Master's thesis

Identification of relevant Concepts in an Ontology-based Context Model

Georg Kompacher

Knowledge Management Institute

Graz University of Technology

Chair: Univ.-Prof. Dr.rer.nat. Klaus Tochtermann

Assessor: Univ.-Doz. Dr. Stefanie Lindstaedt

Graz, May 19, 2011

Abstract

Managing the massive amount of continously increasing personal information is becoming ever more difficult. Unobstrusive observation of the user in order to understanding the present user context can enable better support for the knowledge worker.

The goal of this thesis is to identify entities of a user interaction context model relevant to a user's current task. A spreading activation approach is applied on the graph structure of a user interaction context model in order to discover relevant tasks of the same and other users based on the current user's task. The user interaction context model, the realized ontology and the automatic population mechanisms have been developed by Andreas Rath as part of his research work [Rath, 2010]. The objectives of this thesis are (a) the identification of relevant tasks in a user interaction context model, (b) the determination of concepts and properties of the user interaction context ontology for the spreading activation approach, (c) the evaluation of required number of iterations as well as (d) the evaluation of a good results showing combination of the activation decay, the threshold level, and the relation weights used for the spreading activation algorithm, and (e) the visualization of the resulting spreading activation graph based on the user interaction context graph.

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

19.05.2011	Georg Kompacher
....	....
date	(signature)

Contents

1 Introduction

"We have vast amounts of information to which accurate and speedy access is becoming ever more difficult." [Rijsbergen, 1979] It is very amazing that more than two decades later this statement is more actual than ever. The massive amount of continuously increasing information makes it a great challenge to manage it. And sometimes it can get even worse, especially if a document cannot be found any more even if it was definitely saved previously. "Ironically, in quite a few of these cases nowadays, the document we are looking for can be found faster on the World Wide Web than on our personal computer." [Chirita et al., 2006] The focus in the research area of information retrieval was clearly directed on the World Wide Web. Due to the lack of good search tools for desktop search, users have spent more and more time into managing and organizing their data than into performing their tasks where that data is used for [Katifori et al., 2008]. Currently consumer operating systems provide integrated information retrieval, e.g. Apple's Mac OS X with Spotlight [Inc., 2010], Microsoft's Windows with Windows Search [Corporation, 2010]. However compared to modern web search engines the results supplied by desktop search engines are very sobering. The goal is to increase the retrieval performance to close the present gap between desktop search and web search engines. Major factors for the poor performance of current desktop search engines are on the one hand the fact that from the point of view of an information retrieval system documents and other data are disconnected from each other. On the other hand the lack of appropriate ranking algorithms.

Summing up it is desirable to increase the retrieval performance. One approach is to gather context-aware information. This approach is the starting point for this thesis. User activities serve as a basis for additional context aware information. User activities on different resources are able to serve as nescessary links between user actions and documents. All those user actions and documents as well as the links itself are part of a user interaction context model. The population of such a user interaction context model is accomplished by observing the user interacting with resources, applications and the operating system. By the use of a populated context ontology conclusions can be drawn by applying different context analysis technologies. One approach, namely a spreading activation algorithm, is utilized in this thesis.

1.1 Motivation

Many today's personal information management systems have the user context out of scope. Computer users spend a lot of time on organizing and managing their large amounts of data because otherwise the retrieval of information at a later time would be difficult. Instead, users should focus more on performing their tasks than on managing their information [Katifori et al., 2010].

A major problem traditional information retrival systems have in common is that they operate without considering the context of a request. Speaking casually, it has to be accertained in what information the user is interested in. In most cases the identification of the user's information need is the key to success. To support the user as much as possible it has to be found out on which task the user is currently working on. This means that the user can be supported with relevant information belonging to the task. Relevant information in this case include work and learning resources. Currently, there are many different and highly promising approaches to overcome this problem. One possible approach is to observe the users interacting with their applications by using context sensors respectively application adapters [Budzik and Hammond, 2000, Budzik et al., 2001].

1.2 Goal of this research

The goal of this thesis is to identify tasks of a user interaction context based graph relevant to a user's current task in order to enable better work and learn support. The graph is based on an user interaction context model which is realized as a user interaction context ontology. The user interaction context ontology is a represention model of the user interaction context. The ontology is interpreted as a user interaction context graph and therefore a spreading activation approach can be applied to discover relevant tasks of the same and other users based on the current user's task. The user interaction context is defined as "all interactions of the user with resources, applications and the operating system" [Rath2010].

As noted above, the identification of relevant tasks in a user interaction context model is the overall goal of this thesis. The purpose of the identification of relevant tasks is to offer support to the user while performing a task. In this thesis the term relevant task refers to a task based on the same task model, which is some kind of categorization concept. For this reason, if a task belongs to the same task model as the current task of the user, it is considered to be relevant. In the present user interaction context model used resources are interlinked with the task they have been used in. That implies that if it is possible to enable the identification of relevant tasks, in future work appropriate resources for the current situation can be offered. In order to identify relevant tasks a user interaction context model is utilized to build a associative network respectively graph. This network structure is a subpart of the user interaction context model. Spreading activation is applied as a search paradigm on the associative network.

1.2.1 Determination of concepts and properties

In order to identify relevant tasks, the spreading activation graph is set up by determination of concepts and properties of the user interaction context ontology. This kind of determination can be considered as a filter pattern for the user interaction context ontology, i.e. the spreading activation graph includes only instances of the filtered user interaction context ontology. The size and the structure of the graph used for spreading activation depend on that selection. The size and the structure of the graph are important factors for the achievement of the main goal of this thesis, namely the identification of relevant tasks in a user interaction context model.

1.2.2 Required number of iterations

Another subgoal of this thesis is the evaluation of the required number of iterations, i.e. how many spreading activation iterations are necessary to identify relevant tasks in a user interaction context model. Spreading activation is an iterative searching method for associative respectively semantic networks and the number of iterations has to be set in advance. The quality of the result (relevant tasks) diminishes, when a certain number of iterations are reached. Therefore, the identification of relevant tasks, and more precisely, its performance, is influenced by the number of iterations.

1.2.3 Spreading activation specific attributes

Besides the structure which is affected by the determination of concepts and properties the evaluation of spreading activation specific attributes such as activation decay, threshold level, and relation weights are evaluated in order to retrieve relevant tasks. By the assignment of these attributes the control of activation propagation inside the spreading activation graph is enabled. Since these attributes depend on the structure of the graph as well as on the specific use case there are no standardized values available. Therefore the values of these attributes are empirically determined.

1.2.4 Visualization

The resulting spreading activation graph is being visualized in order to provide the domain expert, e.g. the programmer, a visual verification method for the spreading activation algorithm and more important to give the user an idea how the results are achieved by the algorithm. Moreover the visualization of the spreading activation graph gives the user the opportunity to discover the coherences of the user interactions represented by the graph.

1.2.5 Summary

Summarizing the objectives of this thesis are (a) the identification of relevant tasks in a user interaction context model, (b) the determination of concepts and properties of the user interaction context ontology for the spreading activation approach, (c) the evaluation of required number of iterations as well as (d) the evaluation of a good results showing combination of the activation decay, the threshold level, and the relation weights used for the spreading activation algorithm, and (e) the visualization of the resulting spreading activation graph based on the user interaction context graph.

1.3 Background of this work

The KnowSe service framework was the basis for the development of the mentioned task identification approach based on user interaction context data. "KnowSe forms the basis for dynamically orchestrating a large variety of intelligent knowledge services, highly contextualized to a persons work context and interconnected with a multitude of knowledge sources" [Know-Center2010].

One feature of the KnowSe service framework is the observation of the user interaction context and the generation of a user interaction context model. This model consists of all relations between user interactions, resources and applications and serves as a basis for the identification of relevant tasks. The user interaction context model is realized as an user interaction context ontology (UICO). The UICO is a represention model of the user interaction context. It models relations between the user with resources and applications. The user interaction context model, the realized ontology and the automatic population mechanisms have been developed by Andreas Rath as part of his research work. The instantiation is accomplished by observation of the user interaction context with context sensors and automatic population of the ontology model by utilizing static rules, information extraction and machine learning techniques [Rath2010]. The user interaction context ontology and the described context sensors are not part of this thesis. Further information about the context ontology, the context sensors and the population mechanisms can be obtained from [Rath2010].

The basis for the identification of relevant tasks is the task execution data gathered during three different laboratory experiments designed by Andreas Rath. Populated user interaction context ontologies are the outcome of these laboratory experiments. The populated user interaction context models can be interpreted as user interaction context graphs. On the basis of these populated user interaction context graphs relevant tasks based on a single user's task are identified.

1.4 Approach

A spreading activation approach was applied to the task identification system. In principle spreading activation is a search method for network structures. These network structures can be associative, neural, or semantic networks. In this thesis the network structure is generated from an existing instantiated user interaction context ontology. This data model, which represents the user's context, is described by means of the Resource Description Framework (RDF). The RDF model is considered as a labeled, directed graph, and therefore a graph traversal algorithm such as spreading activation can be applied. This work follows the assumption that associations between two tasks indicate a possible relevance of the related task with regard to the originating task. Every user interaction context can be seen as a cloud of nodes and edges. At some areas those clouds overlap each other, i.e. there is either a shared concept instance (e.g. a resource) or some kind of relationship between two concept instances and therefore an edge refers to this relation. Therefore, relevant concepts, i.e. tasks inside the graph representation of a user interaction context model should be identified by the utilization of already existing associations between concepts. The reason for the determination of relevant respectively similar concepts is the presumption that they have the ability to support the knowledge worker.

1.5 Achieved Results

In order to measure the performance of the approach, the system has been evaluated with the three datasets as mentioned earlier. With these datasets an offline analysis of the user interaction context data becomes possible. The first insight was that there is a dependency between the size of the user interaction context model graph and the execution time of the spreading activation algorithm. For this reason it was decided to disregard specific node types in order to reduce the size of the graph as well as the complexity. The task identification system has been evaluated by the application of different setups concerning the spreading activation specific attributes such as activation decay, threshold level and relation weights. The results achieved with the different setups show a recall up to 0.74 depending on the used dataset. The calculated precision results show values up to 0.32. The results have shown that the performance of the retrieved data depends on the type of the task. Furthermore, it could be shown that the structure of a user interaction context model enables the identification of certain entities. Since the achieved results vary significantly from each other it cannot be generalised that the identification approach is applicable on other domains. In future work it remains to be examined whether the user interaction context model is sufficient or the configuration of the associative network has to be improved. At the end of this thesis various approaches are discussed to improve the performance of the retrieval system.

1.6 Thesis Structure

The remainder of this thesis is structured as follows: Chapter 2 will discuss the basis of this work concerning the user context, the user interaction context ontology as well as concepts and ontologies in general. Chapter 3 provides some detailed information about associative and semantic networks and the application of a spreading activation algorithm on such a structure. Chapter 4 gives an overview over present research work in the field of user context utilization and spreading activation. Chapter 5 provides insights into the concept and design of the associative network. The iterative extraction of information from the context ontology to establish the network structure is explained in detail. Furthermore the application of a spreading activation algrithm in combination with appropriate constraints is discussed in more detail. Moreover, the visualization of the spreading activation approach is outlined. In chapter 6 the architecture and the implementation of the developed model are pointed out. Additionally the developed graphical user interfaces are presented. Then, in chapter 7 the developed model is evaluated with two datasets collected during experiments with researchers and computer science students. By means of these datasets it became possible to test the spreading activation algorithm with different configuration variants. Finally, this thesis is closed in chapter 8 by giving some conclusions. Some possible future directions are also discussed.

2 Basis for this work

This chapter describes the starting point of this thesis. Since the identification of concepts is based on a user context model the term user context will be described here in more detail. It will also be stated out which definition of user context is used in this thesis. Afterwards we take a closer look at the User Interaction Context Ontology which is utilized as a datamodel for the task identification approach described in this thesis. Finally the concept term in the domain of ontologies will be explained.

2.1 User Context

Since the term context is used in many different domains it has be determined what context is about and how it is used in this thesis. As an introduction to the context topic we investigate a conversation between humans: When humans are talking with each other, they are able to use implicit situational information, the context, in order to understand his conversational partner as good as possible. If we substitute one partner with a computer it may be difficult to convey ideas to the computer. The problem is that computers nowadays lack of the ability to take the context of the dialog between humans and computers into account. "By improving the computers access to context, we increase the richness of communication in human-computer interaction and make it possible to produce more useful computational services." [Dey, 2001]

There is a multitude of different definitions of context. The least meaningful definitions are those which substitute the term context by synonyms. These definitions put context on a level with the term environment respectively situation [Schilit and Theimer, 1994, Ward et al., 1997, Salber et al., 1999]. These synonyms are "extremely difficult to apply in practice" [Dey, 2001]. When talking about the context in this thesis we refer to the definition of [Dey, 2001]: "Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves."

2.2 User Interaction Context Ontology

The User Interaction Context Ontology (UICO) used in this thesis serves as the knowledge base for the user context data. [Rath, 2010] defines the user interaction context as

> "all interactions of the user with resources, applications and the operating system on the computer desktop. Resources are digital artifacts on the computer desktop, e.g., documents, web pages, emails, persons, appointments and notes."

The ontology-based context model is build on the basis of the semantic pyramid as a conceptual model (further described in Section 2.2.1). The aim of the user interaction context ontology is to depict fine grained contextual information. To receive the contextual information a set of context sensors are utilized which capture and forward contextual information to a processing module which for his part transforms the received contextual information data into event objects. The user interaction context ontology describes all sensed information about the user interaction context and reflects interconnections between different types of information. These information types include concepts of the semantic pyramid as well as sensed resource data and metadata.The ontology is automatically populated during user's interactions with the computer. To go more into detail it is worth knowing that the ontology was modeled manually using the Protégé ontology modeling tool [Stanford Center for Biomedical Informatics Research, 2010]. The ontology consists of 107 concepts and 281 properties. The properties are subdevided as follows: 224 datatype properties and 57 objecttype properties. The automatic population of the ontology merely refers to the instantiation of the modeled concepts and the relations between them. The already-mentioned context sensors observe the user interaction context. Different types of context sensors have been utilized:

- File System Sensor
- Clipboard Sensor
- Sensor for Microsoft Office
- Sensor for Microsoft Internet Explorer
- Sensor for Mozilla Thunderbird
- Sensor for Mozilla Firefox
- Sensor for Novell GroupWise

After some smaller intermediate steps the process of semantic representation of the user interaction context is completed.

2.2.1 The Semantic Pyramid

The conceptual model of the UICO is referred as the semantic pyramid [Rath et al., 2006]. Figure 2.1 illustrates the composition of the semantic pyramid with all it's components. Basically, the semantic pyramid is a three layer model. The semantic pyramid is structerd such as unlying items are aggregated by items in the overlying layer.

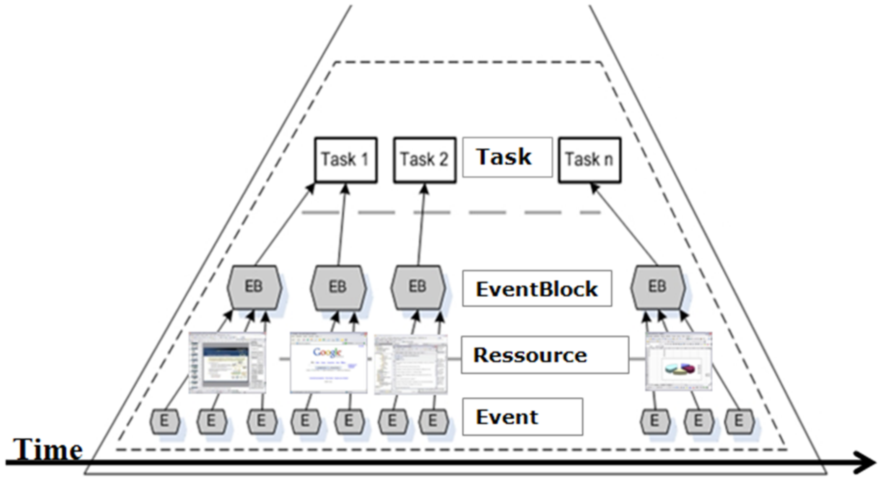

Figure 2.1: The semantic pyramid comprises the event, the event block and the task layer.

At the bottom layer are events which are the representation of different user interactions with several applications respectively the operating system. In addition reactions to user interactions are also represented by events. Examples for events are keyboard strokes, mouse movements, mouse clicks, copy & paste actions, starting a program, creating a folder, creating a folder, a web search, or opening a file.

The aggregation of a set of events forms an event Block. To be part of an event block all events must have one thing in commom: They must act on the same resource. Hence, multiple consecutive interactions (also referred as event sequences) with the same resource are grouped into an event block.

Reaching the third layer of the semantic pyramid similar event blocks are grouped together to tasks. Accordingly, a single task of a knowledge worker represents a semantic set of event blocks. It must be noted that only a single knowledge worker is involved into a task, which, by the way, cannot be subdivided into subtasks. [Rath et al., 2006, Rath et al., 2009]

2.3 Concept and Ontology

Since this thesis is about the identification of relevant concepts it has to be stated what is meant by the term conceptualization:

> "A body of formally represented knowledge is based on a conceptualization: the objects, concepts, and other entities that are assumed to exist in some area of interest and the relationships that hold among them (Genesereth & Nilsson, 1987). A conceptualization is an abstract, simplified view of the world that we wish to represent for some purpose. Every knowledge base, knowledge-based system, or knowledge-level agent is committed to some conceptualization, explicitly or implicitly." [Gruber, 1995]

An abstract, simplified view of the world in this case means, that only those entities that are relevant for a particular application will be adopted by the model as concepts and as relationships among them.

"An ontology is an explicit specification of a conceptualization. The term is borrowed from philosophy, where an Ontology is a systematic account of Existence" [Gruber, 1995]. This is the most commonly used definition of the term ontology in the field of knowledge based systems.

