A Semantic Map as Basis for the Decision Process in the www Navigation

Hartwig Hochmair, Andrew U. Frank

Institute for Geoinformation, Technical University Vienna
Gußhausstr. 27-29, A-1040 Vienna, Austria
{hochmair,frank}@geoinfo.tuwien.ac.at
Tel.: (+43) (0)1 58801-{12715,12710}
Fax.: (+43) (0) 1 58801-12799

Abstract. In the physical world, decision making in common navigation strategies is based on a mental map which includes a mental representation of geometric features such as distances and directions between places. We propose that decision making in the web is also based on a mental map. Contrary to the physical world navigation, the mental map used for web navigation describes a part of the agent's epistemology of the world, hence consists of semantic relations between concepts and includes hardly any geometrical features. We focus on the navigation situation where the detailed web structure is unknown to the agent before the navigation, therefore the agent's decisions are completely based on his semantic mental map. We give a potential structure of the agent's mental map, simulated using WordNet, and simulate the agent's decision making process during the web navigation. The simulation of the strategies allows to assess existing web environments with regard to ease of navigability.

Keywords. semantic map, web navigation strategy, decision process, epistemology, web searching

1 Introduction

1.1 A Mental Map for Navigation Tasks

Navigation towards an unknown goal is a common human activity. Many of the proposed navigation strategies in the literature are based on a mental representation or cognitive map of the environment. A cognitive map can be constructed through environmental observations combined with locomotional information during the exploration (Kuipers 1978), through reading an external map (Sholl 1987; Hochmair to app.) or through communication (Peuquet 1998).

Hochmair, H. and A. Frank (2001). A Semantic Map as Basis for the Decision Process in the www Navigation. Conference on Spatial Information Theory. D. R. Montello (Ed.). Morro Bay, California, USA, Springer: 173-188.

We propose that a web navigating agent uses a mental map for his navigation strategy, too. The agent navigates towards a novel web page and has a rough idea of the content of the target web page. The agent builds a semantic map around the content of the web site that he is searching for. The semantic map consists of several fields, of which certain elements together define the content of the web site. Because of navigating towards a novel goal, the detailed web structure of the web domain is unknown to the agent before the navigation. This means that the agent's mental map itself does not describe the linked positions of web pages. For semantic web navigation, such information is not needed. Directions and Euclidean distances, and path strategies are irrelevant for the web navigation as the web itself is not spatial (Svensson 1998) in the sense of direct bodily locomotion experience.

1.2 Searching in the Web Space

Browsing and searching are common terms for web users. Literature shows a wide variation in the meaning of these terms. Marchionini (1995) distinguishes between direct browsing, semidirected browsing, and undirected browsing. The author also proposes a model of the information seeking process which is composed of eight subprocesses. Ellis et al. propose six categories of information seeking activities for general media: starting, chaining, browsing, differentiating, monitoring and extracting (Ellis and Haugan 1997).

Our proposed web navigation strategy is related to the first of the categories proposed by Marchionini, and to the first four categories of seeking activities in (Ellis and Haugan 1997). At the beginning of the web navigation process, the agent determines a starting point in a web domain, possibly found using a search engine. During the navigation process, the agent is offered a number of links on each visited web site. On the user's computer screen, the links are realized either through colored hyperlink-texts, metaphorical graphics, or icons. If the associated information of a web link is part of the agent's semantic map, the agent is able to determine which of the links is semantically closest to the content of the web page that he is trying to find.

2 The Agent's Semantic Map

2.1 A Glance at Previously Discussed Ontologies in the Literature

A wide range of models to categorize the world is proposed by philosophers. Smith (2001) claims that each scientific field has its own preferred ontology. For the philosopher, ontology is a theoretical construct which is invariant from the used language. Contrary to this, in the world of information systems, such as the web, ontology is a software artifact designed with a specific use and computational environment in mind.

The classical theory uses categories with clear boundaries which are defined by common properties (Lakoff 1987). Wittgenstein (1953) contradicts the classical theory and proposes that not all members of a category share one common property but are united by family resemblances. Conceptual embodiment (Lakoff 1987) claims that the properties of certain categories are a consequence of the nature of the human biological capacities and of the experience made in a social and physical environment. This contradicts the idea that concepts exist independent of a body and experience. In experiments concerning colors and physical categories, Rosch and Mervis (1975) found out asymmetries, so called prototype effects, between members of a category. Membership gradience (Barsalou 1987) describes the idea that some categories have degrees of membership and no clear boundaries. Graded structures are found in a wide range of category types (Rips, Shoben et al. 1973; Smith, Shoben et al. 1974; Lakoff 1987).

