costas' E-mail and Person Name.

Table 6. The final state of STA, MTA and LTA in the experiment. As in this case few entities received activation above 20, those above 10 are presented also.

Entity	Entity Type	STA	MTA	LTA
Costas Vassilakis (person name)	Value Class/Instance	100	1.8	0
costas@di.uoa.gr (E-mail)	Value Class/Instance	100	1.92	0
Costas Vassilakis (person)	Thing/Instance	50	1.71	0
Paper	Thing/Class	73	0.76	0
DELOS Task 4.8 meeting (Meeting)	Thing/Instance	27	0.55	0
PIM Workshop (Conference)	Thing/Instance	27	1.15	10
Creating an ontology (Conference Paper)	Thing/Instance	12	1.15	10
Conference Paper	Thing/Class	12	1.01	2

As we may note, apart from "Costas" and "Paper", which are the immediately activated entities, "PIM Workshop" and "Creating an ontology" also receive an amount of STA, small but sufficient to include them in the activated concepts. DELOS Task 4.8 meeting is an irrelevant concept to this sub-task, having received an amount of activation from Costas.

7 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. By mimicking the way that the human brain recalls and activates concepts related to the current situation we envision to provide users of the PIMS with rapid access to activities that are most likely to be taken in the context of the task at hand. A prominent way to incorporate this feature in the user interface can be by listing related entities in a designated area and including applicable actions in context menus.

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 (Vassilakis, Lepouras & Katifori, 2007). The incorporation of these weights in the algorithm is still 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.

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 be possible, 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 updating and personalizing the ontology (Golemati, Katifori, Vassilakis, Lepouras & Halatsis, 2007) as well as visualization approaches (Katifori, Torou, Halatsis, Lepouras & Vassilakis 2006) are being investigated, taking additionally into account the relation weights.

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? On the one hand, including such concepts in the ontology makes the ontology a single, complete repository of all information needed to support the spreading activation framework. On the other hand, if users –especially novice ones– are presented with an ontology containing an overwhelming amount of information, they may easily be discouraged from using the system. A user interface that will be able to hide the ontology complexity from the user and present the level of detail that the user is able to handle can be a step towards alleviating such problems.

Scalability. 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? This issue includes considerations on amount of processing that must take place, memory and disk requirements for storing the personal ontology, as well as system effectiveness, since (a) the related concepts must be almost instantly presented to the user (if presentation is delayed, the user may have switched task in the meantime thus the presented entities will be out-of-context) and (b) the incorporation of the spreading activation framework in the system must not hinder the normal flow of operations.

Tests on the algorithm are being continued with different sets of parameters, while using different ontologies and users is considered. A testing platform is being created that will allow the use of the spreading activation module for context inference in a more realistic environment where the user is evaluating the algorithm when actually performing the tasks. The first version of this environment will focus on selected applications, including word processing and e-mail management, by building appropriate functionality into specific applications. More generic mechanisms that will employ operating-system level mechanisms so as to seamlessly support all applications will be investigated at a later stage.

The first results of our evaluation are promising, since the spreading activation algorithm recognizes with ample accuracy the entities related to the current user's activity, as the latter are derived by the entities receiving immediate activation. It is expected that the incorporation of additional features in the algorithm, including the weights on relation instances, dynamic LTW relation weights and automatic tuning of spreading activation parameters will further increase the algorithm's accuracy. Fine tuning of the spreading activation algorithm parameters and adaptation to individual user profile is also expected to contribute to the same effect. Finally, the development of user-friendly and efficient mechanisms for personal ontology population and user interaction with the entities identified as relevant will allow the system to be used in everyday tasks by non-expert users.

References

Amann, B. & Fundulaki, I., (1999). Integrating Ontologies and Thesauri to Build RDF Schemas, *Proceedings of the Third European Conference on Research and Advanced Technology for Digital Libraries*, pp. 234 – 253

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

Anderson, J. R. (1995), *Cognitive Psychology and its Implications*, Worth Publishers, 5th Edition.

Atkinson, R.C. & 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.

Catarci, T., Dix, A., Katifori, A., Lepouras, G. & Poggi A., (2007). Task-Centered Information Management. *Proceedings of the DELOS Conference 2007 on Working Notes*, 13-14 February 2007, Tirrenia, Pisa (Italy), C. Thanos and F. Borri (eds.). pp. 253-263.