Important components of ontologies used in this thesis are the following:

- Individuals: instances or objects
- Classes: concepts
- Attributes: properties, characteristics

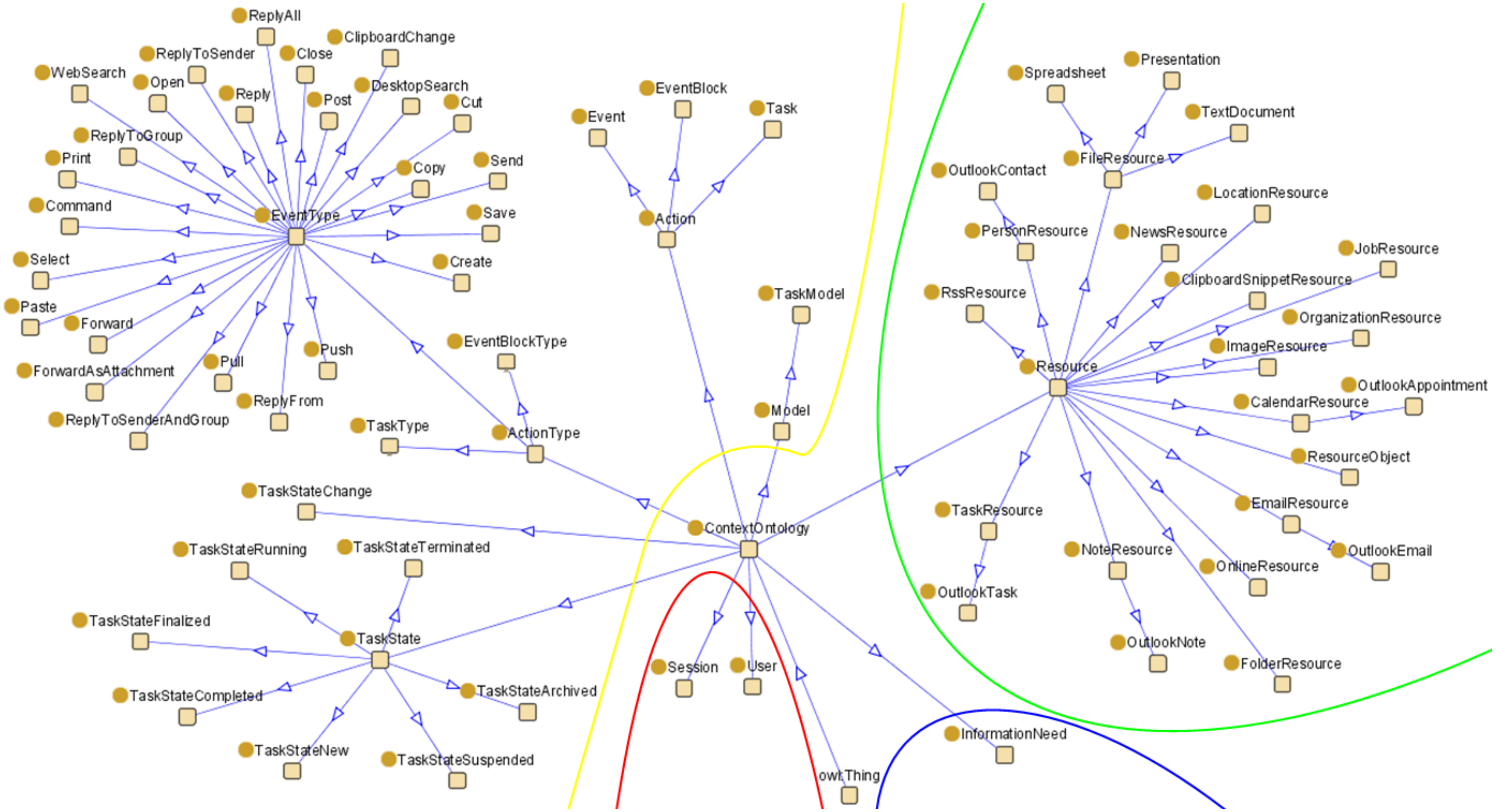

Figure 2.2: The concepts of the User Interaction Context Ontology (UICO). Of particular interest for this thesis are the left area shwoing the action dimension and the right area showing the resource dimension. [Rath, 2010]

3 Information Retrieval and Spreading Activation

The way in which modern retrieval systems are used today has changed significantly in recent years. Due to the interconnectivity of personal computers through the internet the possibilities of using information retrieval systems have increased considerably. Those systems are available to internet users every day by accessing them via their own personal computer. The used computer systems are the same on which the users write their documents, read their emails, browse websites of interest, and manage their contacts and addresses. These data collections can be seen as very large personal repositories storing a huge amount of information. Today's computer users spend most of their time organizing and managing their repositories because otherwise the retrieval of information at a later time would be difficult. Users should focus on performing their tasks instead of managing their information [Katifori et al., 2010]. Unfortunately these personal repositories are unused by most of today's information retrieval systems. Users' information needs are still communicated in terms of queries to different systems that are isolated from knowing the context of that need [Budzik et al., 2001].

This chapter gives an overview of the main topic areas which underlie this thesis. Due to the use of a network structure as a knowledge representation the characteristics of associative and semantic networks are presented. Finally the spreading activation algorithm applied as an associative processing paradigm is described in detail.

3.1 Introduction

The impact of information retrieval regarding to the acquisition of information has increased in the recent years. Resources are spread over the WWW but also over personal storage devices (HDDs, USB-Sticks, etc.) or other physical locations. Unless this tacit knowledge is not made explicit it is useless to the user. By the use of information retrieval systems documents can be found and information needs can be satisfied.

3.2 Information Retrieval

Information retrieval is about the storing but more about the retrieving and the fast access of the stored information. The type of information can be manifold, e.g. text documents, images, or videos. Most of today's information retrieval systems deal merely with textual information. A mayor problem is to do effective retrieval on large collections. Information retrieval systems store the different types of information to be retrieved for future use. Many of today's systems only offer the storage and the retrieval of textual information.

The necessity of an information retrieval system is substantiated in an upcoming information need of the user. The information need in question has to get satisfied by the information retrieval system. Therefore the user has to express his information need in form of a query. This query is then submitted to the information retrieval system. The response to the query consists of a list of references to e.g. text documents. The list is ordered by relevance to the user's query as the information retrieval system determines.

At the beginning the first information retrieval systems were boolean systems. Users of those information retrieval systems had to transform the information needs into boolean expressions. These boolean information retrieval systems lack of a feature to rank the documents according to importance. Basically the main problem is that most information retrieval system have the context out of focus, that means that those systems 'are isolated from the context in which a request occurs' [Budzik and Hammond, 2000]. Focussing on this problem it seems to be necessary to take the user context into account. The information need of the users is associated with a context. This context is not considered by most of todays information retrieval systems.

One form of information retrieval is associative retrieval. When talking about associative retrieval the relation between two or more concepts are taken into account. In other words information, e.g. text documents, is associated with other information wich is known to be relevant for the user.

3.3 Associative and Semantic Networks

[Quillian, 1968] is often referenced as he introduced the basic ideas and principles behind semantic networks. The intent was to find a representational model that describes how semantic information is organized in the human brain. A further aim was the development of a representation that facilitates the readability for humans. He describes a memory model consisting of nodes networked together by associative links. The nodes should represent information items also known as concepts. Furthermore the relationship between various concepts are expressed through hierarchical connections via "is-a" and "instance-of" links [Crestani, 1997, Berger, 2003]. The degree of abstraction increases as the concepts move up the hierarchy of "is-a" relations. Various properties are also assigned to concepts. Hence the properties are represented by nodes connected to concept nodes via labeled links.

A distinction is made between semantic networks and associative networks. As described in [O'Riordan and Sorensen, 1995] semantic networks have different generic link types: synonymy, supercall-subclass, and disjunctive and conjunctive sets of links. On the other hand associative networks utilize only a single link type. This link type however has a weighted edge so that the structure of the network and the associated parameters depict the semantics implicitly.

3.4 Spreading Activation

"Spreading activation has proved to be a model with a high degree of explanatory power in cognitive psychology." [Sharifian and Samani, 1997] The main benefit of that model is that it captures both the presentation of knowledge and the way it is processed. The human memory has the ability to encode the meaning of concepts by their associations with other concepts [Anderson and Bower, 1979, Anderson, 1983, Quillian, 1968]. Vannevar Bush put it this way: "It operates by association. With one item in its grasp, it snaps instantly to the next that is suggested by the association of thoughts"[Bush and Wang, 1945]. An ontology can be seen as a possible approach to model these interconnected entities by classifying concepts in more generic and more specific ones [Sharifian and Samani, 1997].

Spreading Activation has its roots in the cognitive psychology. There it is used as an explanatory model to understand the operation method of the human brain. In the cognitive psychology the idea is about neurons and synapses which interconnect the neurons. The basic idea behind the spreading activation approach is that a neuron's activation spreads to out to the other units connected to that neuron [Anderson, 1983].

When we map that model on a semantic network the neurons are replaced by concepts and the synapses represent the semantic relationship between two concepts. One of the first system using spreading activation on semantic networks was the expert system GRANT. The main purpose for GRANT was to find sources of funding which are only given some research proposals [Cohen and Kjeldsen, 1987]. [Alani et al., 2003] describes in his publication Ontocopi. Ontocopi uses an spreading activation algorithm to identify communities of practice. IBM developed the Galaxy application which also uses spreading activation for searching through socio-semantic networks [Troussov A. and D., 2008].

[Scheir, 2008] gives a good overview of spreading activation approaches on semantic networks.

Spreading activation is an approach for extract informations from associative or semantic networks. Semantic networks have in contrast to associative networks as the name suggests a semantic meaning. That denotes that relation between two vertices (concepts) in the network represent some kind of semantic.

A network in that case is defined by a set interconnected nodes. A subset of at least one node is selected in that network (origin nodes). The basic idea of Spreading Activation follows the following algorithm: Some activation value is defined and assigned to the origin nodes. This activation spreads out over the network. The more activation energy a node receives the better is the relationship to the origin nodes.

3.4.1 Propagation over the Network

Having described the network for spreading activation in the last section, this section will deal with the spreading phase. The spreading phase consists of a sequence of iterations. Each iteration consists of:

1. one or more pulses
2. termination check

A pulse, also known as hop, denotes the spreading of activation from one node to all connected nodes. A single hop consists of

1. a preadjustment phase,
2. the spreading process and
3. a postadjustment phase.

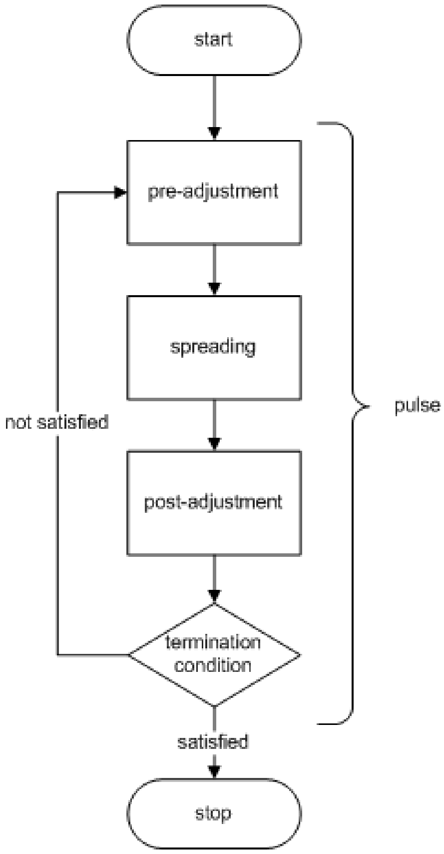

Figure 3.1: The steps of Spreading Activation

The optional preadjustment and the postadjustment phases append additional functionality to the spreading process itself. In the preadjustment phase it may be checked whether an edge is of a certain type and thus has to be traversed. Additionally the postadjustment phase can be used to apply some form of activation decay to the active nodes. This mechanism is used to avoid the retention of activation inside the network as well as the activation of the entire network. At the end of each iteration the algorithm checks if some termination condition is met. Spreading activation in it's simplest form works according to the following formula for the computation of the input activation:

$$I_j = \sum_i^k O_i * w_{ij} \tag{3.1}$$

The input of each node is determined by the output O_i of k nodes. The output of each node is multiplied with the weight w_{ij} of the associated link connecting node j with node i and the grand total for the k connected nodes is calculated.

The numerical type of the input and the weights is dependent on the application. The values define the strength of nodes respectively the relation between them. Usually they are real numbers, however they can also be expressed by binary values (0 or 1) or excitatory/inhibitory values (+1 or -1). Binary and excitatory/inhibitory values are usually used for semantic networks where the links are labeled. The reason is that the semantic value of the relation provides information to ascertain the value to be associatied to the link. Assotiative networks on the other hand generally provide a generic type of association, and that's why real values are used to weight the associations.

Since the computation of the input value has been adressed, the calculation of the output value must be determined. As has been stated on the input, the numerical type of the output also depends on the application. It is to be noted that the activation of a unit corresponds to it's output, i.e. the value for activation and the output value are equal:

$$O_i = A_i \tag{3.2}$$

To determine the activation of a unit the following formula depending on the input activation is used:

$$A_j = f(I_j) \tag{3.3}$$

Many different functions can be used for the determination of the activation, such as a linear function 3.4, a threshold function or a sigmoid function 3.5. However, the most often used function is the threshold function as desribed in 3.6. The threshold level has to be chosen depending on the type of application and every node may receive a different value as indicated by k_j.

$$A_j = \frac{1}{2} * x - 1 \tag{3.4}$$

$$A_j = \frac{1}{1 + e^{-I_j}} \tag{3.5}$$

$$A_j = \begin{cases} 0: & I_j < k_j \\ 1: & I_j > k_j \end{cases} \tag{3.6}$$

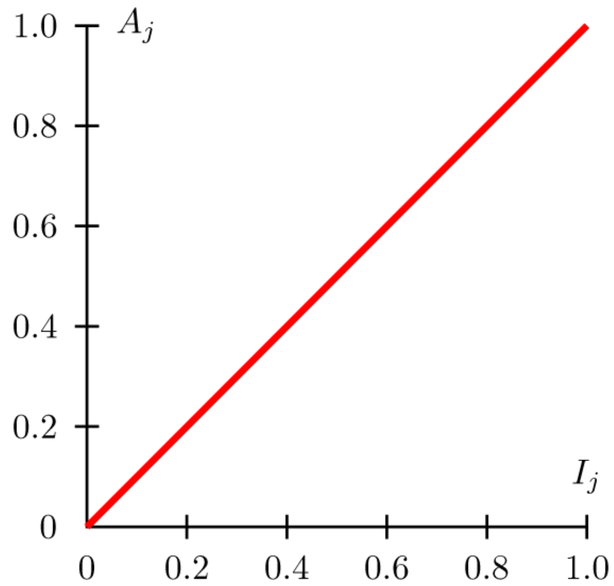

Figure 3.2: Example of a linear activation function from [Scheir, 2008]

The calculated activation which equates to the output of the unit (see 3.2) is passed to all the connected nodes. Typically the same value is transmitted to each node.

This process of activation as just described is iterated, pulse after pulse, reaching nodes far away from the origin nodes. At the end of each cycle a termination condition is checked. That means if the condition is fulfilled then the process of spreading activation stops, otherwise the process of sequencing pulses proceeds. Another reason for a break could be an empty priority queue. When there are no more nodes to process the algorithm cannot proceed. When the algorithm has stopped spreading out, the result, which is the activation level of the nodes, is retrieved. The interpretation and evaluation

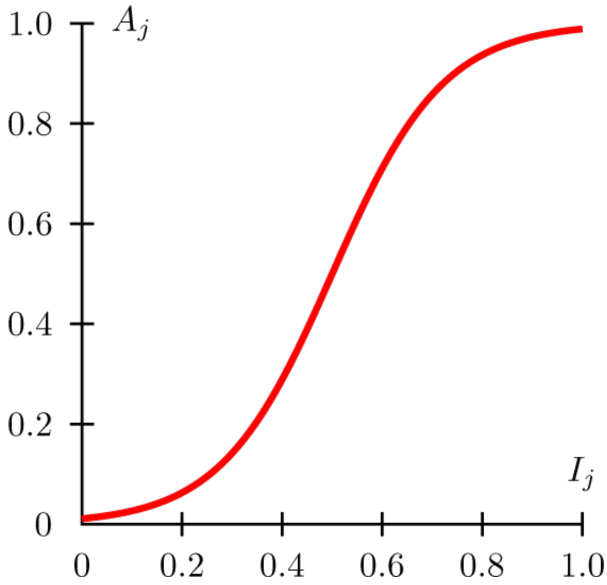

Figure 3.3: Example of a sigmoid activation function from [Scheir, 2008]

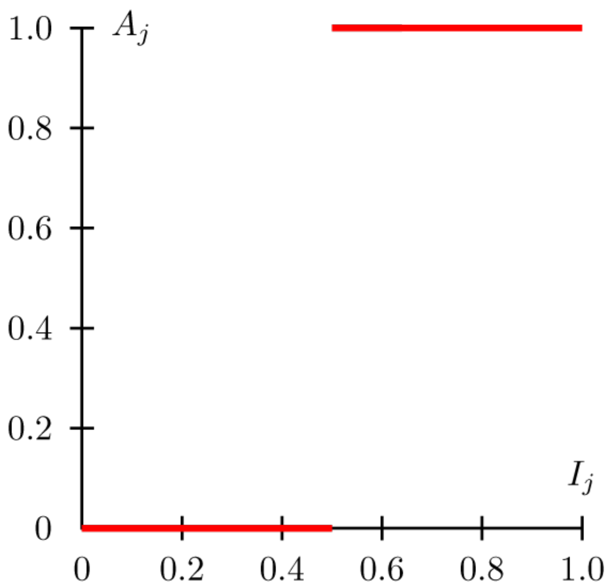

Figure 3.4: Example of a threshold activation function from [Scheir, 2008]

of the result depends on the application. After the specification of the basic spreading activation model the use of constraints to control the spreading of activation over the network has to be considered. That is the content of the following section.

3.4.2 Restrictive Spreading Activation

The problem of the so far described approach is that it lacks of appropriate control and restriction possibilities. It is quite possible that activation floods uncontrolled over the whole network. Another drawback is that the possible semantics are not considered in the basic spreading activation approach. In order to solve these problems the activation and it's propagation over the network has to be processed in order to rules. These rules, better known as constraints, are described and explained below [Crestani, 1997]:

Distance constraint: The spreading of activation arising from a unit stops as soon as a certain distance is reached. This can be explained by direct proportionality between the strength of the relation and the semantic distance. The closer two nodes are connected together, the more similar is their semantic. The classification of relations can be achieved by consideration the number of links. If two nodes are directly connected, than it is a first order relation. If there is only one intermediate node involved, then it is a second order relation, and so on. In the majority of cases only first, second and third order order relations are considered.

Fan-out constraint: Nodes with a very large number of connections to other nodes should receive special attention. The problem associated with very high connective nodes is that they usually have a very broad semantic meaning and activation could propagate out to too many nodes. A possible solution could be the diminishment of activation at nodes with a high connectivity, for example by division.

Path constraint: Activation should spread over preferred paths. Without setting a constraint activation would spread over all paths. For this problem there are some possibilities to counteract. One solution would be the setting of higher or lower weigths to all links in the network. An other solution especially for semantic networks is to take different weights into consideration, that means that different link types receive different activation. Certain paths with less significance can even be disregarded.

Activation constraint: Another possibility to control the spreading over the network is the setting of a threshold value. There are many alternatives for assigning an appropriate value for the threshold: One is the adaption of the threshold value after every cycle depending on the total level of activation over the network. Another possibility is the asssignment of different threshold levels to each node respectively to every set of nodes. A set could by defined by having the same semantic meaning in common.

The moment of application of these constraints presents an unresolved question so far. In general they are applied during the preadjustment phase (distance, fan-out and path constraint) respectively during the postadjustment (activation constraint) phase. It should be noted that the introduced constraints are merely examples for possibilities and make no claim to be complete.

4 Related Work

4.1 Introduction

The first part of this chapter deals whith some related work in the field of user context approaches. Due to the diversity and the importance of retrieval systems utilizing, amongst others, different kinds of spreading activation algorithms some related work in the field of assotiative networks and the application of spreading activation algorithms has been researched. This chapter should give an overview about the status of current research.

4.2 User Context approaches

The aim of the following sections is to give a short overview of actual research work in the field of the utilization of the user context and the features of the described projects.