2.2 Proposed Structure of the Agent's Semantic Map

For the proposed structure of the agent's semantic map, we use the basic idea of Aristotle's ontology which is based on substance and accident. Substances are things and bodies, accidents are qualities, events, and processes. Substances vary in chemical and physical composition, and are hierarchically structured. Some objects involve both substances and accidental parts, so that objects are partially bearers of accidents. A connection between objects and activity is also given through the idea of affordances (Gibson 1977). Raubal distinguishes between several types of affordances, e.g., social-institutional (Raubal to app.) or action affordances (Raubal and Worboys 1999). The latter describe what things or objects offer people to do with them. Affordances are highly based on an individual's life experience.

Concerning computer interfaces, Norman (1999) distinguishes between physical affordances and perceived affordances, where the physical affordances are a synonym for action affordances. In the computer environment, the computer with its keyboard, display screen, pointing device, and mouse buttons affords pointing, touching, looking, and clicking on every pixel of the screen. In graphical, screen-based interfaces, the designer can control only perceived affordances. Norman explains perceived affordances as displayed elements, e.g., a cursor or an icon, that advertise an affordance. The design of a graphical object on the screen does not 'afford clicking', the object only provides a target and helps the user know where to click. The activity to click on an object on a screen with a pointing device is in Norman's view motivated by cultural conventions, which are conventions shared by a cultural group, and not through affordances of the designed object on the screen itself.

We conclude that a displayed link on the screen provides two separate layers of information to the user:

1st layer: advertises to *click* on the hyperlink or icon (*convention*)

2nd layer: *perceived affordance*, which indicates that the *object* or *content* of the web site behind the link affords a specific action or contains information

As a composition of Aristotle's ontology with substances and accidents, and Gibson's affordance theory, we propose the following fields for the structure of the agent's mental map regarding the web site the agent is searching for (italic font indicates Aristotle's terms):

- action affordances (*events*)
- physical object hierarchy (*substances*)
- attributes (*qualities*)

A fourth field of the mental map is formed by the activities a web user wants to perform in the web. We name the field 'user intended actions'. All fields except the attributes are either structured in a partonomy or taxonomy. The highest hierarchical layer of each field expresses the most general term in a field, elements of a lower layer are either parts of or kinds of the term in an upper hierarchy. An example of a semantic map is shown in **Fig. 1**.

The closer the content of a web page is to the lowest elements of the fields in the semantic map, the higher is the agent's subjective feeling of being close to the target page. Therefore, the agent's goal is to reach the lowest element of each field hierarchy. This mental goal can be represented as a set of elements in the semantic map. When analyzing various web pages, we found out that those elements of the mental goal that are not provided by the content of the actual web page can in most cases be found after clicking the link of the lowest element in the user intended action. This fact allows us to simplify the criterion that determines if the goal has been reached or not: The goal is reached if the link labeled with the lowest element in the field 'user intended actions' ('order' in **Fig. 1**) can be perceived.

User intended actions in the web are not limited to seeking (Ellis and Haugan 1997; Wilson 1997) and browsing (Marchionini 1995) but include all potential activities in the internet, such as order a product online, send an e-mail, or view a city map.

2.3 Why Using a Predefined Ontology

Due to human individual life experience and different semantic maps, it is not possible to model the navigation behavior for each individual human. For this reason we create a prototype agent with a semantic map that is based on WordNet (Miller 1995), a database for the English language. The online application of WordNet can be visited at *http://www.cogsci.princeton.edu/~wn/w3wn.html*. The prototype agent for our model uses a semantic map for the purpose to order sneakers (jogging shoes) in the internet.

WordNet combines features of both a traditional dictionary and a thesaurus. All query results are given in form of synsets (Jones 1986) which describe sets of those words which can replace a particular word in a sentence without changing the way the sentence can be employed. The synsets are connected by a number of relations. Unlike in a thesaurus, the relation between concepts and words in WordNet are made explicit and labeled; users select the relation that guides them from one concept to the next and choose the direction of their navigation in conceptual space. WordNet allows semantic queries between nouns, verbs, and adjectives.

3 Filling the Semantic Map of the Prototype Agent

3.1 Physical Object Hierarchy

A hierarchy of nouns is generated by hyponymy and hypernymy relations in WordNet. Usually a noun has only one hypernym but many hyponyms (Miller 1998). Available semantic relations for nouns among others are:

- *coordinate terms (terms that have the same hypernym, 'sisters')*
- *hypernyms (generic term for a whole class)*
- *hyponyms (generic term used to designate a member of a class)*

To fill the field 'physical object hierarchy' we request the hypernyms of 'shoe', as the physical part of sneakers. The bold terms of the result will be included in the mental map.

shoe
  => **footwear**, footgear
    => **covering**
      => **artifact**, artefact
        => object, **physical object**
          => …

Footwear has two meanings in WordNet (covering and clothing). We make an additional step and find hypernyms for footwear in the sense of clothing (see below). For the field 'physical object hierarchy', we unite the results of the two queries.

footwear
  => **clothing**, clothes, apparel, vesture, wearing apparel, wear
    => **covering**
      => …

3.2 User Intended Actions

Like nouns and adjectives in WordNet, verbs are grouped together as sets of synonyms (synsets). English has far fewer verbs than nouns, and verbs are approximately twice as polysemous as nouns (Fellbaum and Miller 1990).