Catarci, T., Habegger, B., Poggi, A., Dix, A., Ioannidis, Y., Katifori, A. & Lepouras, G., (2006). Intelligent user task oriented systems. In *Proc. of the Second SIGIR Workshop on Personal Information Management* (PIM 2006), 2006.

Chirita, P. A., Gavriloai, R., Ghita, S., Nejdl, W. & Paiu, R. (2005). Activity Based Metadata for Semantic Desktop Search, *Proceedings of the 2nd European Semantic Web Conference*.

Corcho, O. & Gómez – Pérez, A. (2000). A roadmap to ontology specification languages, *Proceedings of the 12th International Conference on Knowledge Aquisition, Modelling and Management* (EKAW), 2000, Springer

Crestani F. (1997b). Application of spreading activation techniques in information retrieval. *Artificial Intelligence Review*, 11(6):453–482, December 1997.

Crestani, F. (1997a). 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*, pp. 1210-1214, Aachen, Germany.

Dix, A. (2006). The brain and the web - intelligent interactions from the desktop to the world. *Keynote at Simpósio de Fatores Humanos em Sistemas Computacionais (IHC 2006),* Natal Brazil, 19-22 Nov. 2006. Retrieved from http://www.hcibook.com/alan/papers/brazil2006/

Dix, A., Katifori, A., Poggi, A., Catarci, T., Ioannidis, Y., Lepouras, G. & Mora, M., (2007). From Information to Interaction: in Pursuit of Task-centred Information Management. *Proceedings of the DELOS Conference* 2007, December 2007, Tirrenia, Pisa (Italy).

Ericsson, K. A., & Kintsch, W. (1995). Long-term working memory. Psychological Review, 102, 211-245.

Facebook, (2008). *Facebook home page*. Retrieved from http://www.facebook.com/, February 13, 2008.

Gauch, S., Chaffee, J. & Pretschner, A., (2003). 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.*

Gauch, S., Chaffee, J. & Pretschner, A., (2003). 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.*

Gazzaniga, M. S., Ivry, R. B. & Mangun, G. R., (1998) Cognitive Neuroscience, The Biology of the Mind, pp 289-294, W. W. Norton & Company, New Ed edition (April 1998)

Golemati, M., Katifori, A., Vassilakis, C., Lepouras, G. & Halatsis, C., (2007). Creating an Ontology for the User Profile: Method and Applications, *Proceedings of the First RCIS Conference*, April 23-26, 2007, Ouarzazate, Morocco.

Gruber, T. R. (1993a). Toward principles for the design of ontologies used for knowledge sharing. Originally in N. Guarino and R. Poli, (Eds.), *International Workshop on Formal Ontology*, Padova, Italy, 1993. Revised August 1993. Published in *International Journal of Human-Computer Studies*, Volume 43 , Issue 5-6 Nov./Dec. 1995, pp. 907-928

Gruber, T. R., (1993b). A Translation Approach to Portable Ontology Specifications, *Knowledge Acquisition, Special issue: Current issues in knowledge modelling*, Vol 5, Issue 2, pp. 199-220.

Hasan, M. (2003). A Spreading Activation Framework for Ontology-enhanced Adaptive Information Access within Organisations. *In Proceedings of the Spring Symposium on Agent Mediated Knowledge Management AMKM 2003*, Stanford University, California, USA

Hopfield, J. (1982). Neural networks and physical systems with emergent collective computational properties. *Proceedings of the National Academy of Sciences of the USA*, 79:2554 - 2588.

ICQ (2008). *ICQ home page*. Retrieved from http://www.icq.com, February 12, 2008

Katifori, A., (2008). *Preliminary Evaluation Example of the Spreading Activation Algorithm*. Pened technical report TR-08-01. Retrieved from http://oceanis.mm.di.uoa.gr/pened/?c=pub#tr, February 21, 2008.

Katifori, A., Dix, A. & Vassilakis, C., (2008), *Spreading activation Protégé plugin*. Retrieved from http://oceanis.mm.di.uoa.gr/pened/?c=pub#plugins, February 02, 2008.

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

Katifori, A., Vassilakis, C., Dix, A., Daradimos, I. & Lepouras, G., (2007). *Spreading Activation User profile ontology*. Retrieved from http://oceanis.mm.di.uoa.gr/pened/index.php?category=publications#ontos, February 2, 2008.