4.2.1 Watson

Watson utilized contextual information in order to retrieve documents related to the current task. It suggests information by extracting contextual information of the current web site of the current word document. The aim is to provide just-in-time access to information relevant for the current task. Watson is described as an Information

Management Assistant (IMA). Watson observes the user interactions with applications (e.g. Microsoft Word, Internet Explorer) and automatically queries online information sources to fulfill the information need of the user. When observing the user representative keywords are extracted from active documents or websites using information retrieval metrics. These extracted informations are the input source for the generated queries. [Budzik and Hammond, 2000]

4.2.2 The EPOS project

The EPOS research project is described as a pro-active, context-sensitive support system. EPOS is utilizing an ontology based model of the knowledge worker's context. Context observation plugins have been developed for different application, e.g. Mozilla Firefox and Mozilla Thunderbird. The context data is gathered by utilization of the mentioned context observation plugins. The Personal Information Model (PIMO) created in the EPOS research project consists of incorporated ontologies, which are also able to reflect changes and updates of native structures such as file- or mail folder hierarchies. There is a multilayer ontology schema comprising domain indepentent ontologies, domain ontologies and the user's own domain ontology. Mappings between concept instances of different ontologies are realized by custom properties or by subclass/subproperty relations. Different application have been developed utilizing the Personal Information Model: A drop-box for filing which automatically suggests concepts describing a file that is to be saved. This categorisation is used to relate the file to an instance of the ontology. Another developed application is a desktop search tool called Gnowsis desktop search. Gnowsis utilizes the PIMO and in addition personalization rules can be applied on the results. [Sauermann et al., 2006]

4.3 Spreading Activation on Associative Networks

The aim of following section is to give an overview of actual existing Systems in the area of spreading activation in the field of associative networks:

4.3.1 Search Approach by Rocha et al. in 2004

[Rocha et al., 2004] describe "A Hybrid Approach for Searching in the Semantic Web". This approach "combines classical search techniques with spread activation techniques applied to a semantic model of a given domain" [Rocha et al., 2004]. The aim of the project is to utilize semantic search by providing an interface, which takes the user's information needs in terms of keywords, but performs sematic processing at the same time.

In the first step the knowledge represented in an ontology is utilized to transform the relation instances into an instances network by assigning a numerical weight to each relation instance. The outcome is a hybrid instances network, as it consists of symbolic (semantic label) and sub-symbolic (numerical weight) edges. The numerical weights are calculated by analysing the link structure of the ontology and computing a numerical weight for each relation instance.

Three different measures are presented in [Rocha et al., 2004] for computing the numerical weights: cluster, specificity, and combined. The cluster measure indicates the similarity of two related concept instances. The underlying idea of that measure is that the more relations any concept instance shares with other concept instances, the more similar they are. The specificity formula indicates the importance of a concept instance regarding the whole instances network. This measure is similar to the idf (inverse document frequency) but experiments showed better results with the specificity formula.

As mentioned above the described approach combines traditional keyword-based information retrieval with spreading activation techniques. The first two steps of the semantic search are the same as in traditional searches. In the first step the user enters his information need in terms of keywords into a traditional search engine (Lucene [Foundation, 2010]). For each node (concept instance) in the knowledge base, a node is created in the instances graph. The created node therefore contains all the property values belonging to a single concept instance concatenated into a single node. The instances graph provides the data for Lucene's index. The result of the traditional search provides a set of the best matching node instances ordered by their similarity with the given keywords. This set is taken for the initialization of the spreading activation algorithm. As the Lucene search engine provides a numeric value which reflects the relative importance of that node. Spreading activation is then performed by utilizing the already described weight mapping techniques.

Some nice features of the System are the computation of the shortest path to each activated node starting from one of the origin nodes and the retrieval of the node which has the greatest impact on the activation of each node in the result set [Rocha et al., 2004].

4.3.2 Ontocopi

[Alani et al., 2002] describe in their paper the ONTOlogy-based Community Of Practice Identifier. Communities of Practice are groups of people interested in a particular task (job, procedure, or work domain). The essence of these groups is that the membership is informal. At the same time this informal membership inheres the main problem because the members don't know about their affiliation. With support of the Ontocopi tool those communities of practice can be identified.

The system was developed as part of the Advanced Knowledge Technologies (AKT) project for ontology-based network analysis (ONA) at the University of Southampton. This project also originated the AKT ontology which represents the academic domain of one department of the University of Southampton. This ontology consists of concepts representing people, papers, projects, and conferences. The idea behind the Ontocopi tool can be described as follows: If there is no formal relation between X and Y, but they do have a (formal) relation to Z, then they could be part of a community of practice due to this informal relation.

In general the system works as follows: The user selects some concept from the user interface in the first step. This preselection specifies in which starting concept the user is interested in. As a response the system provides a list of concept instances. One instance of the list has to be selected and represents the origin node of the spreading activation algorithm. The Ontocopi tool supports three different relation configuration modes: manual, automatic, and semiautomatic mode. The configuration mode pertains the selection of relationships and the adjustment of the assigned weights. In the manual mode the user gains the complete control due to the manual choice of connections and the weighting of those. The automatic mode computes the occurence frequency of each relation type in the ontology to select and weight the relationships for the spreading activation algorithm. A potential problem of this approach is the unequally distribution of the relations in the ontology which can cause unrepresentative results. The semiautomatic configuration mode provides the opportunity to select some concepts from the ontology. This selection process acts as a concept filter for spreading activation, i.e. only the specified concepts are applied to the algorithm. Through that selection process the relation types are automatically filtered.

4.3.3 GRANT

One of the first systems applying constrained spreading activation is the GRANT system developed by Paul R. Cohen and Rick Kjeldsen [Cohen and Kjeldsen, 1987] [Kjeldsen and Cohen, 1987]. The intent of the GRANT system was the development of a semantic matching algorithm which detects matched between concepts. The similarity of the concepts is determined by the semantic relations between their properties. The expert system should provide the opportunity for researchers to find appropriate sources of funding. The motive of the development of the GRANT system was the lack of resource-saving methods to find agencies with a reasonable recall und precision. Previously a keyword database system was used whose search took about two hours with a precision about five percent. After switching to the GRANT system the computing time could be reduced to a few minutes. Compared to keyword matching the GRANT system performed better due to different used terms by researchers and agencies. While using syntactic matching at the keyword based approach the GRANT system utilized a large semantic network to receive better results. The system is well known for it's usage of many types of constraints especially for it's intensive use of path constraints, also referred as path endorsements. GRANT can be considered as an inference system applying repeatedly the same inference schema [Berger, 2003, Crestani, 1997]:

$$\text{IF } x \text{ AND } R(x, y) \rightarrow y \tag{4.1}$$

$R(x, y)$ can be thought as a link connecting node x and node y. This inference rule can be interpreted as: "if a researcher (or reader) is interested in topic X then he or she is likely to be interested in a related topic Y." [Cohen and Kjeldsen, 1987] By the use of path endorsements some paths can be preferred (positive endorsement) and others can be omitted (negative endorsement).

4.3.4 Ontology Extension Approach by Liu et al. in 2005

[Liu et al., 2005] introduce a system for semi-automatic ontology extension. This is accomplished by using the textual informations from 156 news media sites. To demonstrate the functionality of the system a seed ontology on "climate change" 4.1 is utilized to be extended.

The ontology is used for the initialisation of a Lexical Analyzer. The outcome of the analysis is a sematic network. This semantic network serves as a basis to find new concepts for an extended ontology. The following types of analysis are utilized to create the semantic network:

- Co-occurence analysis: By applying co-occurence analysis on document as well as on sentence level relevant terms can be included. Identified terms which are below a defined significance threshold are excluded.
- Trigger Phrases: Trigger phrases have the characteristic of containing parent-child descriptions. This means that by using the trigger phrase approach hierarchical relationships in web documents can be detected.
- Disambiguaton: Lexical analysis by utilization of WordNet lexical dictionary [Fellbaum, 1998]. WordNet is used to disambiguate the concepts of the seed ontology. The seed concepts are represented by vector space models to compare the similarity of the concepts against WordNet. When finished hyponyms, hypernyms and synonyms are the results of this step.

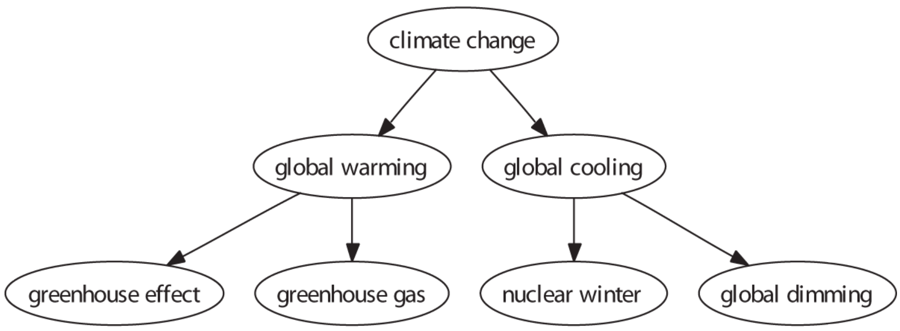

Figure 4.1: Example of a seed ontology from [Liu et al., 2005]

The outcome of the analysis phase is a semantic network which new terms connected to the seed ontology. The links are labeled according to the type of analysis. The weights of the links are calculated depending on the link type. After configuration of the semantic network a spreading activation algorithm is applied to identify appropriate concepts to include in an extended ontology.

4.3.5 Ad.M.In.

In [Berger, 2003] the Ad.M.In. (Adaptive Multilingual Interfaces) system is described as a multilingual information retrieval system. The aim of that system is the access respectively the retrieval of tourism information by natural language query formulation.

The knowlegde base of the information retrieval system is developed as an associative network by connecting several terms of the tourism domain. More precisely, the associative network is structured into three layers:

- Abstraction layer: The abstraction layer has the purpose to abstract terms from the conceptional layer by a more general term, e.g. the terms indoor pool and outdoor pool are expressed through the concept swim facility.
- Conceptual layer: The intention of this layer is the association of semantic related concepts by linking them together.
- Entity layer: This layer contains the entities of the tourism domain and associates them whith items of the conceptional layer.

Different weights are assigned to the links connecting the different concepts. These weights are manually determined before the network is beeing processed.

The query terms formulated in natural language are checked against errors and misspellings just before being linked to corresponding concepts of the abstraction respectively the conceptual layer. These linked concepts serve as origin nodes for the spreading activation approach. After a initialization phase, where all abstract concepts are replaced by concepts of the conceptual layer, the spreading activation process is activated. This process is repeated until a defined number of iterations is reached. The results are ranked in descending activation order.

4.4 Spreading Activation and Text Processing

[Schumacher et al., 2008] distinguishes between two different main research thrusts:

1. Semantic Document Retrieval
2. Fact retrieval

They have developed a semantic desktop search engine which utilizes semantic document retrieval and fact retrieval using a triple-based algorithm and the spreading activation graph travelsal algorithm. Regarding to the triple-based approach spreading activation shows the cohesion of the found triples. Expanding the query in this way provides the users of the semantic desktop a high-output search functionality.

Semantic Document Retrieval supposes the Retrieval of semantically enriched documents. Semantic Document Retrieval expands conventional keyword search with semantic techniques. One of these semantic techniques is the use of thesauri for query expansion. Another technique is the use of graph traversal algorithms to generalize or specify query terms or to match predefined categories. An enhancement of the Semantic Document Retrieval Approch is the utilization of Spreading Activation to improve the resluts by exploiting available domain knowledge. [Schumacher et al., 2008]

4.5 Summary

This section gave an overview over user context utilization systems and information retrieval systems applying spreading activation techniques on different network models. The aim of each approach is to provide pro-active, context sensitive support and/or to improve the retrieval performance. Since the discussed systems showed those promising results, the technique of spreading activation has been choosen as a ontology-bases search paradigm for the task identification approach developed in this thesis.

5 Concept and Design

As described in Section 1.2 it is the goal of this thesis to identify entities of a user interaction context model. In Section 3.4 the spreading activation approach in general has been indruduced. In the same section the application of a spreading activation approach on associative respectively semantic networks is described. Therefor the knowledge base, an populated user interaction context ontology has to be transformed into such a network model. This network model serves then as the base for the spreading activation approach delevoped in this thesis.

5.1 Introduction

The goal of this thesis is to identify entities of a user interaction context model relevant to a users current task. The user interaction context ontology serves as such a model. A relevant entity in the scope of this thesis describes an entity which is associated with the same task model instance as the current task. The task model is a class modeled in the user interaction context ontology. This thesis is based upon different knowledge bases using the same conceptualization (user interaction context ontology). Each knowledge base can be considered as the user interaction context model of a particular user. Due to the use of the same specification of the domain (the same ontology) all knowledge bases can be considered as one large knowledge base. For this reason the following parts of this thesis will use the idea of a single knowledge base.

This chapter deals with the detailed concept and design of the entitiy identification approach developed in this thesis. It will be explained how the construction of a semantic network is accomplished, and how the semantic network is utilized to identify entities. This also includes the determination of concepts and properties of the user interaction context ontology.

In addition, a visualization component of the semantic network is also presented in this chapter. The visualization provides on the one hand the facility to discover the coherences of the entities of the semantic network and on the other hand, to demonstrate the mode of operation of the spreading activation algorthm.

The further sections are structured as follows: First of all, the different steps from utilizing a user interaction context ontology up to the retrieval of identified entities are discussed in Section 5.2. Thereafter the model will be described in more detail. The transformation from the ontology to the associative network is discussed in Section 5.3.1. Furthermore the spreading activation approach as well as the representation of the network will be presented. Section 5.4 deals with the visualization components of the associative network on which the spreading activation algorithm is applied. Finally a short summary will round off this section.

5.2 Approach

Accessing tacit knowledge is a very challenging task. Although the knowledge is stored but the user is often not able the find appropriate information for the current situation. One approach we are using in this paper is registering the user's interactions with the desktop to posterior provide the user informations (resources) based on those registered activities. The user interaction context is observed by context sensors to automatically populate an user interaction context ontology.

In Figure 5.1 the different steps of the developed approach is presented. It shows a sketch of the populated user interaction context ontology as the basis for the entity identification approach developed in this thesis. Concepts and concept instances of the ontology are represented as nodes. Related concept instances are linked to each other by edges.

Figure 5.1 also illustrates the generation of the spreading activation graph by determining concepts and properties from the ontology. Since the main objective is to find a set of entities which are related to the initial set, the determination of appropriate concepts and properties is a challenging task. The concepts and properties are determined to keep the spreading activation graph as small as possible. The reason given is the size of the populated ontology. When taking the whole user interaction context ontology into account the spreading activation approach would lead to a nonsatisfying performance.

After setting up the spreading activation graph it is configured by definition of spreading activation specific attributes. These are the activation decay, the threshold level and the relation weights. After this configuration phase the spreading activation algorithm is processed in order to retrieve a set of resources ordered by relevance to the request.

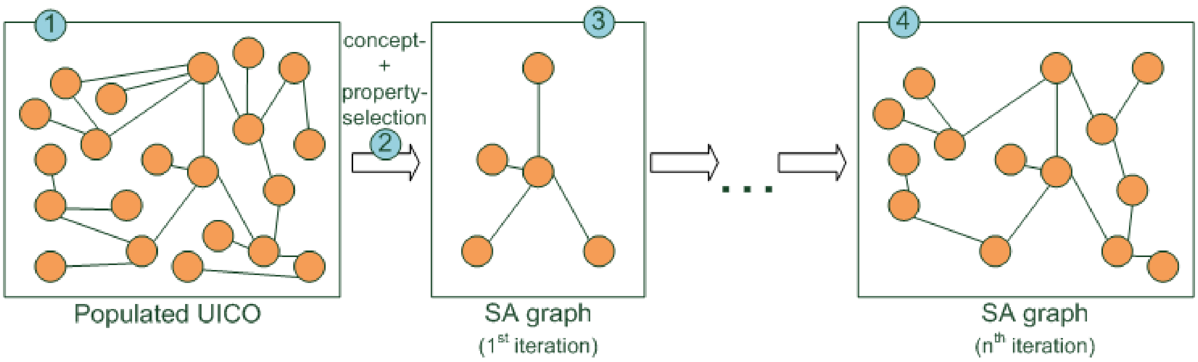

Figure 5.1: This figure visualizes the different steps used for the identification of relevant entities of a populated user interaction context ontology. (1) shows the populated user interaction context ontology as a starting point. (2) indicates the determination of concepts and properties to setup the spreading activation graph. (3) illustrates the start of the spreading activation processing. The process continues until at (4) a termination condition is met.

5.3 Model

In this Section the developed retrieval model will be presented. The goal is to provide better work and learn support to the user. Originating from the current task of a user the model enables the identification of relevant entities. As a basis for the identification of relevant entities the user interaction context utilized. The user interaction context is defined as "all interactions of the user with resources, applications and the operating system" [Rath2010]. The current user context can be for example a user writing an email or a word document. Relevant resources can be other resources (documents, persons, etc.) which could be a matter of interest for the user. The user interaction context is modeled as a user interaction context ontology. Populated with the support of context sensors, the ontology reflects a model of the user interaction context. An associative search method should find relevant concept instances for the current situation respectively the current user context. The current user context can be seen as a limited number of concept instances of the user interaction context ontology. For this purpose the populated user interaction context ontology serves as a basis for a network model which can be traversed by a spreding activation approach.

5.3.1 Establishing the Associative Network

Since spreading activation is not directly applicable on the knowledge base, an associative network is build based on the populated ontology. The objective is to exploit the structural relationships of the populated ontology in order to gain additional information. Because the spreading activation search is an iterative approach, it is not nescessary to build the entire network at once. At the beginning of development of this thesis it has been attempted to transform the whole knowledge base into an associative network at once. Due to memory- and computation-intensive operations an iterative approach has been adopted. This is also a beneficial for the performance of the complete system, since it has to deal with a huge amount of instances (over 1500 concept instances per task). The associative network is created and expanded just before the next spreading activation cycle. For each spreading activation cycle new concept instances are fetched from the knowledge base. For this reason, only those nodes and edges are inserted into the assotiative network, which are nescessary for the next iteration.

In order to setup the associative network, concepts and relationships have to be identified from the user interaction context ontology. Prior to the selection process of concepts, the ontology with all it's relationships has to be analysed manually. The aim in this thesis is to identify instances from a specific concept, namely the concept *Task*. It is regarded as imperative to find out, which concepts are feasible to occur in an informal relation between two tasks. Now assume the following example: There is is no formal relation between concept X and concept Y, but each of those concepts have a relation to concept Z (formal relation). This is a possible indication that concept X could be relevant for concept Y and vice versa (informal relation). In this case concept Z could be e.g. a word document or an online resource. Accordingly, the network is build in such a way that only those nodes and edges are constructed, which have the capacity to contribute to the identification of relevant entities, in this case relevant tasks. A manual analysis of the user interaction context ontology made it clear, that instances of the concepts *Event* and *EventBlock* have no influence on the results. This is because instances of the *Resource*-concept are associated with the Task, the Event, and the EventBlock instances at the same time this information is redundant and would unnecessary increase the size of the network. For this reason instances of the Event and the Event block concept are not considered for the network construction.