The elements within the field 'user intended actions' are described by verbs. In WordNet, the user intended action 'order' is not used in the context we use it, therefore, we request the hierarchy of 'order' used as a noun. Combined with the definition of the verb 'trade' in WordNet, we get the hierarchy 'do business' - 'buy' - 'order' for the field 'user intended actions'.

3.3 Action Affordances

The verb 'run' in the sense of doing outside sport is used as a noun (gerundive) in WordNet ('running'). We search for hypernyms of the word *running* in the noun category and start with the term 'track, running', which is listed as one of the meanings of running. We get the following result:

track**, running**
  => track and field
    => sport, athletics
      => diversion, **recreation**
        => …

3.4 Attributes

A physical object can have attributes, e.g., color or size. The attributes are expressed by nouns whereas attribute values are expressed by adjectives or values. Nouns can be said to serve as arguments for attributes. What is realized in WordNet so far, is the connection between attribute nouns and adjectives which express values of that attribute, e.g., between the noun *size* and the adjectives *large* and *small* or between the noun *color* and the adjectives *red*, *yellow*, *green* and so on. WordNet has not implemented adjective-noun pairs so far. This means that the database does not allow to determine which are important attributes of a noun.

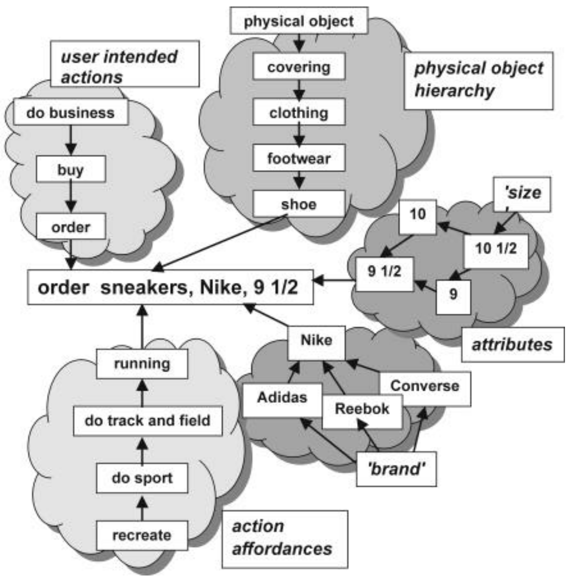

Fig. 1. Mental map around 'order sneakers', based on WordNet

Fig. 1 shows the model of the agent's mental map. It can be seen that we add two attributes (*brand* and *size*) for 'sneakers' to the agent's semantic model. These fields are also hierarchically structured. The agent imagines a subjective 'ideal' value of each attribute to be included in the content of the web page that is being sought. In our example, *Nike* and *9 1/2* are set as the goal values for the two attributes.

4. The Web Environment

4.1 Web Structure

A web environment consists of web pages and connecting links. A web domain has a start page, from which one can move into deeper levels of the domain hierarchy. Links do not mediate any Euclidean and metric information but semantic relations between web pages. Abstracting a web domain, web pages can be represented as nodes, links as edges (**Fig. 2**).