Katifori, A., Vassilakis, C. & Dix, A., (2008). Using Spreading Activation through Ontologies to Support Personal Information Management. *Proceedings of Common Sense Knowledge and Goal-Oriented Interfaces,* held in conjunction with the 2008 International Conference on Intelligent User Interfaces (IUI 2008).

Katifori, V., Poggi, A., Scannapieco, M., Catarci, T. & Ioannidis, Y., (2005). OntoPIM: how to rely on a personal ontology for Personal Information Management, *Proc. of the 1st Workshop on The Semantic Desktop*, 2005.

Kobsa, A. (1993). User Modelling: Recent work, prospects and hazards, in *Adaptive User Interfaces: Principles and Practices*, edited by Schneider-Hufschmidt, T. Khme and U. Malinowski, North Holland.

Lepouras, G., Dix, A., Katifori, A., Catarci, T., Habegger, B., Poggi, A. & Ioannidis, Y., (2006). OntoPIM: From Personal Information Management to Task Information Management, *Proceedings of Personal Information Management, SIGIR 2006 workshop*, August 10-11, 2006, Seattle, Washington.

Liu, W., Weichselbraun, A., Scharl, A. & Chang, E., (2005). Semi-Automatic Ontology Extension Using Spreading Activation, *Journal of Universal Knowledge Management*, no. 1 (2005), pp. 50 - 58

Lømo T. (2003). The discovery of long-term potentiation. *Philosophical Transactions: Biological Sciences*. 358 (1432): 617-20. PMID 12740104.

Lorch, R. F., (1982). Priming and search processes in semantic memory: A test of tree models of spreading activation. *Journal of Verbal Learning and Verbal Behavior*, 8, pp. 240-248

Maedche, A. & Staab, S. (2002). Mining Ontologies from Text, *Proceedings of EKAW 2000*, pp. 189-202, 2002.

Microsoft Corporation, (2008). *Windows Vista: Features Explained: Windows Sidebar and Gadgets*. Retrieved from http://www.microsoft.com/windows/products/windowsvista/features/details/sidebargadgets.mspx February 13, 2008.

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

Miller, G., A. (1956). The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information, *The Psychological Review*, vol. 63, pp. 81-97

Mitchell T. M., (1997). *Machine Learning*, McGraw-Hill, 1997, ISBN 0-07-042807-7.

MySpace, (2008). *MySpace home page*. Retrieved from http://www.myspace.com/, February 12, 2008.

Noy, N. F. & McGuiness, D. L., (2001) *Ontology Development 101: A Guide to Creating Your First Ontology*, Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, 2001

Peterson, L.R., & Peterson, M.J. (1959). Short-term retention of individual verbal items. *Journal of Experimental Psychology*, 58, 193-198.

Protégé, (2008). *The Protégé Ontology Editor and Knowledge Acquisition System*. Retrieved from http://protege.stanford.edu/, February 09, 2008.

Reder, L. M. & Anderson J. R. (1980). A partial solution of the paradox of interference: The role of integrating knowledge. *Cognitive Psychology*, 12, 447-472

Renato, I., (2001). *Representing vCard Objects in RDF/XML*. Retrieved from http://www.w3.org/TR/vcard-rdf, February 10, 2008.

Sauermann, L. (2005). The Gnowsis Semantic Desktop for Information Integration, *Proceedings of the 3rd Conference Professional Knowledge Management* (WM 2005)

Sharifian, F. & Samani, R. (1997), Hierarchical Spreading of activation, *Proc. of the Conference on Language, Cognition and Interpretation*, Isfahan, pp. 1-10

Tazari, M. R., Grimm, M. & Finke, M. (2003). Modeling User Context, *Proceedings of the 10th International Conference on Human-Computer Interaction* (HCII2003), Crete, Greece, June 2003

Trajkova, J. & Gauch, S. (2004). Improving Ontology-based User Profiles, *Proceedings of RIAO 2004*, University of Avignon (Vaucluse), France, April 26-28, 2004, pp. 380-389

Vassilakis, C., Lepouras, G. & Katifori, A., (2007). *wt-Protégé – An Extension for Protégé Supporting Temporal and Weighted Data*, Technical Report TR-SSDBL-07-001, September 2007, Tripoli, Greece

Xue, G. R., Zeng, H. J., Chen, Z., Ma, W. Y., Xi, W., Fan, W. & Yu, Y. (2004). Optimizing Web Search Using Web Click-through Data. *Proceedings of CIKM 2004*: 118-126