Furthermore instances with literal values, i.e. enties which have a relation to concept instances represented by datatype properties, are are not considered for the construction of the network. The reason is that literal values would represent end nodes in a network structure where activation energy cannot spread further.

The fetching of appropriate concept instances works as follows: For the first iteration only those concept instances are necessary, which are directly related to the current task. A triple pattern is then utilized to fetch all related concept instances as well as the relation itself. Then it is checked if the concept instance is considered at all. This is a special variant of an activation constraint. If the concept instance has to be considered, it is added to the associative network and accordingly connected to the other nodes. This step for adding new concept instances is repeated for every new activated node respectively concept instance in each iteration.

There are, however, certain concepts modeled in the ontology that would influence the results. For example, tasks are modeled in the ontology as they belong to a *TaskModel*. Thus, the *TaskModel* concept can be considered as some kind of categorisation for *Task* concepts. An identification with the inclusion of these concepts would therefore distort the results. The *TaskModel* is, however, nescessary for the automatic evaluation of the results.

The directions of the relationships are not considered in this case, i.e. if there is a relation between two concept instances a link is established in the network, regardless of the direction. After all nodes and edges have been created in the associative network another spreading activation cycle is performed. Those edges, which have been activated at the last cycle are utilized for the next query. This iterative process is continued as long as there are no more pattern matches or a termination condition is met. The detailed functionality of the spreading activation process on the associative network is described in the following section.

5.3.2 Spreading Activation on the Associative Network

The accotiative network as described in the previous section is search using a spreading activation approach. Starting from an initial activated node (origin node) activation spreads over the edges of the network and activates nodes that are connected with the initial activated node. The origin node is the node representing the current task. The activation for each node is determined by using equation 3.1. A maximum activation value is set, i.e. the activation energy of a single node is limited at this threshold. This prevents the over activation of the network.

A challenging part is the selection of the properties. The properties of the ontology represent the links in the associative network. The main emphasis is on receiving relevant entities to the current situation. Relevant entities have the characteristic that they are closely related to actual resources, e.g. the active word document. In other words, there are relationships between tasks with resources in between.

Just before each spreading activation cycle the originating nodes and the edges are analysed with regard to some important factors. Those factors are:

1. Cycling
2. Inverse direction and
3. Concept instance in scope.

The motivation to check for those three factors is grounded on the following ideas: While we don't want to visit activated nodes twice and so building up too much activation we have to check if we have already visited some specific node. Those nodes are no longer considered by the algorithm. The reason for the second factor is that there are sometimes relations between two node pointing in opposite directions. So if there is an inverse relation, the relation will be preventative filtered out. The third factor determines if the concept instance is in the scope of concepts. As the concepts are preselected only instances of those picked concepts will be used for the spreading activation approach.

The limit on iterations was set to 20 in the first version. An appropriate value for the iteration limit has been discovered as part of the evaluation process. Furthermore, spreading activation specific values such as activation decay, threshold level and relation weights have also been determined empirically.

5.4 Visualization

Most information retrieval systems that are available today have one big issue in common: The results don't indicate how they have been achieved. The objective of the visualization of the spreading activation graph is to illustrate the mode of operation of the entity identification approach developed in this thesis. Moreover the structural coherences between the different concept instances are visualized.

The benefit of the visualization for the user is to comprehend each iteration of the spreading activation algorithm. In this way the user can browse through a subpart of the user interaction context model.

As the user interaction context model can be seen as a graph, a graphical representation of the model has been created. This graphical representation consists of all concept instances and all relations used for the spreading activation approach. The graphical representation of a subpart of the user interaction context shows the interrelations between the different concept instances.

The visualization of the approach is some kind of representation of the results. The visualization provides an alternative interface for novice as well as for expert users. In order to provide support for the determination of relevant concepts and for the evaluation of spreading activation specific attributes, this kind of representation can be utilized by expert users. The visualization illustrates the selected subpart of the user interaction context model.

The visualization of the spreading activation graph has also a benefit for the developer of the spreading activation approach: It is a fact that there is little knowledge about the coherences between the concept instances of the populated user interaction context ontology due to the automatic population mechanisms not developed in this thesis. For this reason the visualizaton of the spreading activation graph presents the connections between the concept instances that are actually present. In this way the iterative construction of the spreading activation graph as well as the propagation of activation inside the graph can be displayed. Especially in the early stage of development it is a very helpful feature to demonstrate the coherences between action and reaction: In this case, the action would be an other configuration of the spreading activation graph and the reaction would be differet respectively differently activated entities.

6 Implementation

This chapter describes the implementation of the different components of the developed system. After a general description of the different parts of the system a prototype will be introduced. Furthermore a visualization unit, which was developed to illustrate the functionality of the spreading activation approach, will be presented.

6.1 Introduction

One requirement for the implementation of the entity identification approach was the integration into the existing KnowSe service framework. KnowSe is an abbreviation for Knowledge Services. KnowSe is implemented as an Eclipse RCP application. The benefits of an RCP application are: On the one hand the plug-in based architecture to produce modular structures and independent modules (OSGi-bundles), and on the other, a native user interface by utilizing the Standard Widget Toolkit (SWT) to access the graphical components provided by the operating system.

The different components developed in this thesis were implemented into different OSGi-bundles to seperate different functionalities. The most important Plug-in is the spreading activation plugin. It is responsible for the processing of the spreading activation steps and the fetching of required data. The required data is hold by another OSGi-bundle, namely the COBS Plug-in. COBS (Context OBservation Service) was designed and implemented as part of Andreas S. Rath's PhD research work. Further information about COBS can be taken from [Rath, 2010]. An important part of COBS is the RDF store which holds the user interaction context model. The OpenAnzo framework[OpenAnzo.org, 2010] was choosen so serve as a semantic middleware and triple store. The OpenAnzo framework utilizes Glitter, a SPARQL query engine to support the query of named graphs.

6.2 Components

The basic components developed in this thesis are: (i) the `SpreadingActivationManager`, (ii) the `SpreadingActivationGraph`, (iii) the `ContextVisualizationWebService`, (iv) the `ContextVisualizationView`, (v) the `StatementSelectionView` and (vi) `KnowSeWave`. The basic architecture is visualized in Figure 6.1. For completeness reasons (vii) the `ContextObservationService` is also pictured, even though it has not been developed in this thesis. The following paragraphs should provide a more detailed description of the individual components.

`SpreadingActivationManager:` The `SpreadingActivationManager` is the main component which utilizes the `ContextObservationService` on the one hand and holds the `SpreadingActivationGraph` on the other hand. The `SpreadingActivationManager` is responsible for the creation of the associative network and for the execution of the spreading activation algorithm. The nescessary data for the construction and expansion of the network structure provides the `ContextObservationService`. By means of a suitable interface the knowledge base represented by a RDF graph can be queried to fetch the required data. A detailed description of the queries can be found in Section 6.3. The results of the queries are then wrapped into nodes and edges to build the spreading activation graph. The `SpreadingActivationManager` provides several methods to define several constraints. These

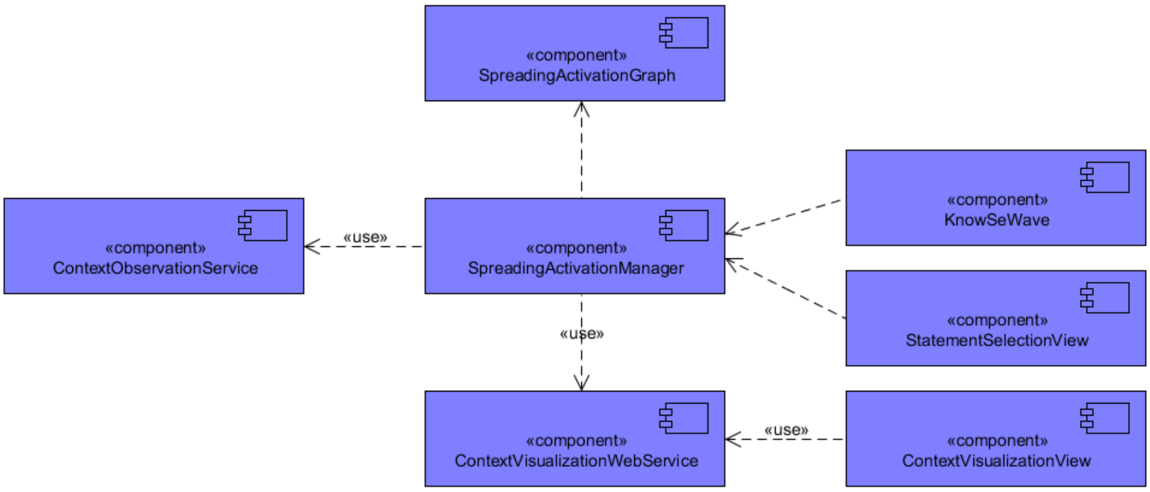

Figure 6.1: The component diagram showing (i) the `SpreadingActivationManager`, (ii) the `SpreadingActivationGraph`, (iii) the `ContextVisualizationWebService`, (iv) the `ContextVisualizationView`, (v) the `StatementSelectionView`, (vi) `KnowSeWave` and (vii) the `ContextObservationService`.

include the filtering for concepts as well as for property types. The maximum allowed number of cycles is also defined in the `SpreadingActivationManager`.

`SpreadingActivationGraph`: The `SpreadingActivationGraph` holds the graph structure for the graph traversal algorithm. Furthermore this component provides the calculation logic for the spreading activation algorithm. This means, therefore, that the computation of the activation value of each node in the network is managed by the `SpreadingActivationGraph`. Provided methods offer the possibility to process spreading activation cycle by cylce or until a termination condition is met, e.g. the maximum number of allowed cycles has been reached. The `SpreadingActivationGraph` is implemented as a `HashMap` containing all `Nodes` of the associative network. The resource URI serves as the key for the `HashMap`. Each node instance holds it's own set of links relating to other nodes.

`ContextVisualizationWebService`: Since the `ContextVisualizationView` requires a web service for fetching xml data to visualize it the `ContextVisualization-WebService` has been developed. The `ContextVisualizationWebService` is based on the jetty web server [Foundation, 2011]. The `SpreadingActivationManager` performs the transformation of the network structure into a xml representation and sends it to the `ContextVisualizationWebService`. The `ContextVisualization-WebService` makes the xml structure available for the `ContextVisualization-View`.

`ContextVisualizationView:` The `ContextVisualizationView` is based on the SWT `Browser` widget. It uses the rendering engine of the operation system, in the case of Microsoft Windows the Internet Explorer. On this account the `Browser` widget is also able to display Flash content. The content area of the `ContextVi-sualizationView` consists of such a Flash application. The application utilizes the Adobe Flex library RaVis [Bellone, 2011] to create a data visualization for the spreading activation graph. The data is fetched from the `ContextVisualization-WebService`.

`StatementSelectionView:` The `StatementSelectionView` is a tool that supports the user in constructing an assiciative network for spreading activation based on the existing knowlegde base. At the very beginning of this thesis it has been developed to get familiar with different settings. The `StatementSelectionView` is also a tool for demonstration purposes. The `StatementSelectionView`, which is implemented as an Eclipse RCP View provides a multitude of setting options. On the one hand a RDF resource can be choosen as origin node, which can be a task or a resource as defined in the user interaction context ontology. Moreover the concepts as well as datatype and object properties can be selected as constraints for spreading activation. The functionality is described in more detail in Section 6.4.1.

`KnowSeWave`: `KnowSeWave` is a first prototype that makes use of the spreading activation approach described in this thesis. While performing a task, KnowSe Wave searches through the user interaction model and recommends resources that might be appropriate for the current situation. This iterative processing of the user interaction context model is done by the `SpreadingActivationManager` which is utilized by `KnowSeWave`. A more detailed description of `KnowSeWave` can be found in Section 6.5;

`ContextObservationService`: The `ContextObservationService` is a component developed by Andreas S. Rath during his PhD research work. "The service interface provides methods for starting and stopping the context observation, [...] controlling the task handling and most importantly querying the UICO with the SPARQL semantic web query language." [Rath, 2010] The feature of querying the UICO is the basic feature used in this thesis. For this reason it will be further described in the next section.

6.3 Querying the User Interaction Model

A SPARQL query is composed of a set of triple patterns. "Triple patterns are like RDF triples except that each of the subject, predicate and object may be a variable." [Prud'hommeaux and Seaborne, 2008] SPARQL is utilized in the introduced approach to retrieve RDF resources respectively literals. By reason of the dimension of the user interaction context model an iterative execution approach has been developed.

The iterative execution of the spreading activation algorithm is accomplished by the `SpreadingActivationManager`. The `SpreadingActivationManager` is responsible for fetching the correct items from the context graph and for the iterative execution of the several spreading activation steps. This method is one of the most important parts of the module. While starting at a predefined `ConceptInstance` all nodes are fetched that are connected to that origin node.

The following code snippet illustrates a SPARQL query to fetch nescessary data for building the associative network. First, the variables to appear in the query results are identified(`?predicate, ?inversePredicate, ?object, ?propertyType`). The where clause provides the graph pattern which is defined to match against the graph. In this example, the graph pattern consists of two triple patterns. The pattern can be interpreted as follows: Return all statements, whose subject is equal with the resource `http://www.dyonipos.at/rdf#Task@ExampleTaskInstance`. In addition, the predicate must have a `?propertyType` set which will be bound to the identified variable. If there is an optional inverse predicate, then it will be bound to the `?inversePredicate` variable. Finally, the statements are filtered by object and datatype properties. The matching statements are then added to the associative network.

```
SELECT DISTINCT ?predicate
                ?inversePredicate
                ?object
                ?propertyType
WHERE {
  "http://www.dyonipos.at/rdf#Task@ExampleTaskInstance"
  ?predicate
  ?object .
  ?predicate
  <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
  ?propertyType
} .
OPTIONAL {
  ?predicate
  <http://www.w3.org/2002/07/owl#inverseOf>
  ?inversePredicate
} FILTER (
  sameTerm
  (str(
    ?propertyType),
  "http://www.w3.org/2002/07/owl#DatatypeProperty"
  ) ||
  sameTerm
  (str(
    ?propertyType),
  "http://www.w3.org/2002/07/owl#ObjectProperty"
```

```
    )
)
```

6.4 Detailed description of implemented Views

In Section 6.2 the different components have been described in general. The following sections will provide a more detailed description of the visual components developed in this thesis. Section 6.4.1 describes the different visual components of the `StatementSelectionView`. Section 6.4.2 will give an overview of the capabilities of the graph visualization component. Finally Section 6.5 will describe a first prototype using the user interaction context ontology to provide pro-active information retrieval. It has to be particularly mentioned that this prototype called Knowse Wave is utilizing the spreading activation algorithm developed in this thesis.

6.4.1 Statement Selection View

As already mentioned the `StatementSelectionView` is used as a tool to construct the associative network. Moreover it provides the functionalities to easily set spreading activation constrains as well as a simple tablular represenatation of the spreading activation results. Figure 6.2 shows a screenshot of the `StatementSelectionView` and its different components.

On top of the `StatementSelectionView` there is a group with two buttons together with a table. Figure 6.3 shows an enlarged view. The two buttons trigger a fetching process of concept instances from the knowledge base. As the buttons indicate they are responsible for the fetching of tasks respectively resources from the populated context ontology. The results are then visualized in tabular form. The checkboxes are used to select a concept instance as origin node for the spreading activation algorithm.

The next part of the `StatementSelectionView` shows the concept selection area.

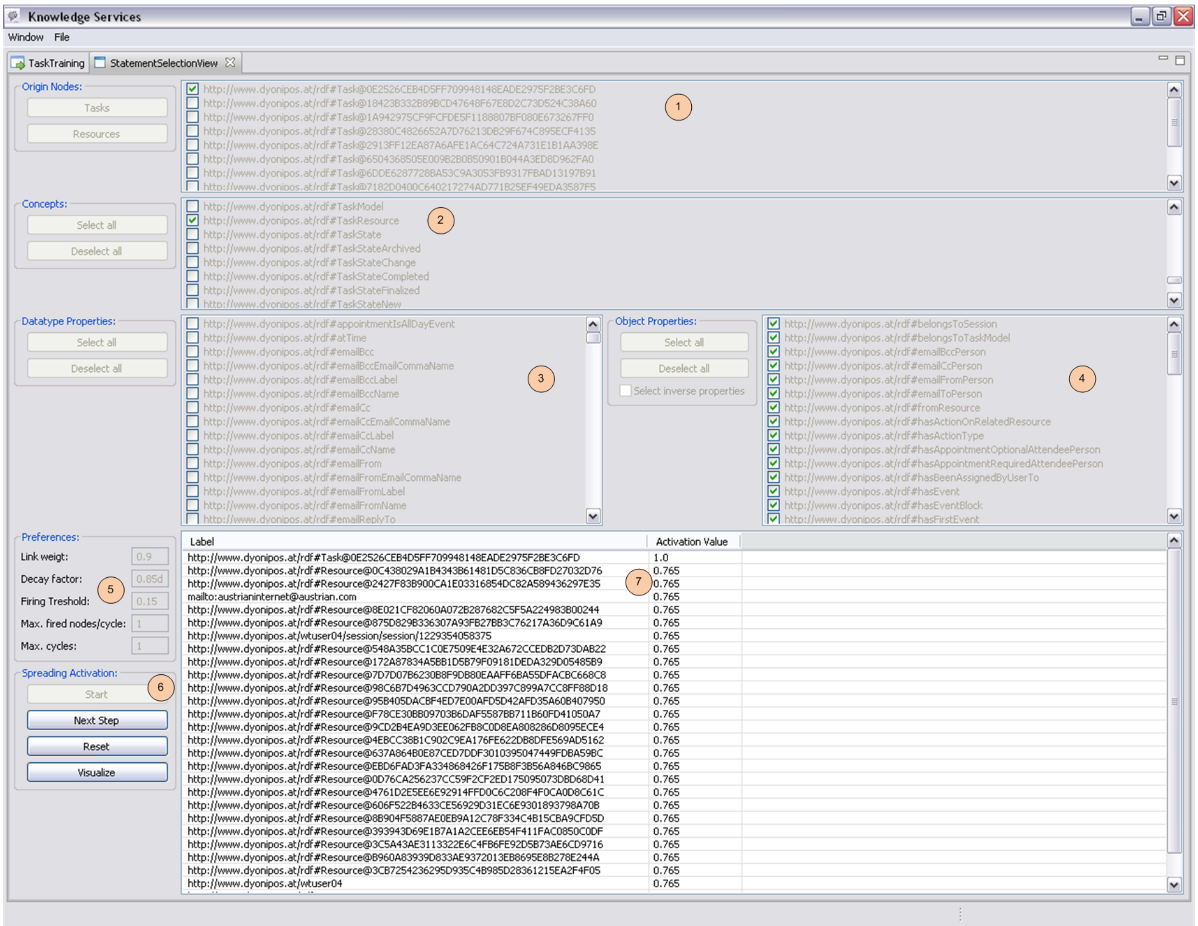

Figure 6.2: This figure visualizes the different parts of the Statement Selector. (1) shows the different tasks, from which one has to be selected as origin node for spreading activation. (2) diplays the different concepts from the user interaction context ontology. (3) and (4) show two tables with checkboxes for selecting the datatype and the object properties to use for the spreading activation approach. (5) displays the attributes which have to be set. (6) acts as execution control and (7) displays the calculated results.