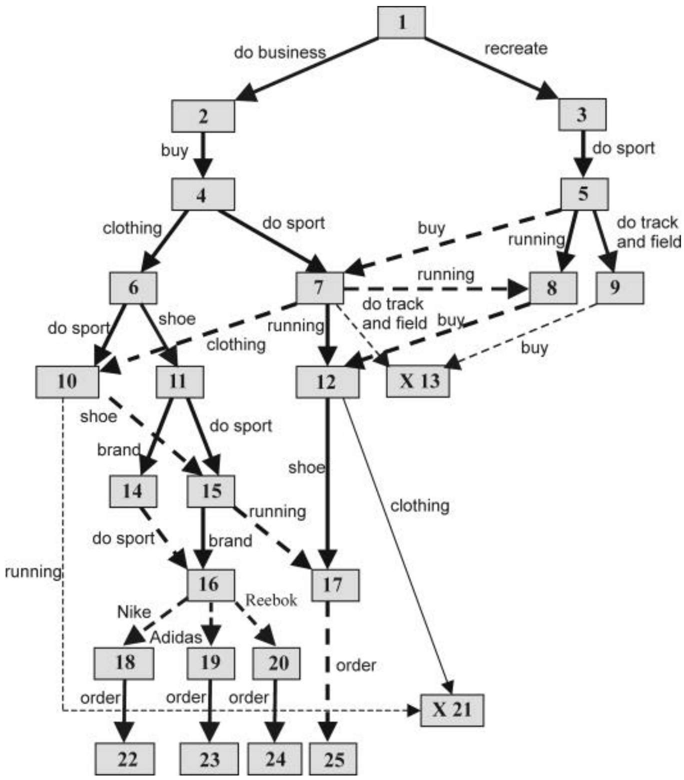

Fig. 2. Part of the link structure of an existing web environment

Each link contains information of the web page to which it leads. Due to this information, a navigating agent gains the cumulative information of all the links along the path from the start page to the actual position in the web. A part of the information offered on the web links is also contained in the agent's semantic map. This overlap is the basis for the agent's navigation decisions.

Web domains are constructed from several categories, where categories themselves can be organized as taxonomies or partonomies. Cross links connect web pages of different categories, and allow web pages to be reached through several paths. For our model we take an existing web environment *(http://www.yahoo.com/)* and use parts of its structure that is visualized in **Fig. 2**.

The visualized graph shows two main categories ('do business' and 'recreate') with their subcategories. Crosslinks between different categories are visualized as dashed arrows. Links which lead to a 'dead end' and require backtracking, are visualized as thin arrows.

4.2. Treatment of Space in the www

Over the past few years, in the internet the number of applications which use at least some concept of spatiality in the way they describe themselves and how they structure interactions, is increasing (Dieberger 1998). Tools like chat rooms, news groups, email lists, and Multi User Dungeons (MUD) are in widespread use. Spatialization of the web can be provided through spatial metaphors (Kuhn 1996) which allow to mimic the real world or certain aspects of it.

Our model proposes a semantic navigation strategy. As navigation in graphical virtual environments and interfaces using spatial metaphors is geometrically but not semantically based, we exclude the discussion of navigation strategies in such spaces.

Many human activities are related to common places. Such common places provide a context for everyday action and a means for identification of the surrounding environment (Jordan, Raubal et al. 1998). The term 'place' in our meaning does not stand for a specific geographic area, such as a state or a town, but rather for a social or federal institution. A part of the activities which are related to common places, can be mapped to the web space. Examples are posting letters from a post office *(send an e-mail)*, buying goods in a store *(do internet shopping)*, meeting people in a café *(chat in a chatroom)* or attending school *(subscribe to an internet educational course)*. Such user intended actions in the web are implicitly connected to the attribute 'place', therefore, no extra field 'space' is included in the agent's cognitive map.

Unlike the examples mentioned above, there exist activities that are not by default associated with a place. An example of such an activity is a person searching for a satellite picture of a certain geographical area in the internet: The activity of searching and browsing in the web is not spatial itself, merely the area of interest (the satellite picture) has a spatial component. Therefore, an additional attribute 'place' is added to the agent's mental map.

5. The Navigation Process in the www

5.1 Structure of the Navigating Agent

Several types of agents are introduced in (Wooldridge 1999). For our approach we take a 'utility-based agent' that has an internal state which can store information about the agent and the environment. Further, the agent applies a utility function that maps a state onto a real number. This function allows the agent to select the activity that leads him as close as possible to his predefined goal.

The agent's state consists of the semantic map, a goal, the agent's actual position, and a history of positions. The agent's actual position is split: One position describes the actually visited web page in the web graph, the other position refers to the mental map and describes which element of each field has been reached in the navigation process so far.

5.2 The Navigation Strategy

The agent starts at the index page of the given web domain. During the navigation process, the agent perceives a number of links with the labeled information. The agent's strategy is to select the link, of which the corresponding element in the semantic map has the shortest path to its predefined goal in the mental map. The strategy serves for the local decisions at each node, but it is not capable of providing the shortest overall path to the target page in the web (in the sense of the number of mouse clicks to reach the target web page). Choosing the overall shortest path in the web would require knowledge of the web structure (e.g., through reading a site map or exploration of the web environment) or navigating along a familiar path.

As we exclude knowledge of the web structure (and therefore the familiarity with any path) from our model, we assume that the agent relies on his semantic map and chooses the shortest path within that mental map.

5.3 Visualization of a Navigation Step

We give an example to visualize the web navigation strategy (**Fig. 3**). The agent's web position is at the start node '1' of the web domain in **Fig. 2**. The agent perceives two links ('do business' and 'recreate') and finds the corresponding position of these terms in the mental map: 'do business' is part of the field 'user intended actions', 'recreate' is member of the field 'action affordances' (see **Fig. 1**). The agent determines the shortest path between the perceived links and their goals in the mental map. As the length from 'do business' to 'order' amounts to 2 steps, and the length from 'recreate' to 'running' amounts to 3 steps, the link 'do business' is chosen at this point of decision (see **Fig. 3**).

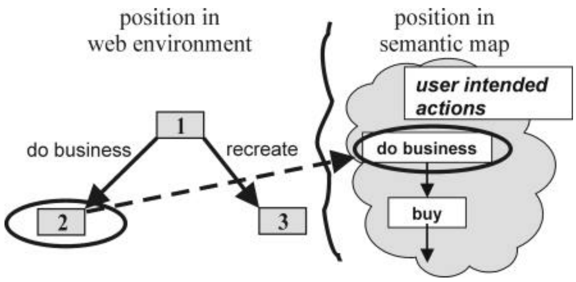

Fig. 3. Determining the best link through the corresponding element in the mental map

6. Simulation

In this section we simulate a navigating agent in a web environment, following the web navigation strategy introduced before. We realize the navigation model through algebraic descriptions using classes with their functions. The model contains abstractions of the web environment, and a navigating agent whose decisions are based on a mental map. We implemented the algebraic specifications through the functional programming language Haskell (Thompson 1996). In this paper we show few lines of code and restrict ourselves mostly to describing the functionality of the model. The complete code can be downloaded from *ftp://ftp.geoinfo.tuwien.ac.at/hochmair* and will not be listed here.

The implementation of the formalized model allows to assess a given web environment with regard to ease of navigability. Various software agents can be tested in existing web environments, 'dead ends' and loops can be detected while the agent is navigating from the start page to the target web page. The web environment and the agent's mental map stay invariant during the web navigation process whereas the agent's positions in the web and the mental map change with each navigation step.

6.1 The www Environment and the Mental Map

For modeling the navigation process we abstract web pages in the web domain as nodes and the connecting links as edges. The web nodes, as shown in **Fig. 2**, consist of an integer identifier, but do not contain semantic information. The semantic information of the web environment is stored in the web links, they contain the semantic information of the web page they are leading to. A data type representing the web graph is built with the constructor function `G EW NodeW`, where `G` indicates a graph, `EW` stands for a list of elements (a list of `NodeW` here) with an additional string for the semantic information, and `NodeW` is the data type representing a web node.

Similar to the web environment we abstract elements in the mental map as nodes and their connections as links. Contrary to web nodes it is the nodes in the mental map, which carry the semantic information. A field in the mental map is coded in the form `Field EM NodeM`, where `Field` constructs a graph with an additional string (for the name of the field), `EM` stands for a list of elements (without any additional semantic information) and `NodeM` expresses a mental node. As the mental map consists of several fields (see **Fig. 1**), we unite all fields in a list to create the complete mental map. Square brackets symbolize a list in Haskell, therefore the data type of a mental map is represented as `[Field EM NodeM]`.

The following lines demonstrate how to create the web environment of **Fig. 2** in the simulation. The data are initially given in a list of triples consisting of start node, end node and the semantic content of the link (*webStrings*). The *makeEW* function takes one triple and converts it to a web link. The *map* function applies *makeEW* to each element of the *webStrings* data, and the *insertG* function inserts the resulting list of web links into an empty graph.

```
webStrings = [(1,2,"do business"), (1,3,"recreate"),...]
wmap = foldr insertG (G []) (map makeEW webStrings)
```

Creating the fields of the mental map is a similar process. We demonstrate the process for the field 'user intended actions' (*mmapUia*), which contains three nodes. To get the complete mental map, all fields are united in one list, which is labeled *mmap*.

```
mmapUiaS = [("do business", "buy"),("buy","order")]
mmapUia = foldr insertF (Field "Uia" (G [])) (map makeEM
          mmapUiaS)
mmap = [mmapUia, mmapPhys, mmapAa, mmapBrand, mmapSize]
```

The agent's mental goal is formalized as a list of edges. Each edge describes the goal for one single field and consists of the fieldname (data type `String`) and the goal node (data type `NodeM`). For all elements of the mental goal see **Fig. 1**.

6.2 The Navigating Agent

During the navigation, the agent stores the data in the agent's state (see section 5.1) which is needed for the further decision process. In the agent's state, the web position is stored as a web node, the mental position and the goal are stored as lists of edges (each of them consisting of a string and a mental node), the mental map as a list of fields, and the history as a list of tupels, where each tuple contains the agent's visited position in the mental map and in the web. The agent is created through the definition of a new data type *Agent*, which consists of the five elements of the agent's state.