Figure 6.3: This figure shows a detail of the StatementSelectionView. The buttons are fetching tasks respectively resources from the populated context ontology. The results are visualized in tabular form. For Concept instance selections as origin nodes the checkboxes are used.

Figure 6.4: This figure shows a detail of the StatementSelectionView. The table displays all concepts of the user interaction context ontology. Selections in this table are interpreted as constraints for the spreading activation model. The two buttons trigger a select/deselect event for all items in the table.

Figure 6.4 shows this part of the `StatementSelectionView`. The table displays all concepts of the user interaction context ontology. The selection of items in the table effect a constraint for the spreading activation model. This means that only those instances of concepts are fetched from the knowledge base which are selected in this table. The two buttons trigger a select/deselect event for all items in the table.

Below the concept selection part of the `StatementSelectionView` is the property selection area. Figure 6.5 displays all datatype properties of the user interaction context ontology. Datatype properties relate a concept instance (object) to a data value (String, Integer). The selection of items in the table effect a path constraint for the spreading activation model. It influences the structure of the spreading activation network as follows: When a datatype property is selected in the table it is used in the network, otherwise is is disregarded in the spreading activation process. The two buttons trigger

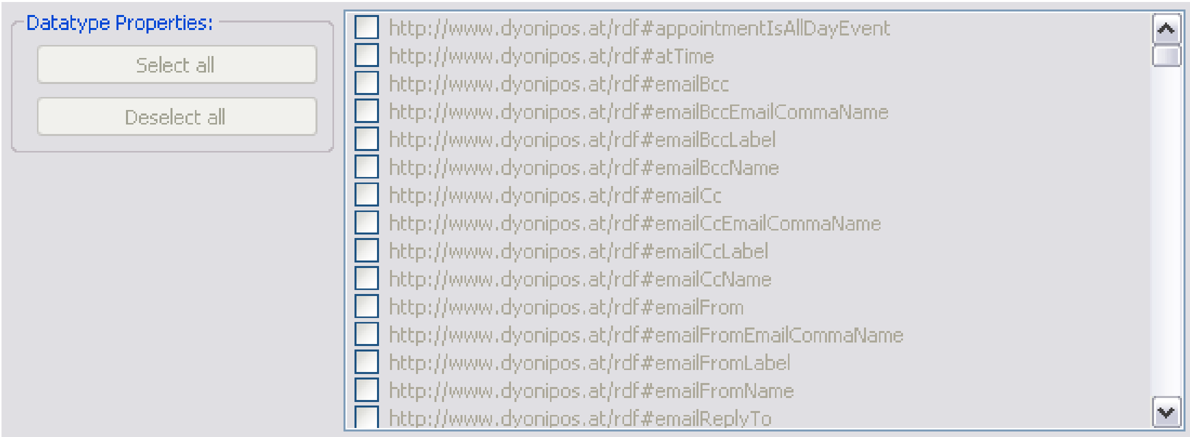

Figure 6.5: This figure shows a detail of the StatementSelectionView. The table displays all datatype properties of the user interaction context ontology. The selection of items in the table effect a path constraint for the spreading activation model. The two buttons trigger a select/deselect event for all items in the table.

a select/deselect event for all items in the table.

Next to the datatype selection area is the selection area for object properties. Figure 6.6 displays a table whith all object properties of the user interaction context ontology. Object properties relate a concept instances (objects) to other concept instances. The selection of items in the table effect a path constraint for the spreading activation model. It influences the structure of the spreading activation network as follows: When a object property is selected in the table it is used in the network, otherwise is is disregarded in the spreading activation process. The two buttons trigger a select/deselect event for all items in the table. Another special feature offers the checkbox which is the last element in the object properties group. If there is an inverse object property defined in the user interaction context ontology, the corresponding element is automatically selected in the table.

At the bottom of the `StatementSelectionView` the most important part of this view can be seen. On the left side of that part two groups can be identified. The Preferences group is responsible for spreading activation specific settings. The settings are used to initialize the `SpreadingActivationManager`. The Spreading Activation group controls

Figure 6.6: This figure shows a detail of the StatementSelectionView. The table displays all object properties of the user interaction context ontology. The selection of items in the table effect a path constraint for the spreading activation model. The two buttons trigger a select/deselect event for all items in the table. The checkbox which is the last element in the object properties group offers an automatic selection event of all inverse object properties defined in the user interaction context ontology.

the excecution of the spreading activation algorithm. The last button in this group is used to open the `ContextVisualizationView`. On the right side of the image a tabular representation of the spreading activation results can be seen. The first column provides a short description of the concept instance and the second column indicates the level of activation.

6.4.2 Context Visualization View

The `ContextVisualizationView` is a component to visualize to results of the spreading activation approach. The `StatementSelectionView` offers the command to open the `ContextVisulaisationView` in a new tab. As previously described the `ContextVisualizationView` is based on the SWT Browser widget. This widget has been configured, so that an Flash application is started each time the view is opened. The Flash application itself is based on the Adobe Flex library RaVis [Bellone, 2011]. The development effort of the application could be kept low due to the use of a demo application of the

Figure 6.7: This figure shows a detail of the StatementSelectionView. On the left side two groups can be identified. The Preferences group is responsible for spreading activation specific settings. The settings are used to initialize the `SpreadingActivationManager`. The Spreading Activation group controls the excecution of the spreading activation algorithm. The last button in this group is used to open the `ContextVisualizationView`. On the right side of the image a tabular representation of the spreading activation results can be seen. The first column indicate the level of activation and the second column provides a short description of the concept instance.

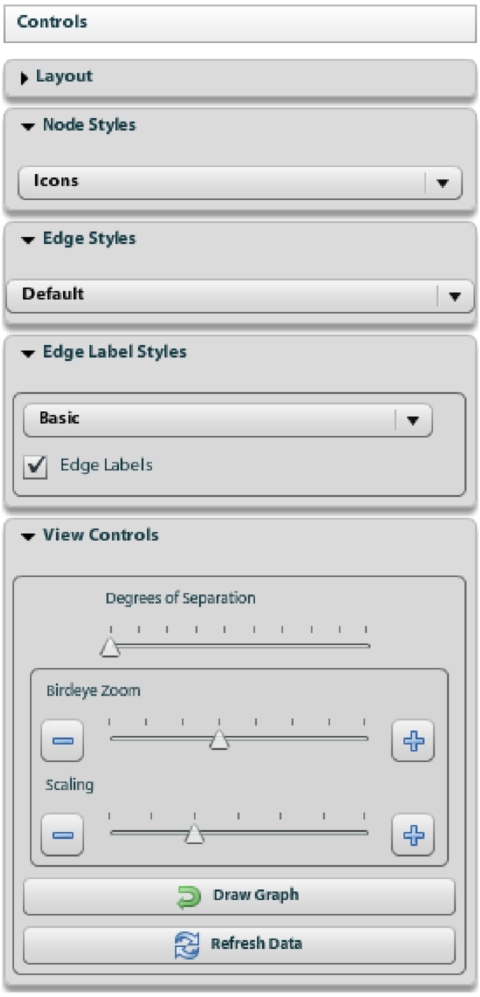

Figure 6.8: The control panel of the `ContextVisualizationView`. This panel provides the control of graph specific settings such as layout changes, node and edge styles, as well as zoom level changes.

RaVis website. Only some smaller adaptions have been implemented to fit the needs. These include the integration of the web service usage instead of using a single file as source as well as the addition of some images for different concepts. Figure 6.8 shows the control panel of the `ContextVisualizationView`. This panel provides the control of graph specific settings such as layout changes, node and edge styles, as well as zoom level changes.

The main Viewer area of the `ContextVisualizationView` is for the display of the graph structure fetched from the `ContextVisualizationWebService`. Figure 6.9, 6.10

and 6.11 show the associative network after different number of spreading activation cycles.

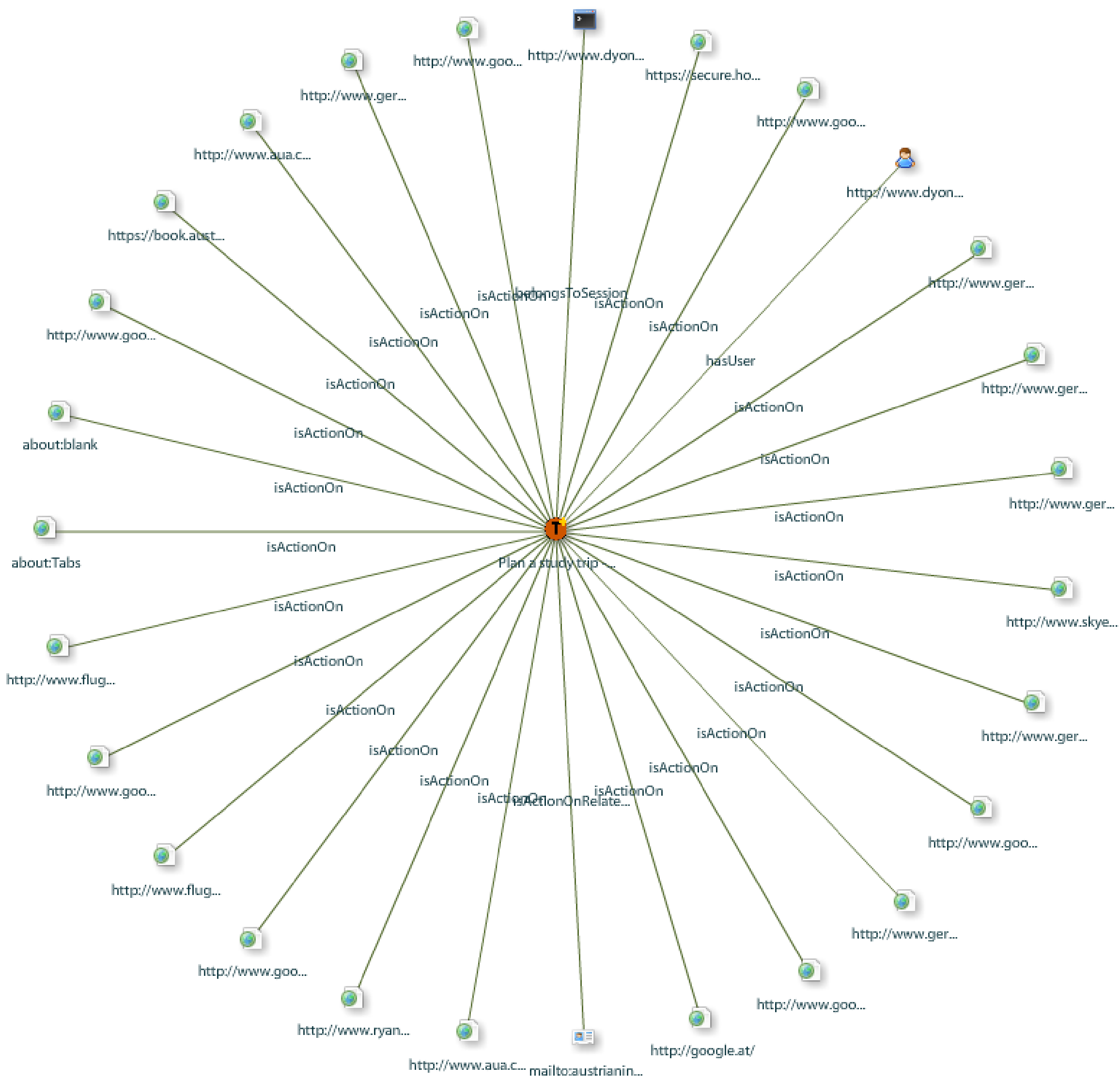

Figure 6.9: `ContextVisualizationView` which shows the associative network created after configuration in the `StatementSelectionView`, thus after one iteration step. This screenshot shows the representation of an instance of the concept Task in the middle colored light brown. The connections to other concept instances are also visualised in this view. The connections respectively the edges are labeled with the corresponding property.

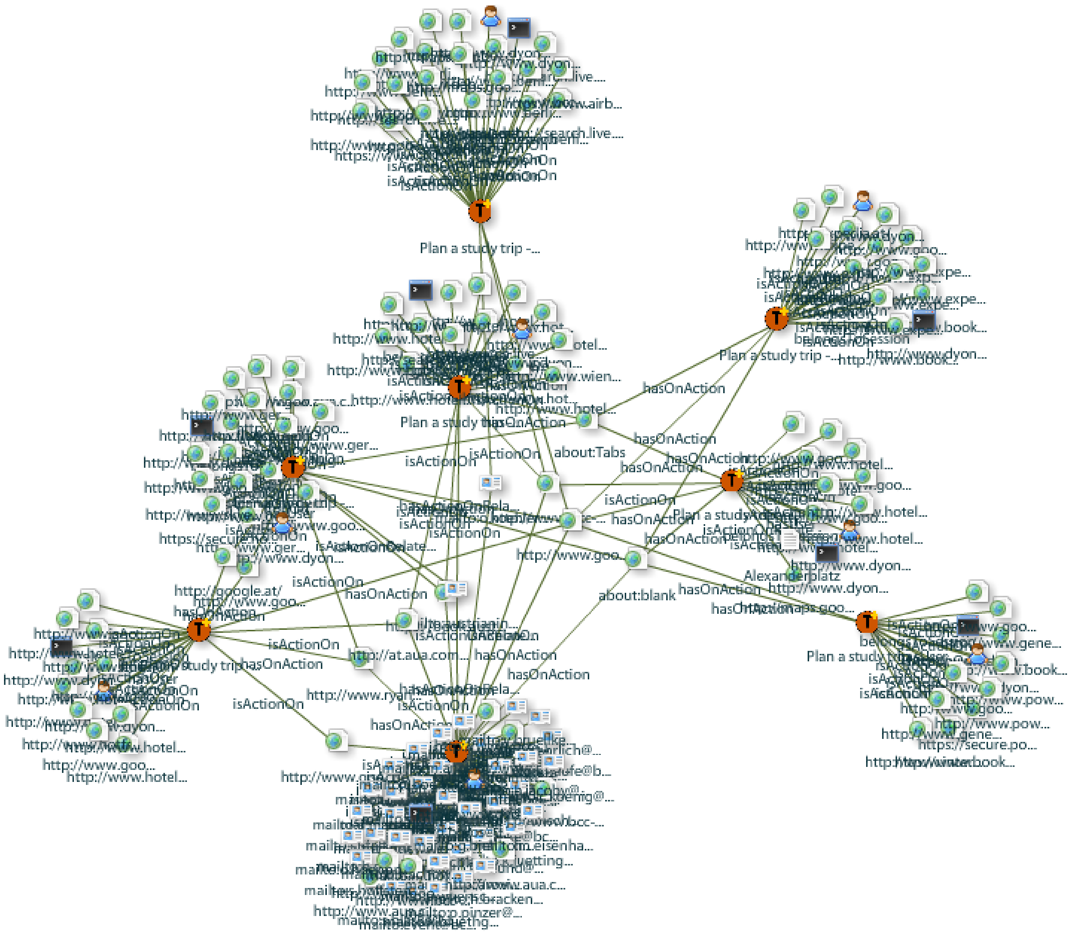

Figure 6.10: `ContextVisualizationView` which shows the associative network created after configuration in the `StatementSelectionView`, thus after three iteration steps. The screenshot shows the representation of 8 instances of the concept Task. The task instance at the bottom was the origin node of the spreading activation process. The images shows that the task instances are related to other tasks by sharing the same instance of a resource. Such connections form the basis for the spreading activation approach described in this thesis.

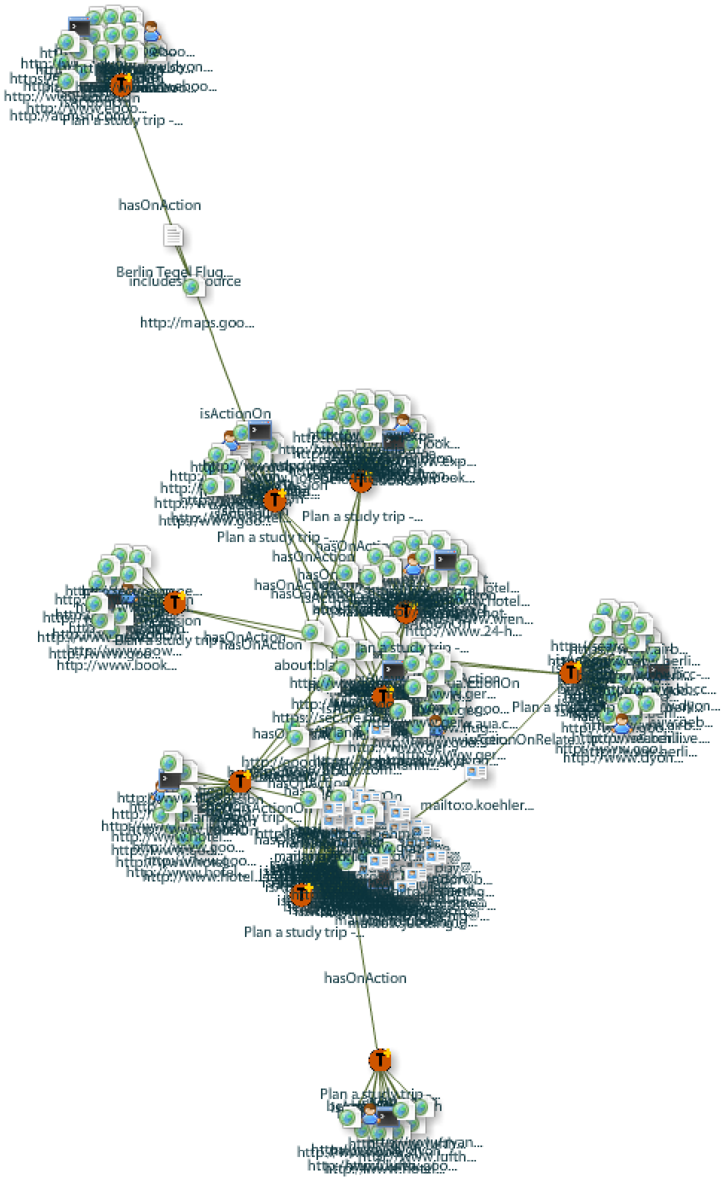

Figure 6.11: `ContextVisualizationView` which shows the associative network created after configuration in the `StatementSelectionView`, thus after 7 iteration steps. The screenshot shows the representation of 10 instances of the concept Task. The task instance in the center was the origin node of the spreading activation process. The image shows that the task instances are related to other tasks by sharing the same instance of a resource. The task instance node at the top and the node at the bottom show the large distance to the origin node. This could mean that these two task instances and in particular their used resources are not relevant for the current task (the origin node).

6.5 KnowSe Wave

KnowSe Wave is the first prototype that takes the populated user interaction context ontology into account. It utilizes the spreading activation approach described in this thesis and provides pro-active information retrieval It demonstrates a possible approach how to enable better learn and work support. While performing a task, KnowSe Wave searches through the user interaction model and recommends resources that might be appropriate for the current situation.