We construct a virtual agent who navigates in the given web environment visualized in **Fig. 2**. The following lines of code show how the agent's state is filled with data. The agent, called *fred1*, starts at the top page of the domain at web node 1 (*posw*). His mental position (*posm*) has an empty value for each field (defined in *posstartM*), the set of goal elements (*goal1*), and the mental map (*mmap*) are the ones defined above, the history list (*hi*) of previously visited nodes is empty.

```
fred1 = Agent posm goal mentalmap posw hi
   where posw = NodeW 1      -- position in web
         posm = posstartM    -- position in mental map
         goal = goal1        -- mental goal
```

```
mentalmap = mmap   -- mental map
hi = []            -- history list
```

To define the agent's operations - they cause changes in the agent's state - we define a class *State*. The class signature describes the parameters of the two included operations *step* and *navigate*. Both operations have a web graph *(Web)* and one parameter (*agent*) as input, and have the changed parameter as output. In the instance for the data type *Agent*, elements of the state, such as the agent's position or history, may change through one of the operations. The *step* function defines one single navigation step of the agent, whereas *navigate* performs a complete navigation process from the start page to the target using the *step* function recursively.

```
class State agent where
      step :: Web -> agent -> agent
      navigate :: Web -> agent -> agent
```

The *step* function consists of several sub activities that are implemented in the instance of the class *State* for a data type *Agent*:

- a 'step' in the mental map which changes the actual position in the mental map
- a step in the web environment, which changes the position in the web graph
- update of the history list

We use three conditions to break the recursion in the *autoNav* function: The first condition is the perception of the link with the lowest element in the 'user intended actions' field. This indicates that the goal is reached. The second condition is given through a limitation of navigation steps. A specific function checks the number of elements in the history list to be smaller than a given maximum value. If the length of the history list is higher than the maximum value, this indicates that the agent is caught in a loop. The third condition for a break is that no 'useful' link can be perceived, which means that no information of the provided links can be matched with the elements of the semantic map.

6.3 The Decision Process

The core of the decision process is the shortest path algorithm of Dijkstra (1959) as given by Kirschenhofer (1995). It is used for the assumption that the agent determines the shortest possible path in the mental map (see section 5.2). The shortest path function has two nodes and a graph (a field in the mental map) as input and returns a list of nodes. The result describes the shortest path between the two given nodes in the graph, in our case between an element of a mental field and its target element of the goal definition. To come to a decision, several substeps have to be performed mentally by the simulated agent:

- determine all outgoing links from the actual web page
- find the corresponding field and goal in the mental map for the information on each of the perceived links
- calculate the length of the mental shortest path from each matched element to its goal in the mental map

- select the link with the shortest mental distance from its corresponding goal in the mental map

7. Case Testing

7.1 Single Step

This section demonstrates the simulated agent's navigation process in the web environment of **Fig. 2**. The agent starts at web node 1 and perceives two links, 'do business', and 'recreate'. We apply the *step* function on the agent to see the result for one single step.

```
Test input> step wmap fred1

>> POSMentalMAP  Uia 'do business', Phys '', Aa '', Brand '',
Size '', POSWEB 2,
HISTORY  [ POSMentalMAP Uia 'do business', Phys '', Aa '',
Brand '', Size '', PosWEB 2 ]
```

The result gives us information about the agent's position and the link he has decided to choose. We see that the agent's mental position (*POSMentalMAP*) has changed to 'do business' in the field 'user intended actions' (*Uia*) of the mental map. The other fields have not been entered yet, which is indicated by the empty strings of the mental position. The agent's web position (*POSWEB*) changes to node 2 (see also **Fig. 3**).

7.2 Complete Navigation Process

To perform a complete navigation process trough the web environment we apply the *navigate* function.

```
Test input> navigate wmap fred1

>> POSMentalMAP Uia 'order', Brand 'Nike', Aa 'do sport',
Phys 'shoe', Size '', POSWEB 22,
HISTORY  [ POSMentalMAP Uia 'do business', Phys '', Aa '',
Brand '', Size '',PosWEB 2 ] , [ POSMentalMAP ...Uia
'buy'..., PosWEB 4 ] , [ POSMentalMAP ...Phys 'clothing'...,
PosWEB 6 ] ,    [ POSMentalMAP ...Phys 'shoe'..., PosWEB 11],
[ POSMentalMAP ...Brand 'brand'..., PosWEB 14 ],
[ POSMentalMAP ...Aa 'do sport'..., PosWEB 16 ],
[ POSMentalMAP ...Brand 'Nike'..., PosWEB 18 ],
[ POSMentalMAP Uia 'order', Brand 'Nike', Aa 'do sport', Phys
'shoe', Size '', PosWEB 22 ]   GOAL REACHED = True
```

The history list shows the visited pages in the web environment (2, 4, 6, 11, 14, 16, 22), see also **Fig. 4a**, and in the mental map during the navigation position. The position after the navigation in the web (*POSWEB*) is node 22. The mental position after the navigation (**Fig. 4b**) can be read from *POSMentalMAP* in the first two lines.