The aim of KnowSe Wave is the support of the user during task executions. This means that the context model is growing during runtime. Since the results of a spreading activation approach refer to the state of the context model at a certain point in time, the spreading activation approach has to take the modification of the context model into account. This obviously means that the results of a spreading activation approach have to be readjusted to the changing context model. For this reason an iterative algorithm has been implemented that is reapplied on the context model in predefined time intervals. For testing purposes an interval of 20 seconds has been choosen. That means that every 20 seconds the context model is transformed into an associative network and the developed spreading activation algorithm is applied. The results of that approach are finally visualized in a RCP View. Each view in RCP is used to display some kind of information or to change some data in the application. In this case the view displays the results of the continually execution of the spreading activation cycles in tabular form. In general the table consists of three columns:

1. **Relevance indication**: A flag indicating the relevance of the concept instance. The color is based on the activation value. For testing purposes the following ranges have been defined: A green flag indicates an activation value between 1 and 0.78. A yellow flag indicates an activation value between 0.77 and 0.33. And finally a red flag indicates an activation value between 0.32 and 0.

2. **Activation value**: A numerical value representing the calculated activation value for this concept instance.

3. **Concept type**: An image representing the concept type.

4. **Concept description**: A short description of the concept instance, e.g. the URI for a website or the filename for a word document.

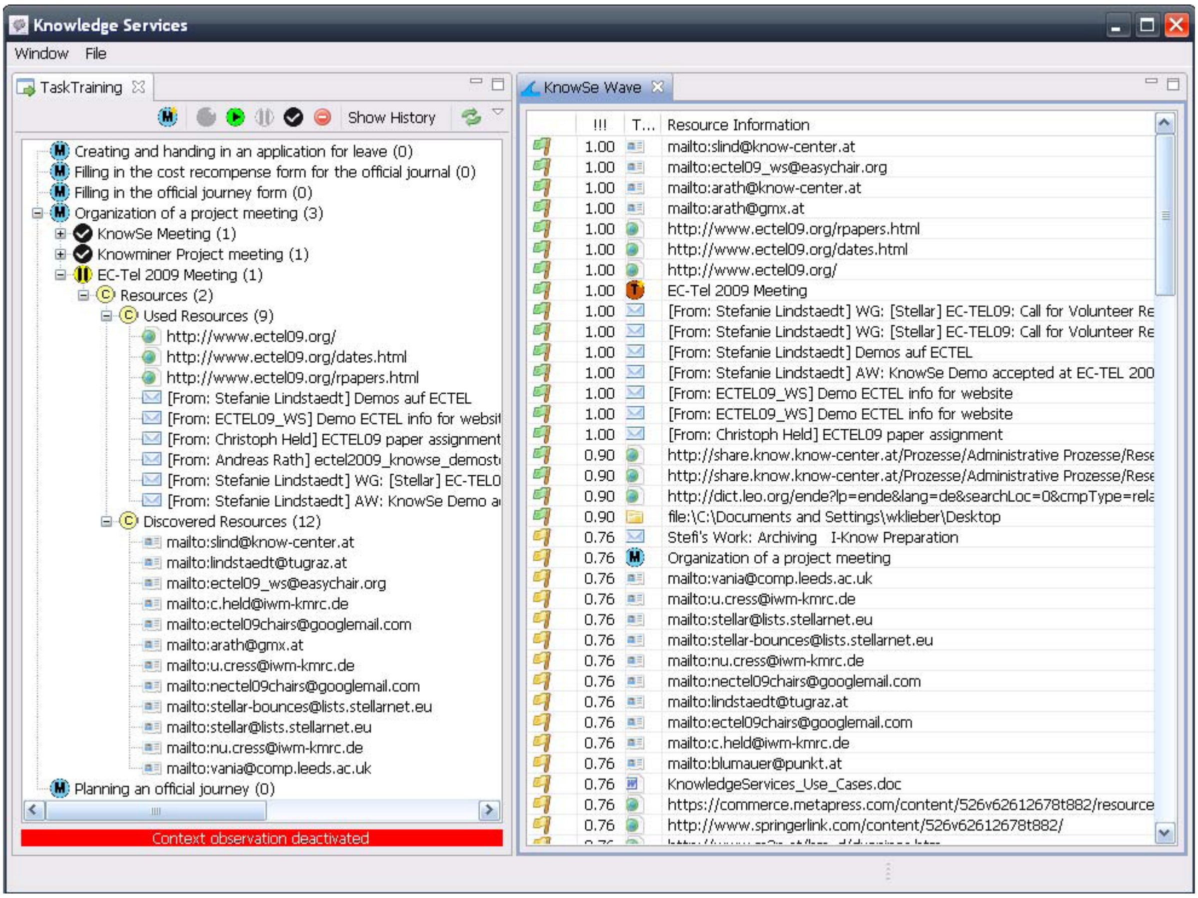

Figure 6.12: A first prototype named KnowSe Wave that takes the populated user interaction context ontology into account and uses spreading activation as a graph traversal algorithm. The rigth part of the figure shows the UI of KnowSe Wave. The first two columns indicate the relevance of recommendation. The third column shows the type of the suggested resource and the last column provides a short description of the suggestes resource.

7 Evaluation

In this thesis a system has been developed that utilizes a populated user interaction context ontology in order to identify relevant entities based on the current situation. More precisely, it has been investigated if the structure of a ontology-based user interaction context model is suitable to identify relevant resources. A network-based model has been implemented on the basis of the user interaction context ontology to enable the application of a search method for associative networks. In order to determine whether the identified entities are useful to support the user an evaluation of the system has been performed. For this reason different test sets have been utilized to prove the performance of the system. By means of different performance measures the empirically determined values are evaluated.

This chapter deals with the evaluation of the developed approach. First of all the performance measures used for the evaluation will be explained in detail. Afterwards the test sets are described in detail combined with the achieved results. Subsequently the results and their relevance are discussed. Finally, a short summary about the findings rounds up this chapter.

7.1 Introduction

In order to judge about the effectiveness of the retrieval a major simplifying assumption has been made: Tasks are modeled in the ontology as they belong to a TaskModel. Thus, the TaskModel concept can be considered as some kind of categorisation for Task concepts. A task is relevant to a user's current task if they belong to the same task model (category). The different test sets all have the characteristic that similar tasks are annotated whith the same concept (task model). Hence, it can be determined whether an identified entity is relevant to a user's current task.

The spreading activation algorithm on its own does not indicate any performance measures. In order to determine the effectiveness respectively the usefulness of the developed system it has been applied on different testsets of real user data. By using these test collections of real user data the performance of the developed approach becomes measurable.

7.2 Performance Measures

There is a multiplicity of methodologies to measure the performance of information retrieval systems. The most frequently used performance measures are precision and recall. [Manning et al., 2008] denote them as the two key statistics to assess the effectiveness of an IR system. They phrase precision as the fraction of the retrieved results that are relevant to the informantion need. Precision is defined as:

$$precision = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{retrieved\ documents\}|} \tag{7.1}$$

[Manning et al., 2008] phrase recall as the fraction of the relevant documents in the collection that are retrieved by the system. Recall is defined as:

$$recall = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{relevant\ documents\}|} \tag{7.2}$$

The weighted harmonic mean between precision and recall is called the F_1-measure [Guyon et al., 2006] and is defined as:

$$F = 2 * \frac{precision * recall}{precision + recall} \tag{7.3}$$

Applied to the information retrieval model developed in this thesis, documents can be substituted by concept instances. A challenge is the correct identification of relevant concept instances for the actual user context. By analysis of the user interaction context ontology it becomes obvious that resources (e.g. documents, web pages, emails, persons, appointments, notes, etc.) are directly related to the task in which it was used.

7.3 Evaluation Setup Design

Many different use cases, which were developed by Andreas S. Rath [Rath, 2010], have been executed in the context of a series of laboratory experiments.

For the evaluation of the information retrieval system a set of requests has been defined and the retrieved results have been evaluated. The requests to the system concern a set of concept instances of the populated user interaction context ontology. The intention as described in Section 1.4 is the following: The user interaction context can be seen as a cloud of nodes and edges. At some areas those clouds overlap each other, i.e. there is either a shared concept instance (e.g. a resource) or some kind of relationship between two concept instances and therefore an edge refers to this relation.

In order to achieve significant evaluation results, three different usage datasets have been selected. A characteristic which all datasets have in common is the modality, i.e. different users have been observed and recorded simultaneously while executing several tasks by means of the KnowSe system. The participants of these laboratory experiments were from two different domains. The first laboratory experiment has been performed in the domain of the Know-Center GmbH, Austria's competence center for knowledge management. Computer science students of Graz University of Technology were involved in the other two laboratory experiments. The experiments with the students have been supervised by the students Philipp Ghirardini, Daniel Resanovic and myself. The recorded user context data has been exported to different RDF-files to utilize them at a later time.

The serialized RDF-data imply a collection of RDF statements. Those statements actually reprepresent a labeled, directed graph. Every executed task is represented by a single RDF file respectively a RDF graph. The experiment data is evaluated by loading all different graphs into the triple store. The triple store is queried to establish a network structure containing a subgraph to perform spreading activation on it.

Since the spreading activation on a context model constitutes a very novel field of research empirical values are sparse. To determine a good configuration of the model, several different configuration settings have been specified. Due to variations in the activation decay, the threshold level, and the relation weights, the best possible results should be retrieved. For the model developed in this thesis the following values have been choosen:

- Activation decay: 0.95, 0.80, 0.65
- Threshold level: 0.8, 0.5, 0.2
- Relation weight: 0.9, 0.5, 0.2

Based on the evaluation results, the following questions should be answered:

- How suitable is the presented model in order to identify relevant tasks?
- Which concepts and properties are necessary for the identification of entities?
- How many iterations are required for achieving a good outcome?

- What are the best values for the spreading activation specific attributes such as activation decay, threshold level and relation weights?

7.4 Evaluation Setup 1 - Task Detection with KnowSe Dataset

The usage data for the first evaluation setup was the outcome of an experiment at the Know-Center GmbH, as mentioned above. Tasks of different categories (task models) have been performed by multiple users on laboratory computers as well as on their own personal computers. Summarizing tasks of five different task models have been performed by each person with modifications in the environment and in the specificity. In total, 218 tasks (task instances) have been performed.

The analysis of the generated results show that precision and recall are changing only on every second iteration. For this reason a more detailed consideration of the user interaction context ontology seems appropriate. By taking the mode of operation of the spreading activation approach into account, it becomes clear that only adjacent nodes respectively concept instances can be activated during a single cycle. The structure of the instantiated user interaction context onotolgy seems to be the reason for this behavior of changing precision only every second iteration. Instances of the task concept are not directly connected with each other. There are no object properties that could interconnect two instances of the concept task. Instead, there are relations to instances of the resource concept.

Table 7.1 illustrates the first change of the recall performance measure at the second iteration. This seems to be plausible since starting from a task instance the spreading activation approach will not find an other task instance at the first cycle. The colored cells reflect the trend of the recall for each column. For example, the second column in Table 7.1, representing an activation decay of 0.8, a threshold level of 0.2 and a relation weight of 0.9, indicates a positive progression until iteration 10. That means that after iteration 10 no more task instances are found that have the same task model in common with the starting task instance.

Table 7.2 illustrates the precision performance of the spreading activation task identification approach on the KnowSe dataset. The reason for the changing values only on every second cycle is still the structure of the user interaction context model. The colored cells reflect the trend per column as already mentioned at Table 7.1. The table indicates that the precision decreases at iteration 4. This means that at iteration 4 the proportion of task instances belonging to the same task model as the origin task instance decreases. This fact could be measured for all configuartion combinations of the spreading activation graph.

It can be noticed that the recall and the precision performances behave contrary. The recall performance increases with the number of iterations but on the other hand the precision performance decreases at the same time. The F_1-measure is the harmonic mean between the meantioned performance measures. Figure 7.1 shows the trend of the F_1-measure over 20 iterations for different configuration combinations. The harmonic mean shows the highest slopes at iteration 2 and 4. After iteration 4 the performance increases only marginally and not at all configuration combinations.

iteration	d_0.8_t_0.2_w_0.5	d_0.8_t_0.2_w_0.9	d_0.8_t_0.5_w_0.9	d_0.65_t_0.2_w_0.5	d_0.65_t_0.2_w_0.9	d_0.65_t_0.5_w_0.9	d_0.95_t_0.2_w_0.5	d_0.95_t_0.2_w_0.9	d_0.95_t_0.5_w_0.9	d_0.95_t_0.8_w_0.9
1	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
2	0,10	0,10	0,10	0,10	0,10	0,10	0,10	0,10	0,10	0,10
3	0,10	0,10	0,10	0,10	0,10	0,10	0,10	0,10	0,10	0,10
4	0,13	0,29	0,22	0,12	0,29	0,14	0,22	0,29	0,29	0,16
5	0,13	0,29	0,22	0,12	0,29	0,14	0,22	0,29	0,29	0,16
6	0,14	0,44	0,33	0,12	0,39	0,16	0,27	0,47	0,44	0,22
7	0,14	0,44	0,33	0,12	0,39	0,16	0,27	0,47	0,44	0,22
8	0,15	0,54	0,40	0,12	0,45	0,18	0,31	0,56	0,54	0,27
9	0,15	0,54	0,40	0,12	0,45	0,18	0,31	0,56	0,54	0,27
10	0,15	0,56	0,45	0,12	0,48	0,19	0,32	0,57	0,56	0,31
11	0,15	0,56	0,45	0,12	0,48	0,19	0,32	0,57	0,56	0,31
12	0,15	0,56	0,46	0,12	0,49	0,19	0,32	0,58	0,56	0,33
13	0,15	0,56	0,46	0,12	0,49	0,19	0,32	0,58	0,56	0,33
14	0,15	0,56	0,47	0,12	0,49	0,20	0,33	0,58	0,56	0,34
15	0,15	0,56	0,47	0,12	0,49	0,20	0,33	0,58	0,56	0,34
16	0,15	0,56	0,47	0,12	0,49	0,20	0,33	0,58	0,56	0,35
17	0,15	0,56	0,47	0,12	0,49	0,20	0,33	0,58	0,56	0,35
18	0,15	0,56	0,47	0,12	0,49	0,20	0,33	0,58	0,56	0,35
19	0,15	0,56	0,47	0,12	0,49	0,20	0,33	0,58	0,56	0,35
20	0,15	0,56	0,47	0,12	0,49	0,20	0,33	0,58	0,56	0,35

Table 7.1: The table shows each iteration's recall values of 20 spreading activation iterations. The columns represent different combinations of activation decay (d), threshold level (t) and relation weights (w). Since not all possible combinations (27 in total) result in positive values, those zero-values are blinded out. The table illustrates the first change of the recall performance measure at the second iteration. This seems to be plausible since starting from a task instance the spreading activation approach will not find an other task instance at the first cycle. The colored cells reflect the trend of the recall for each column. For example, the second column indicates a positive progression until iteration 10. That means that after iteration 10 no more task instances are found that have the same task model in common with the starting task instance.

iteration	d_0.8_t_0.2_w_0.5	d_0.8_t_0.2_w_0.9	d_0.8_t_0.5_w_0.9	d_0.65_t_0.2_w_0.5	d_0.65_t_0.2_w_0.9	d_0.65_t_0.5_w_0.9	d_0.95_t_0.2_w_0.5	d_0.95_t_0.2_w_0.9	d_0.95_t_0.5_w_0.9	d_0.95_t_0.8_w_0.9
1	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
2	0,31	0,31	0,31	0,31	0,31	0,31	0,31	0,31	0,31	0,31
3	0,31	0,31	0,31	0,31	0,31	0,31	0,31	0,31	0,31	0,31
4	0,28	0,21	0,24	0,31	0,21	0,28	0,24	0,21	0,21	0,25
5	0,28	0,21	0,24	0,31	0,21	0,28	0,24	0,21	0,21	0,25
6	0,27	0,19	0,20	0,31	0,20	0,26	0,22	0,19	0,19	0,22
7	0,27	0,19	0,21	0,31	0,20	0,26	0,22	0,19	0,19	0,22
8	0,26	0,20	0,20	0,31	0,20	0,24	0,21	0,20	0,20	0,21
9	0,26	0,20	0,20	0,31	0,20	0,24	0,21	0,20	0,20	0,21
10	0,25	0,20	0,20	0,31	0,20	0,23	0,21	0,20	0,20	0,21
11	0,25	0,20	0,20	0,31	0,20	0,23	0,21	0,20	0,20	0,21
12	0,25	0,20	0,20	0,31	0,20	0,23	0,21	0,20	0,20	0,20
13	0,25	0,20	0,20	0,31	0,20	0,23	0,21	0,20	0,20	0,20
14	0,25	0,20	0,20	0,31	0,20	0,23	0,21	0,20	0,20	0,20
15	0,25	0,20	0,20	0,31	0,20	0,23	0,21	0,20	0,20	0,20
16	0,25	0,20	0,20	0,31	0,20	0,23	0,21	0,20	0,20	0,20
17	0,25	0,20	0,20	0,31	0,20	0,23	0,21	0,20	0,20	0,20
18	0,25	0,20	0,20	0,31	0,20	0,23	0,21	0,20	0,20	0,21
19	0,25	0,20	0,20	0,31	0,20	0,23	0,21	0,20	0,20	0,21
20	0,25	0,20	0,20	0,31	0,20	0,23	0,21	0,20	0,20	0,21

Table 7.2: The table shows each iteration's precision values of 20 spreading activation iterations. The columns represent different combinations of activation decay (d), threshold level (t) and relation weights (w). Since not all possible combinations (27 in total) result in positive values, those zero-values are blinded out. The table illustrates the precision performance of the spreading activation task identification approach on the KnowSe dataset. The colored cells reflect the trend of the precision performance per column. The table indicates that the precision decreases at iteration 4. This means that at iteration 4 the proportion of task instances belonging to the same task model as the origin task instance decreases.