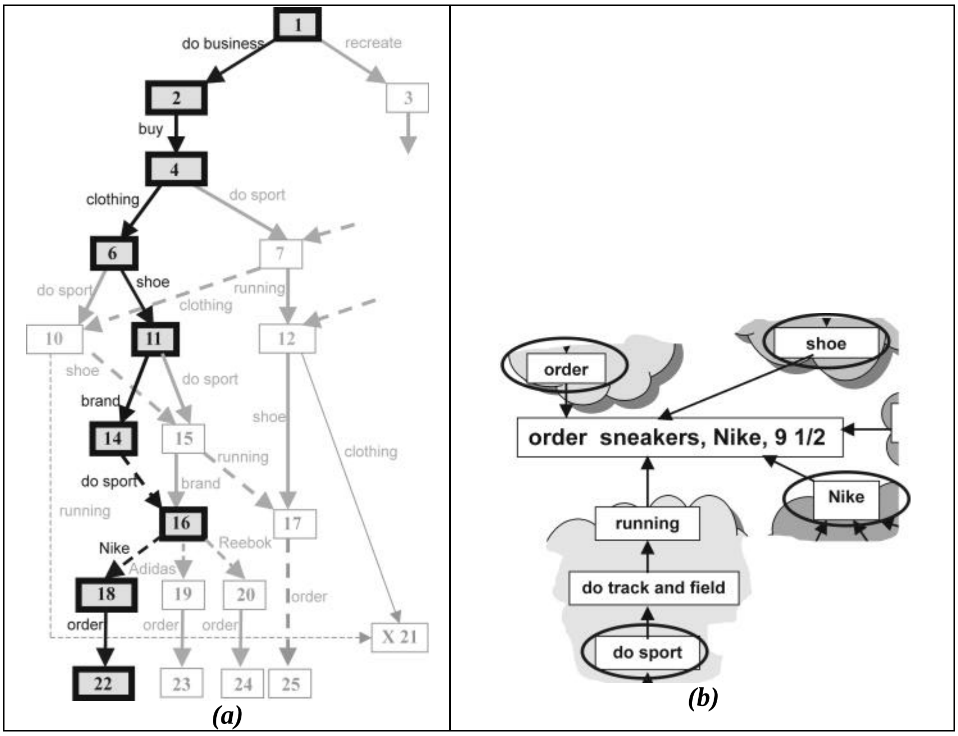

Fig. 4. Agent's position in the web environment (a) and the mental map (b) after a complete navigation process

The attribute 'Size' in *POSMentalMAP* contains an empty value string, indicating that no information about the shoe size was given throughout the navigation process. The reached element in the field 'action affordances' is 'do sport' instead of 'running' from the goal definition. These missing elements can be found after clicking the 'order' link (which is not visualized in Fig. 4a). The last line of the result indicates that the agent has reached the goal.

Conclusions and Future Work

We proposed a web navigation strategy that is based on a semantic map and can be applied in unknown web environments. The criterion for the navigation decision is the semantic distance of the perceived information from the mental target defined within the semantic map. Depending on the content and structure of an individual's mental map, navigation decisions can vary between individual users, although the same strategy is used. This fact means a challenge for web designers: The web environment and its links must be structured in such a way that the target web page can be reached via several paths, applicable for users with different mental maps.

The simulation of the model helps to detect critical navigation situations in a web environment and to improve the link structure in a given web domain. As we restricted ourselves to proposing a web navigation strategy in this paper, we did not discuss in detail the types of potential errors during the navigation process, e.g., getting lost in a 'dead end' or being caught in a loop of links. Such errors result from differences between the web user's and the web designer's mental maps. As one cannot directly observe a person's mental map, human subjects testing is a method to obtain more realistic models (than the ones constructed from WordNet) of a web user's mental map. Through analyzing the decisions of the test persons who navigate through an unknown web domain, the content and structure of the semantic map may be constructed stepwise. The results may be used to improve the link structure of a specific web domain. We consider these ideas as part of the future work.

Besides this, a goal for future work is to compare error taxonomies and navigation strategies in different environments (such as the web and the real world) using the algebraic structures of the formalized models. Another goal is to compare further features of navigation strategies such as shortcuts and deviations in different environments. A more general web navigation model with additional strategies and features may allow a more comprehensive comparison of navigation methods in the real world and the web.

References

Barsalou, L. W. (1987). The instability of graded structure: implications for the nature of concepts. Neisser**:** 101-140.

Dieberger, A. (1998). Social Connotations of Spatial Metaphors and Their Influence on (Direct) Social Navigation. Workshop on Personalized and Social Navigation in Information Space. A. M. K. Hook, D. Benyon. Kista, Sweden, Swedish Institute of Computer Science.