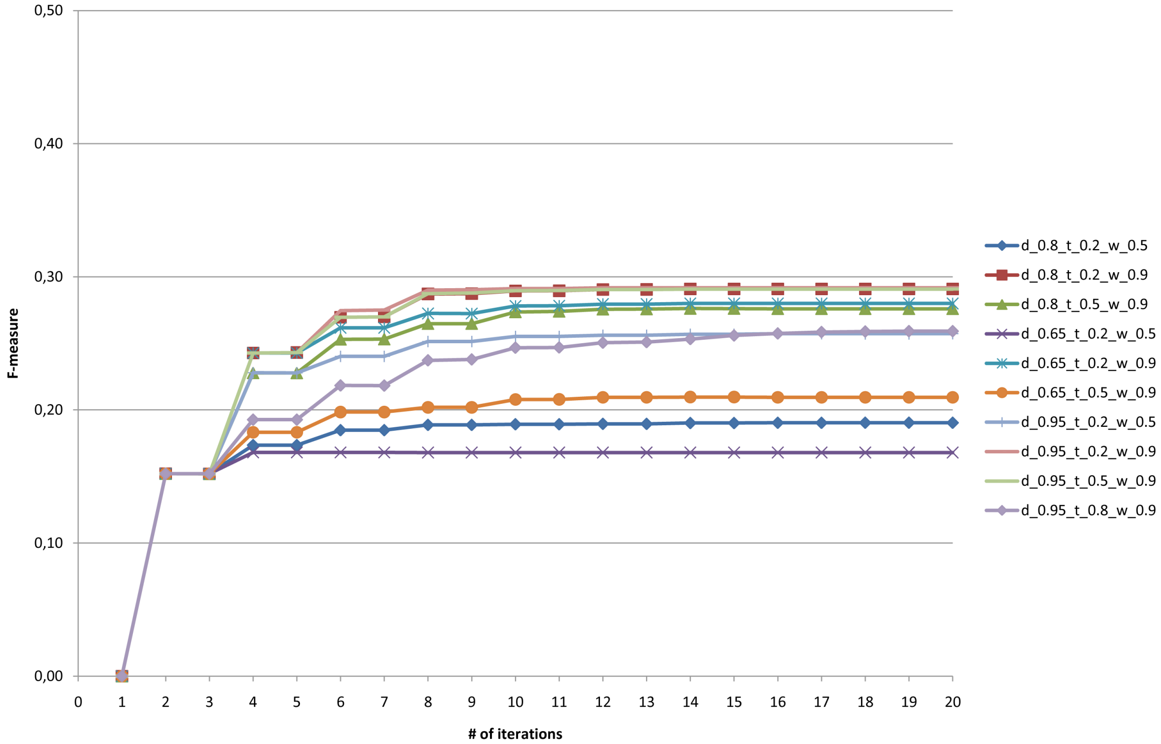

Figure 7.1: This figure shows the trend of the F_1-measure over 20 iterations for different configuration combinations applied on the Knowse dataset. The harmonic mean shows the highest slopes at iteration 2 and 4. After iteration 4 the performance increases only marginally and not at all configuration combinations.

7.5 Evaluation Setup 2 - Task Detection with WT1 Dataset

This second evaluation setup was the outcome of a laboratory experiment together with students of Graz University of Technology. In particular the students were participants of the study programmes of Software Development and Business Management respectively Telematics. The intention was the selection of a domain that is as homogeneous as possible. Seven different tasks types (task models) have been performed, whereby each task was performed several times. Altogether 133 tasks (task instances) have been performed. The task models included the following activieties of the participants:

1. Registration for an examination: The online system of Graz University of Technoloy is utilized for the registration of the students to different examinations. The assignment was to visit the university's website, to login and to register for a specified exam.

2. Finding course dates: The purpose was the discovery of actual course dates for a specific lecture.

3. Reserve a book in the university's library: Students were requested to perform the task of making a reservation for a specific book.

4. Course registration: This task is very similar to the first one. The students had to register for a predetermined lecture.

5. Algorithm programming: The students have been instructed to write a small program that computes the factorial of a given number.

6. Prepare a scientific talk: The goal of this task was the creation of a presentation with Microsoft Powerpoint with the support of online available resources.

7. Plan a study trip: The objective was to plan a study trip including the transportation as well as the accomodation.

An analysis of the generated results show the same behaviour as the first dataset (KnowSe dataset) concerning the performance measure changes on every second iteration. This makes sense since the underlying ontology is the same and in this ontology there are no object properties defined that could connect two instances of the task concept.

Table 7.3 illustrates the recall performance measures on the WT1 dataset. The colored cells reflect the trend of the recall for each configuration combinantion represented by a single column. The overall recall for all combinations is over 0.54 after the second iteration. That means that after two iterations 54 percent of all tasks of the same task model are found by the spreading activation approach.

Table 7.4 illustrates the precision performance of the spreading activation task identification approach on the WT1 dataset. The colored cells reflect the trend for each configuration combinantion represented by a single column. The table shows a decreasing trend of the precision performance for all configuartion combinations starting at the fourth iteration.

Figure 7.2 shows the trend of the harmonic mean over 20 iterations for different configuration combinations. The figure illustrates a downtrend after the 2 second iteration. That means that the best values are achieved when stopping the spreading activation algorithm after the second iteration.

iteration	d_0.8_t_0.2_w_0.5	d_0.8_t_0.2_w_0.9	d_0.8_t_0.5_w_0.9	d_0.65_t_0.2_w_0.5	d_0.65_t_0.2_w_0.9	d_0.65_t_0.5_w_0.9	d_0.95_t_0.2_w_0.5	d_0.95_t_0.2_w_0.9	d_0.95_t_0.5_w_0.9	d_0.95_t_0.8_w_0.9
1	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
2	0,54	0,54	0,54	0,54	0,54	0,54	0,54	0,54	0,54	0,54
3	0,54	0,54	0,54	0,54	0,54	0,54	0,54	0,54	0,54	0,54
4	0,59	0,71	0,68	0,57	0,71	0,60	0,68	0,71	0,71	0,63
5	0,59	0,71	0,68	0,57	0,71	0,60	0,68	0,71	0,71	0,63
6	0,62	0,73	0,71	0,57	0,73	0,64	0,70	0,74	0,73	0,67
7	0,62	0,73	0,71	0,57	0,73	0,64	0,70	0,74	0,73	0,67
8	0,63	0,74	0,72	0,57	0,74	0,65	0,71	0,74	0,74	0,67
9	0,63	0,74	0,72	0,57	0,74	0,65	0,71	0,74	0,74	0,67
10	0,63	0,74	0,72	0,57	0,74	0,65	0,71	0,74	0,74	0,67
11	0,63	0,74	0,72	0,57	0,74	0,65	0,71	0,74	0,74	0,67
12	0,63	0,74	0,72	0,57	0,74	0,65	0,71	0,74	0,74	0,67
13	0,63	0,74	0,72	0,57	0,74	0,65	0,71	0,74	0,74	0,67
14	0,63	0,74	0,72	0,57	0,74	0,65	0,71	0,74	0,74	0,67
15	0,63	0,74	0,72	0,57	0,74	0,65	0,71	0,74	0,74	0,67
16	0,63	0,74	0,72	0,57	0,74	0,65	0,71	0,74	0,74	0,67
17	0,63	0,74	0,72	0,57	0,74	0,65	0,71	0,74	0,74	0,67
18	0,63	0,74	0,72	0,57	0,74	0,65	0,71	0,74	0,74	0,67
19	0,63	0,74	0,72	0,57	0,74	0,65	0,71	0,74	0,74	0,67
20	0,63	0,74	0,72	0,57	0,74	0,65	0,71	0,74	0,74	0,67

Table 7.3: The table shows each iteration's recall values of 20 spreading activation iterations. The columns represent different combinations of activation decay (d), threshold level (t) and relation weights (w). Since not all possible combinations (27 in total) result in positive values, those zero-values are blinded out. The table illustrates the first change of the recall performance measure at the second iteration. This seems to be plausible since starting from a task instance the spreading activation approach will not find an other task instance at the first cycle. The colored cells reflect the trend of the recall for each column. For example, the second column indicates a positive progression until iteration 8. That means that after iteration 8 no more task instances are found that have the same task model in common with the starting task instance.

iteration	d_0.8_t_0.2_w_0.5	d_0.8_t_0.2_w_0.9	d_0.8_t_0.5_w_0.9	d_0.65_t_0.2_w_0.5	d_0.65_t_0.2_w_0.9	d_0.65_t_0.5_w_0.9	d_0.95_t_0.2_w_0.5	d_0.95_t_0.2_w_0.9	d_0.95_t_0.5_w_0.9	d_0.95_t_0.8_w_0.9
1	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
2	0,26	0,26	0,26	0,26	0,26	0,26	0,26	0,26	0,26	0,26
3	0,26	0,26	0,26	0,26	0,26	0,26	0,26	0,26	0,26	0,26
4	0,21	0,15	0,16	0,22	0,15	0,20	0,16	0,15	0,15	0,18
5	0,21	0,15	0,16	0,22	0,15	0,20	0,16	0,15	0,15	0,18
6	0,17	0,14	0,14	0,21	0,14	0,16	0,14	0,14	0,14	0,15
7	0,17	0,14	0,14	0,21	0,14	0,16	0,14	0,14	0,14	0,15
8	0,16	0,14	0,14	0,20	0,14	0,16	0,14	0,14	0,14	0,15
9	0,16	0,14	0,14	0,20	0,14	0,16	0,14	0,14	0,14	0,15
10	0,16	0,14	0,14	0,20	0,14	0,15	0,14	0,14	0,14	0,15
11	0,16	0,14	0,14	0,20	0,14	0,15	0,14	0,14	0,14	0,15
12	0,16	0,14	0,14	0,20	0,14	0,15	0,14	0,14	0,14	0,15
13	0,16	0,14	0,14	0,20	0,14	0,15	0,14	0,14	0,14	0,15
14	0,16	0,14	0,14	0,20	0,14	0,15	0,14	0,14	0,14	0,15
15	0,16	0,14	0,14	0,20	0,14	0,15	0,14	0,14	0,14	0,15
16	0,16	0,14	0,14	0,20	0,14	0,15	0,14	0,14	0,14	0,15
17	0,16	0,14	0,14	0,20	0,14	0,15	0,14	0,14	0,14	0,15
18	0,16	0,14	0,14	0,20	0,14	0,15	0,14	0,14	0,14	0,15
19	0,16	0,14	0,14	0,20	0,14	0,15	0,14	0,14	0,14	0,15
20	0,16	0,14	0,14	0,20	0,14	0,15	0,14	0,14	0,14	0,15

Table 7.4: The table shows each iteration's precision values of 20 spreading activation iterations. The columns represent different combinations of activation decay (d), threshold level (t) and relation weights (w). Since not all possible combinations (27 in total) result in positive values, those zero-values are blinded out. The table illustrates the precision performance of the spreading activation task identification approach on the WT1 dataset. The colored cells reflect the trend of the precision performance per column. The table indicates that the precision decreases at iteration 4. This means that at iteration 4 the proportion of task instances belonging to the same task model as the origin task instance decreases.

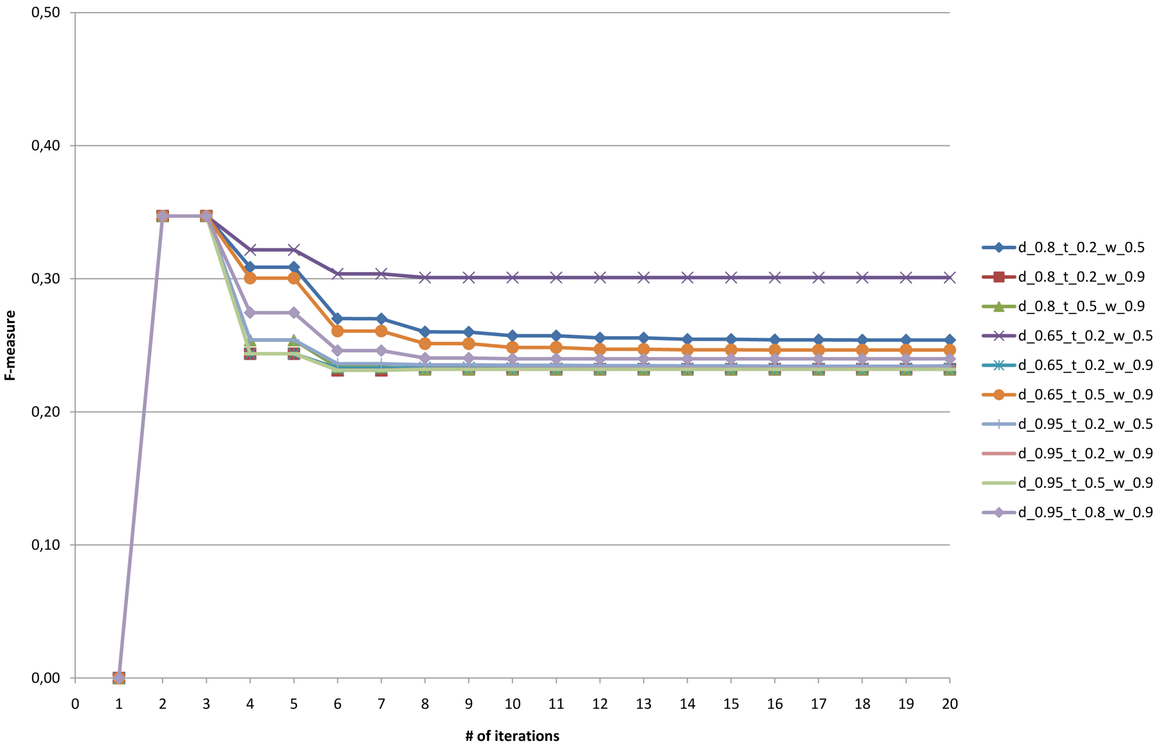

Figure 7.2: This figure shows the trend of the F_1-measure over 20 iterations for different configuration combinations applied on the WT1 dataset. The harmonic mean shows a positive slope at iteration 2. After iteration 2 the performance decreases for all configuration combinations.

7.6 Evaluation Setup 3 - Task Detection with WT2 Dataset

The data for the last evaluaton setup come from a laboratory experiment very similar to the second one. This is also reflected by the participants of the experiment. These participants were also students of Software Development and Business Management respectively Telematics. Each student had to complete a set of tasks, whereas every participant had to do it in a different order. In total 139 tasks (task instances) have been performed by 18 partitipants. The performed tasks (task models) can be described as follows:

1. Task Assess: Finding out if a specific student has to tuition fees for a given term is the goal of this task. As a starting point a website is given to the participant.
2. Task Assign: The task is to assign available study assistants to a specific amount of students registered to a programming course. For that purpose some restrictions are contained in the instruction.
3. Task Classify: A couple of well known computer scientific terms are given. The challenge is to put them into a given classification schema.
4. Task Design: The purpose of this taks is the conceptual design of an elevator system.
5. Task Diagnose: The participant has to find the origin of a malfunction in a program. The source code of the program is provided to the student and the results have to be stored on the computer.
6. Task Plan: The tasks involves the planning of a software project. More precisely, a document management system has to be developed by planning all contained activities.

7. Task Predict: The task is trying to predict three questions for the next exam of economics at Graz University of Technology. Therefore some historical exams are provided to the participants by a web link.

8. Task Schedule: The task's purpose is the scheduling of a software project. A timeslot is provided to the students and the given activities and deadlines have to scheduled inside that time period.

An analysis of the generated results show the same behaviour as the first and the second dataset (KnowSe and WT1 dataset) concerning the performance measure changes on every second iteration. Since there are no object properties defined in the user interaction context ontology two instances of the task concept cannot be connceted with each other.

Table 7.5 illustrates the recall performance measures on the WT2 dataset. The colored cells reflect the trend of the recall for each configuration combinantion represented by a single column. The overall recall for all combinations is over 0.21 after the second iteration. That means that after two iterations 21 percent of all tasks of the same task model are found by the spreading activation approach. Furthermore it can be determined that after 8 iterations the recall reaches a maxiumum of 0.38. That means that only 38 percent of all task of the same task model can be identified by the spreading activation approach.

Table 7.6 illustrates the precision performance of the spreading activation task identification approach on the WT2 dataset. The colored cells reflect the trend for each configuration combinantion represented by a single column. The table shows a decreasing trend of the precision performance for all configuartion combinations starting at the fourth iteration.

Figure 7.3 shows the trend of the harmonic mean over 20 iterations for different configuration combinations. The figure illustrates a downtrend after the 2 second iteration. That means that the best values are achieved when stopping the spreading activation

iteration	d_0.8_t_0.2_w_0.5	d_0.8_t_0.2_w_0.9	d_0.8_t_0.5_w_0.9	d_0.65_t_0.2_w_0.5	d_0.65_t_0.2_w_0.9	d_0.65_t_0.5_w_0.9	d_0.95_t_0.2_w_0.5	d_0.95_t_0.2_w_0.9	d_0.95_t_0.5_w_0.9	d_0.95_t_0.8_w_0.9
1	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
2	0,21	0,21	0,21	0,21	0,21	0,21	0,21	0,21	0,21	0,21
3	0,21	0,21	0,21	0,21	0,21	0,21	0,21	0,21	0,21	0,21
4	0,24	0,29	0,26	0,23	0,29	0,24	0,26	0,29	0,29	0,25
5	0,24	0,29	0,26	0,23	0,29	0,24	0,26	0,29	0,29	0,25
6	0,24	0,33	0,29	0,23	0,32	0,24	0,28	0,34	0,33	0,26
7	0,24	0,33	0,29	0,23	0,32	0,24	0,28	0,34	0,33	0,26
8	0,24	0,36	0,30	0,23	0,34	0,25	0,28	0,37	0,36	0,27
9	0,24	0,36	0,30	0,23	0,34	0,25	0,28	0,37	0,36	0,27
10	0,24	0,37	0,30	0,23	0,34	0,25	0,28	0,38	0,37	0,27
11	0,24	0,37	0,30	0,23	0,34	0,25	0,28	0,38	0,37	0,27
12	0,24	0,37	0,30	0,23	0,34	0,25	0,28	0,38	0,37	0,27
13	0,24	0,37	0,30	0,23	0,34	0,25	0,28	0,38	0,37	0,27
14	0,24	0,37	0,30	0,23	0,34	0,25	0,28	0,38	0,37	0,27
15	0,24	0,37	0,30	0,23	0,34	0,25	0,28	0,38	0,37	0,27
16	0,24	0,37	0,30	0,23	0,34	0,25	0,28	0,38	0,37	0,27
17	0,24	0,37	0,30	0,23	0,34	0,25	0,28	0,38	0,37	0,27
18	0,24	0,37	0,30	0,23	0,34	0,25	0,28	0,38	0,37	0,27
19	0,24	0,37	0,30	0,23	0,34	0,25	0,28	0,38	0,37	0,27
20	0,24	0,37	0,30	0,23	0,34	0,25	0,28	0,38	0,37	0,27

Table 7.5: The table shows each iteration's recall values of 20 spreading activation iterations. The columns represent different combinations of activation decay (d), threshold level (t) and relation weights (w). Since not all possible combinations (27 in total) result in positive values, those zero-values are blinded out. The table illustrates the first change of the recall performance measure at the second iteration. This seems to be plausible since starting from a task instance the spreading activation approach will not find an other task instance at the first cycle. The colored cells reflect the trend of the recall for each column. For example, the second column indicates a positive progression until iteration 10. That means that after iteration 10 no more task instances are found that have the same task model in common with the starting task instance.

iteration	d_0.8_t_0.2_w_0.5	d_0.8_t_0.2_w_0.9	d_0.8_t_0.5_w_0.9	d_0.65_t_0.2_w_0.5	d_0.65_t_0.2_w_0.9	d_0.65_t_0.5_w_0.9	d_0.95_t_0.2_w_0.5	d_0.95_t_0.2_w_0.9	d_0.95_t_0.5_w_0.9	d_0.95_t_0.8_w_0.9
1	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
2	0,32	0,32	0,32	0,32	0,32	0,32	0,32	0,32	0,32	0,32
3	0,32	0,32	0,32	0,32	0,32	0,32	0,32	0,32	0,32	0,32
4	0,26	0,15	0,18	0,29	0,15	0,25	0,18	0,15	0,16	0,22
5	0,26	0,15	0,18	0,29	0,15	0,25	0,18	0,15	0,16	0,22
6	0,23	0,14	0,16	0,27	0,14	0,22	0,17	0,13	0,14	0,19
7	0,23	0,14	0,16	0,27	0,14	0,22	0,17	0,13	0,14	0,19
8	0,22	0,13	0,16	0,27	0,14	0,21	0,17	0,13	0,13	0,18
9	0,22	0,13	0,16	0,27	0,14	0,21	0,17	0,13	0,13	0,18
10	0,22	0,13	0,16	0,27	0,14	0,20	0,17	0,13	0,13	0,18
11	0,22	0,13	0,16	0,27	0,14	0,20	0,17	0,13	0,13	0,18
12	0,22	0,13	0,16	0,27	0,14	0,20	0,17	0,13	0,13	0,18
13	0,22	0,13	0,16	0,27	0,14	0,20	0,17	0,13	0,13	0,18
14	0,22	0,13	0,16	0,27	0,14	0,20	0,17	0,13	0,13	0,18
15	0,22	0,13	0,16	0,27	0,14	0,20	0,17	0,13	0,13	0,18
16	0,22	0,13	0,16	0,27	0,14	0,20	0,17	0,13	0,13	0,18
17	0,22	0,13	0,16	0,27	0,14	0,20	0,17	0,13	0,13	0,18
18	0,22	0,13	0,16	0,27	0,14	0,20	0,17	0,13	0,13	0,18
19	0,22	0,13	0,16	0,27	0,14	0,20	0,17	0,13	0,13	0,18
20	0,22	0,13	0,16	0,27	0,14	0,20	0,17	0,13	0,13	0,18

Table 7.6: The table shows each iteration's precision values of 20 spreading activation iterations. The columns represent different combinations of activation decay (d), threshold level (t) and relation weights (w). Since not all possible combinations (27 in total) result in positive values, those zero-values are blinded out. The table illustrates the precision performance of the spreading activation task identification approach on the WT2 dataset. The colored cells reflect the trend of the precision performance per column. The table indicates that the precision decreases at iteration 4. This means that at iteration 4 the proportion of task instances belonging to the same task model as the origin task instance decreases.

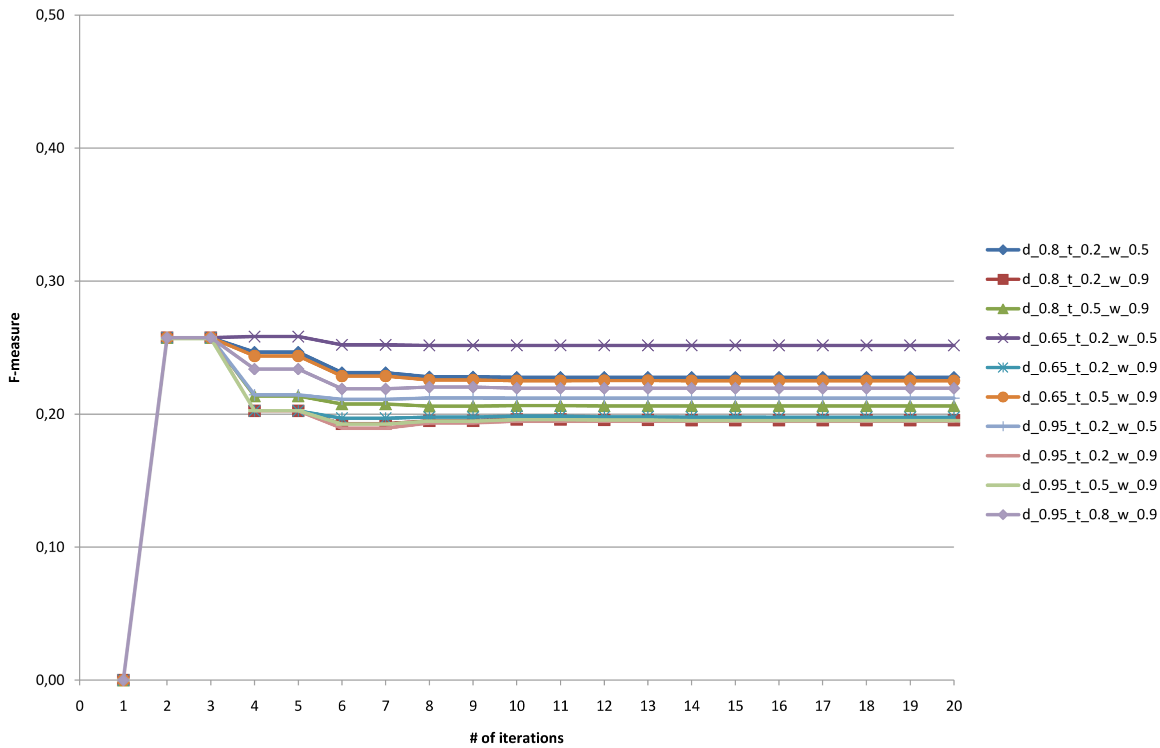

Figure 7.3: This figure shows the trend of the F_1-measure over 20 iterations for different configuration combinations applied on the WT2 dataset. The harmonic mean shows the highest slopes at iteration 2 and 4. After iteration 4 the performance increases only marginally and not at all configuration combinations.

algorithm after the second iteration.

7.7 Configuration of the network

Evaluation setup 1 consists of over one million statements, which are devided into 396.906 statements containing object properties and 688.065 statements containing datatype properties. Evaluation setup 2 contains about half a million and setup 3 over 1.3 million statements. Since a first evaluation cycle showed some performance problems in terms of out of memory errors the size of the spreading activation graph had to be reduced. For this reason all datatype properties have been filtered out from the evaluation. The reason for this lies in the fact, that datatype properties express an relationship between an individual and a data literal. Data literals represent end points in the spreading activation graph and have no influence on the results. Due to the number of statements a consideration of all statements was not possible. For the configuration of the network an iterative approach fitted the size problem. Instances of events and event blocks which have been described in Section 2.2.1 were not considered for the construction of the spreading activation network since they would have a negative impact on the performance of the algorithm and in addition these concepts have no effect on the results. Furthermore instances of the task model concept were also not considered for the construction of the network structure since connections between task and task model instances would distort the results.

7.8 Identification of similar tasks

The main goal of this work is the identifacation of tasks similar to the current task. As the determination of the current task is not part of this thesis, each executed task, one after another, has been used as the current task. Tables 7.1, 7.3 and 7.5 illustrate recall performance values up to 0.74 depending on the used dataset. These deviations show that the developed spreading activation approach is not able to find all similar task. This means that the structure of the associative network respectively the algorithm is insufficient to find all similar task.

The precision performance values show in contrary to the recall more evenly values up to 0.32. That means that around one third and one fourth of all identified task were correct (belong to the same task model as the origin task instance).

7.9 Determination of concepts and properties

The user interaction context ontology comprises 107 concepts and 281 properties. A structural analysis reveals that the relations between tasks are utilizing different types of concepts and properties. This means that by consideration of specific concepts and properties the propagation of activation inside the network is not affected. The goal is to identify those concepts and properties with a substantial contribute to the results. The identification of relevant tasks is considered as the search for instances of the concept task in the user interaction context model. In the user interaction context ontology the concept task is modeled as there can be relations to task models, event blocks and events. Instances of the concept resource are modeled as direct relations to task instances. That means that instances of the concepts `TaskModel`, `Event` and `EventBlock` are not nescessary for the identification of entities. Furthermode all datatype properties have been excluded from the evaluation. On the one hand they had to be removed due to performance issues and on the other hand these properties are not able to build up network structures between objects (concenpt instances) as explained in Section 5.3.1.

7.10 Required number of iterations

The results achieved with three different evaluation setups have a common trend. Between eight and ten iterations the recall reaches its peak. The precision, on the other hand, decreases dramatically from the third iteration. It depends on the use case, how many iterations are advisable. If the focus is on a large amount of retrieved entities with a high proportion of non-relevant data, a larger number of iterations would be recommended. If the user focuses on a few, but useful entities then two iterations would be the right number.

8 Conclusion and Future Work

This thesis introduces and evaluates a spreading activation based approach to identify entities of a user interaction context model. In order to enable better work and learn support the graph structure of the user interaction context ontology is used to apply the graph traversal algorithm on it. This final chapter fill reflect on the defined goals of the first chapter and will assess the evaluation to judge the achievement of objectives.

8.1 Determination of concepts and properties

To set up the spreading activation graph in order to identify relevant tasks, concepts and properties have to be determined. These concepts and properties can be considered as spreading activation constraints. To determine the concepts and properties various aspects have to be taken into consideration. One the one hand a huge amount of user interaction data reaches the limits concerning CPU and memory consumtion. It follows that the spreading activation graph has to be kept as small as possible. One the other hand the essential information concerning the user interactions need not be lost.

For the identification of the essential concepts a manual analysis of the user interaction context ontology has been performed. Particular attention was paid to the very often used event and event block concepts. It was noticed that all informations used by the spreading activation approach are already existent in the task concepts and their relations. Due to the removal of those two concept types and all of their concept instances from the spreading activation graph the graph could be kept small. Moreover the concept instances of the task model concept also had to be removed from the network structure since connections between task and task model instances would distort the results.

The determination of required properties was performed as follows: The graph structure of laboratory datasets showed that a multitude of nodes in the graph structure where not further connected to other nodes. A detailed analysis revealed that many of those nodes where datatype objects connected to concept instances by datatype properties. Datatype objects can be considered as end nodes in a network structure where activation energy cannot spread further. For this reason all datatype properties of the user interaction context ontology have been removed from the network structure.

8.2 Required number of iterations

Another goal of this thesis was the evaluation of required number of spreading activation iterations. This means it has to be determined hove many iterations are nescessary in order to identify relevant tasks. A relevant task in this case is a task that belongs to the the same task model as the origin/current task. In this thesis the number of required iterations has been determined empirically by utilizing three different testsets and calculating different performance measures. Recall, precision and the harmonic mean (F_1-measure) have been choosen. Between eight and ten iterations the recall reaches its peak. The precision, on the other hand, decreases dramatically from the third iteration. In concrete terms this means that the quantity of the relevant tasks is increasing until iteration 10. At the same time the quality of the results decreases rapidly. For this reason the harmonic mean has been calculated in addition. The harmonic mean shows decreasing trend after the second iteration for 2 datasets, but an increasing trend for the first dataset. The reason for these divergent results are the bad recall results of the first dataset. Within this thesis the reason for this bad performance of the first dataset could not be determined. However, the concept instances of the first dataset seemed to be not highly interconnected in comparison with the other two datasets. Summarizing it can be stated that if a high quality level of results should be achieved the number of iterations should be kept at two iterations at all. If the focus is on a large amount of retrieved entities with a high proportion of non-relevant data, a larger number of 8 iterations would be recommended. All further iterations would have no mentionable impact on the results.

8.3 Spreading activation specific attributes

The evaluation of spreading activation specific attributes such as activation decay, threshold level, and relation weights are evaluated in order to retrieve relevant tasks. By the assignment of these attributes the control of activation propagation inside the spreading activation graph is enabled. Since these attributes depend on the structure of the graph as well as on the specific use case there are no standardized values available. Therefore the values of these attributes are empirically determined. For this reason the three testsets and the calculated performance measures are used for the determination of the best configuration of the spreading activation specific attributes. If the qualitiy of the results has the main priority, then all testsets verify that the following combination of attributes is the best:

Activation decay: 0.65

Threshold level: 0.2

Relation weights: 0.5

8.4 Suitability and future work

The evaluation of the spreading activation approach deleveloped in this thesis showed some areas for improvement. The precision of the algorithm showed an overall bad performance. That means that the quality of the achieved spreading activation results were not as good as expected.

In order to validate the performance of the spreading activation approach, different scenarios will have to be evaluated. The different scenarios include different tasks and different situations in order to make a statement about the generalisability of the received results. In addition it should be taken into consideration to adapt the ontology and sensor design in such a way that concept instances retrieve a higher degree of connections, e.g. adding additional concept instances (tagging) that are applied manually by the user.

Bibliography

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

[Alani et al., 2002] Alani, H., O'Hara, K., and Shadbolt, N. (2002). Ontocopi: Methods and tools for identifying communities of practice. In *IFIP 17th World Computer Congress - TC12 Stream on Intelligent Information Processing*. Kluwer.

[Anderson, 1983] Anderson, J. R. (1983). A spreading activation theory of memory. *Journal of Verbal Learning and Verbal Behavior*, 22:261–295.

[Anderson and Bower, 1979] Anderson, J. R. and Bower, G. H. (1979). *Human associative memory*. Lawrence Erlbaum, Hillsdale, N.J.

[Bellone, 2011] Bellone, J. (2011). BirdEye/RaVis Library.

[Berger, 2003] Berger, H. (2003). *Activation on the Move: Adaptive Information Retrieval via Spreading Activation*. PhD thesis, Vienna University of Technology.

[Budzik et al., 2001] Budzik, J., Hammond, K., and Birnbaum, L. (2001). Information access in context. *Knowledge based systems*, 14(1-2):37–53.

[Budzik and Hammond, 2000] Budzik, J. and Hammond, K. J. (2000). User interactions with everyday applications as context for just-in-time information access. In *IUI '00: Proceedings of the 5th international conference on Intelligent user interfaces*, pages 44–51, New York, NY, USA. ACM.

[Bush and Wang, 1945] Bush, V. and Wang, J. (1945). As we may think. *Atlantic Monthly*, 176:101–108.

[Chirita et al., 2006] Chirita, P.-A., Costache, S., Nejdl, W., and Paiu, R. (2006). Beagle++ : Semantically enhanced searching and ranking on the desktop. In *ESWC '06: 3rd European Semantic Web Conference*.

[Cohen and Kjeldsen, 1987] Cohen, P. R. and Kjeldsen, R. (1987). Information retrieval by constrained spreading activation in semantic networks. *Inf. Process. Manage.*, 23(4):255–268.

[Corporation, 2010] Corporation, M. (2010). Windows Search 4.0.

[Crestani, 1997] Crestani, F. (1997). Application of spreading activation techniques in information retrieval. *Artif. Intell. Rev.*, 11(6):453–482.

[Dey, 2001] Dey, A. K. (2001). Understanding and using context. *Personal Ubiquitous Comput.*, 5(1):4–7.

[Fellbaum, 1998] Fellbaum, C., editor (1998). *WordNet: An Electronic Lexical Database.* MIT Press, Cambridge, MA.

[Foundation, 2010] Foundation, A. S. (2010). Apache Lucene.

[Foundation, 2011] Foundation, T. E. (2011). Jetty Web Server.

[Gruber, 1995] Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human Computer Studies*, 43:907–928.

[Guyon et al., 2006] Guyon, I., Gunn, S., Nikravesh, M., and Zadeh, L. A. (2006). *Feature Extraction: Foundations and Applications (Studies in Fuzziness and Soft Computing)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[Inc., 2010] Inc., A. (2010). Spotlight.

[Katifori et al., 2010] Katifori, A., Vassilakis, C., and Dix, A. (2010). Ontologies and the brain: Using spreading activation through ontologies to support personal interaction. *Cognitive Systems Research*, 11(1):25–41. Brain Informatics.

[Katifori et al., 2008] Katifori, A., Vassilakis, C., and Dix, A. J. (2008). Using spreading activation through ontologies to support personal information management. In *CSKGOI*.

[Kjeldsen and Cohen, 1987] Kjeldsen, R. and Cohen, P. R. (1987). The evolution and performance of the grant system. *IEEE Expert*, 2(2):73–79.

[Liu et al., 2005] Liu, W., Weichselbraun, A., Scharl, A., and Chang, E. (2005). Semi-automatic ontology extension using spreading activation. *j-jukm*, 0(1):50–58.

[Manning et al., 2008] Manning, C. D., Raghavan, P., and Schtze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

[OpenAnzo.org, 2010] OpenAnzo.org (2010). The Open Anzo Project.

[O'Riordan and Sorensen, 1995] O'Riordan, A. and Sorensen, H. (1995). An intelligent agent for high-precision text filtering. In *CIKM '95: Proceedings of the fourth international conference on Information and knowledge management*, pages 205–211, New York, NY, USA. ACM.

[Prud'hommeaux and Seaborne, 2008] Prud'hommeaux, E. and Seaborne, A. (2008). Sparql query language for rdf.

[Quillian, 1968] Quillian, M. R. (1968). Semantic memory. In Minsky, M., editor, *Semantic Information Processing*, pages 216–270. MIT Press.

[Rath et al., 2009] Rath, A., Devaurs, D., and Lindstaedt, S. (2009). UICO: an ontology-based user interaction context model for automatic task detection on the computer desktop. In *Proceedings of the 1st Workshop on Context, Information and Ontologies*, pages 1–10. ACM.

[Rath et al., 2006] Rath, A., Kr
"oll, M., Andrews, K., Lindstaedt, S., Granitzer, M., and Tochtermann, K. (2006). Synergizing standard and ad-hoc processes. *Practical Aspects of Knowledge Management*, pages 267–278.

[Rath, 2010] Rath, A. S. (2010). *User Interaction Context: Studying and Enhancing Automatic User Task Detection on the Computer Desktop via an Ontology-based User Interaction Context Model*. PhD thesis, Graz University of Technology.

[Rijsbergen, 1979] Rijsbergen, C. J. v. (1979). *Information retrieval*. Butterworths, London, 2 edition.

[Rocha et al., 2004] Rocha, C., Schwabe, D., and Aragao, M. P. (2004). A hybrid approach for searching in the semantic web. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 374–383, New York, NY, USA. ACM.

[Salber et al., 1999] Salber, D., Dey, A., and Abowd, G. (1999). The context toolkit: Aiding the development of context-enabled applications. In *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit*, pages 434–441. ACM.

[Sauermann et al., 2006] Sauermann, L., Dengel, A., Van Elst, L., Lauer, A., Maus, H., and Schwarz, S. (2006). Personalization in the EPOS project. In *Proceedings of* *the International Workshop on Semantic Web Personalization, Budva, Montenegro*, pages 42–52. Citeseer.

[Scheir, 2008] Scheir, P. (2008). *Assoziative Suche fr das Semantic Web*. PhD thesis, Graz University of Technology.

[Schilit and Theimer, 1994] Schilit, B. and Theimer, M. (1994). Disseminating active map information to mobile hosts. *IEEE network*, 8(5):22–32.

[Schumacher et al., 2008] Schumacher, K., Sintek, M., and Sauermann, L. (2008). Combining fact and document retrieval with spreading activation for semantic desktop search. In *ESWC '08: 5th European Semantic Web Conference*, pages 569–583.

[Sharifian and Samani, 1997] Sharifian, F. and Samani, R. (1997). Hierarchical spreading of activation. *Proc. of the Int'l Conference on Language, Cognition, and Interpretation*, pages 1–10.

[Stanford Center for Biomedical Informatics Research, 2010] Stanford Center for Biomedical Informatics Research (2010). The Protégé Ontology Editor and Knowledge Acquisition System.

[Troussov A. and D., 2008] Troussov A., Sogrin M., J. J. and D., B. (2008). Mining socio-semantic networks using spreading activation technique. In *TRIPLE-I '08*.

[Ward et al., 1997] Ward, A., Jones, A., and Hopper, A. (1997). A new location technique for the active office. *IEEE Personal Communications*, 4(5):42–47.