Dijkstra, E. W. (1959). “A note on two problems in connection with graphs.” Numerische Mathematik **1**: 269-271.

Ellis, D. and M. Haugan (1997). “Modelling the Information Seeking Patterns of Engineers and Research Scientists in an Industrial Environment.” Journal of Documentation **53**(4): 384-403.

Fellbaum, C. and G. A. Miller (1990). “Folk psychology or semantic entailment? A reply to Rips and Conrad.” The Psychological Review **97**: 565-570.

Gibson, J. J. (1977). The Theory of Affordances. J. Bransford. R. E. Shaw. Hillsdale, NJ, Lawrence Erlbaum Associates.

Hochmair, H. (to app.). Adapting One's Mental Model: An Essential Process for Successful Navigation in an Environment. Spatial Information in the Environment, Innovations in GIS 8, Taylor & Francis.

Jones, K. S. (1986). Synonymy and semantic classification. Edinburgh, Edinburgh University Press.

Jordan, T., M. Raubal, et al. (1998). An Affordance-Based Model of Place in GIS. 8th Int. Symposium on Spatial Data Handling, SDH'98, Vancouver, Canada (July 11-15, 1998), International Geographic Union.

Kirschenhofer, P. (1995). The Mathematical Foundation of Graphs and Topology for GIS. Geographic Information Systems - Materials for a Post Graduate Course. A. U. Frank. Vienna, Department of Geoinformation, TU Vienna. **1:** 155-176.

Kuhn, W. (1996). Handling Data Spatially: Spatializing User Interfaces. 7th Int. Symposium on Spatial Data Handling, SDH'96, Delft, The Netherlands, Faculty of Geodetic Engineering, Delft University of Technology.

Kuipers, B. (1978). “Modeling Spatial Knowledge.” Cognitive Science **2**.

Lakoff, G. (1987). Women, Fire, and Dangerous Things. Chicago and London, The University of Chicago Press.

Marchionini, G. M. (1995). Information Seeking in Electronic Environments. Cambridge, England, Cambridge University Press.

Miller, G. A. (1995). “WordNet: A Lexical Database for English.” Communications of the ACM **38**(11): 39-41.

Miller, G. A. (1998). Nouns in WordNet. WordNet - an electronic lexical database. C. Fellbaum. Cambridge, Massachusetts, MIT Press.

Norman, D. A. (1999). “Affordances, Conventions, and Design.” interactions.

Peuquet, D. (1998). Cognitive Models of Dynamic Phenomena and their Representations, http://www2.sis.pitt.edu/~cogmap/ncgia/peuquet.html.

Raubal, M. (to app.). “Ontology and Epistemology for Agent-based Wayfinding Simulation.” **15**(7) IJGIS.

Raubal, M. and M. Worboys (1999). A Formal Model of the Process of Wayfinding in Built Environments. Conference on Spatial Information Theory - Cognitive and Computational Foundations of Geographic Information Science. C. Freksa and D. Mark. Berlin-Heidelberg, Springer-Verlag. **1661:** 381-399.

Rips, L. J., E. J. Shoben, et al. (1973). “Semantic distance and the verification of semantic relations.” Journal of Verbal Learning and Verbal Behavior(12): 1-20.

Rosch, E. and C. B. Mervis (1975). “Family resemblance: Studies in the internal structure of categories.” Cognitive Psychology(7): 573-605.

Sholl, M. J. (1987). “Cognitive maps as orienting schemata.” Journal of Experimental Psychology: Learning Memory, and Cognition **13**(4): 615-628.

Smith, B. (2001). Objects and Their Environments: From Aristotle to Ecological Ontology. The Life and Motion of Socioeconomic Units. A. U. Frank, J. Raper and J.-P. Cheylan. London, Taylor and Francis.

Smith, E. E., E. J. Shoben, et al. (1974). “Structure and process in semantic memory: A featural model for semantic decisions.” Psychol. Rev.(81): 214-241.

Svensson, M. (1998). Social Navigation. Exploring Navigation; Towards a Framework for Design and Evaluation of Navigation in Electronic Spaces. N. Dahlbaeck. Kista, Sweden, Swedish Institute of Computer Science.

Thompson, S. (1996). Haskell - The Craft of Functional Programming. Harlow, England, Addison-Wesley.

Wilson, T. D. (1997). “Information Behavior: An Interdisciplinary Perspective.” Information Processing & Management **33**(4): 551-572.

Wittgenstein, L. (1953). Philosophical Investigations. New York, Macmillan.

Wooldridge, M. (1999). Intelligent Agents. Multiagent Systems - A modern Approach to Distributed Artificial Intelligence. G. Weiss. Cambridge, Massachusetts, The MIT Press.