The Necessary But Not Sufficient Role of Ontologies in Applications Requiring Personalized Content Filtering

Santtu Toivonen
VTT Technical Research Centre of Finland
P.O.Box 1000, FIN-02044 VTT
FINLAND
santtu.toivonen@vtt.fi

Oriana Riva
Helsinki Institute for Information Technology
P.O. Box 9800, FIN-02015 HUT
FINLAND
oriana.riva@hiit.fi

Abstract

We discuss an interactive approach for personalized information retrieval. Applying such approach allows users to create, store, and consume content that assists them in carrying out their current domain-specific tasks. The approach is grounded to the theory of distributed cognition, which emphasizes the involvement of external elements in thinking processes. We extend the scope of distributed cognition theory by applying it in the Semantic Web, where the content consumers and creators can be software agents in addition to humans. This leads us to consider the benefits and limitations of ontologies in the information retrieval process. We formalize and discuss a model for information usefulness determination. The model combines the level of understanding the information with the level of relevance assigned to the information. The understanding is based on whether the agent trying to make use of the information has access to the ontology/ontologies, to which the piece of information in question conforms. Assigning the relevance, instead, is based on the agent's preferences combined with contextual details currently characterizing the agent. Finally, we discuss a case example, where the information usefulness determination is applied.

Introduction

In the 1990s people finally confronted *information overload*, a phenomenon envisaged already over twenty years earlier (Toffler 1970). This was mainly due to the Web's expansive growth. Since the Web is an open environment where anyone can create content, people accessing it continuously come across material created by someone unknown to them. Typically people can make sense of the textual content they access, as long as they speak the language in which it is written. However, the emerging Semantic Web (Berners-Lee & others 2001) brings software agents in addition to human beings as content consumers—and possibly also creators. Understanding the content by both kinds of agents therefore presupposes the usage of *ontologies* or something alike.

This paper discusses ontologies, but also—and more importantly—recognizes the fact that conforming to ontologies is not enough in informationally rich environments such as the (Semantic) Web. For coping with information overload, *useful* material should be filtered from available Web content. How the usefulness is determined and utilized in information filtering, is one of the main topics of this paper.

The work is grounded on the theory of *distributed cognition*, which emphasizes external elements participating in cognitive processes of thinking creatures. We see the distribution of cognition as a powerful method for both explaining and designing actions involving the Web. In the physical world in which the distributed cognition theory has traditionally been applied, the external elements can range from pen and paper to calculators, computers, and other people. The virtual world, in contrast, changes the situation, since the external elements are digital in nature.

We approach the information filtering problem in the Web by introducing a thinking tool called the *Semantic Note*. Semantic Note is a piece of information realized as a set of statements. It is used for distributing domain-specifically useful information assisting an agent in performing some specific task. Whenever a situation in the Web is entered, should there be accessible Semantic Notes available, the information contained in them is typically provided. However, if the number of accessible Semantic Notes is large, usefulness of each is first determined. Prior to providing them, they are also sorted based on the usefulness, and some are possibly considered irrelevant and left out.

With the creator's permission, a Semantic Note can later on be consumed by others in addition to the creator. Sharing Semantic Notes with others in particular presupposes the usage of ontologies. This is because understanding a Semantic Note is very crucial in the usefulness determination process.

In addition to the understanding, the context of the evaluator is taken into account in the usefulness determination process. Context can be given various more specific meanings, but generally it refers to anything that can be used to characterize the situation of an entity. An entity, in turn, can be a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and the application themselves (Dey & others 2001).

User context is important to grasp in many Web applications. This entails trying to figure out what the user is doing or about to do in various situations, and what information there is that would be relevant to the user in these situations. The context detection task becomes even more relevant in mobile scenarios where the user is on the move and typically accessing the Web with a wireless device. The changing surroundings form a rich context around the user, and can be utilized when filtering suitable content for provision. In our approach the user profile is formalized as a set of rules utilized in matching the information content with the user's current context. If the user's current context has a match with a statement via the profile rule, it increases the usefulness value of the Semantic Note in which that statement is found. In addition to giving formalizations to understanding Semantic Notes as well as considering them relevant, we describe a case example in which these are put to use.

The rest of the paper is organized as follows: In the next section we present the theory of distributed cognition, and how it can be applied in the Semantic Web. The section after that discusses and formalizes the information usefulness determination. The following two sections present a case example where the information usefulness determination has been applied. Finally, we outline related work followed by concluding remarks.

Distributing Cognition in the Semantic Web

This section discusses the theory of *Distributed Cognition*, and especially how it can be extended in a Web environment by taking advantage of the Semantic Web technologies.

Extending the Distributed Cognition Theory

Generally, the theory of distributed cognition emphasizes the involvement of external elements in cognitive processes. The research subjects of distributed cognition research have traditionally been humans, and the external elements taking part in cognitive processes have been any entities that are outside the human brain. Examples of these external elements are books, calculators, rulers, maps, and other humans (Hutchins 1996). Releasing the cognitive load has traditionally been identified as the main reason for distributing cognition. In this kind of distributed cognition research, where the research subjects are human beings, considering the design and usage of ontologies has not been essential.

We extend the scope of distributed cognition research in a way that results in ontologies having a more central role in the investigation. In the Semantic Web software agents in addition to human beings can be content creators and consumers. In order to understand a piece of content, an agent has to have access to the ontology to which the content conforms. Along the same lines, software agents in addition to human beings are seen as creatures distributing their cognition. This differs from the traditional conception of computer involvement in distributing cognition, where human has always been the center of cognitive processes, and computer programs have only assisted (see for example (Hollan & others 2000; Albrechtsen & others 2001; Fjeld & others 2002)). In other words, we see *computation* instead of computers as an important medium for distributing cognition (Dourish 2001). Computation can be realized as a process involving both humans and software agents.

When distributing cognition, an agent interacts with its surroundings through the processes of *externalization* (also called objectification (Fjeld & others 2002)) and *internalization* (Vygosky 1978; Leont'ev 1978). More specifically,

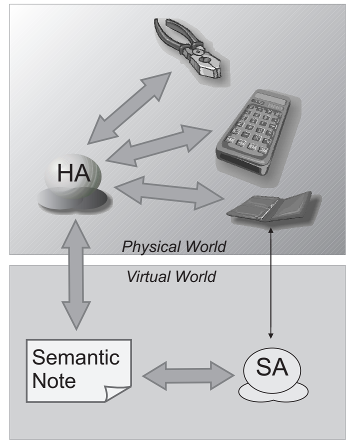

Figure 1: Means of distributing cognition

externalization refers to the process of grounding cognitive capabilities to the environment, and internalization to retrieving cognitive assistance from the environment. Since this paper concentrates on determining information relevance with regard to the current task of an agent, the focus is mainly on internalization part. Note however, that an agent externalizing information can assist the subsequent internalization activity by specifying the ontology to which the information conforms. This makes information retrieval an interactive process depending also on how the information is created, serialized and stored in the first place.

Another difference between distributing cognition in the physical world and in the Semantic Web is the nature of the media with which the cognition can be distributed. In the physical world, anything conceivable to a thinking creature can be used for distributing cognition. Typically these are physical structures (Kirsh 1996). The Semantic Web narrows the scope of distribution media, as Figure 1 depicts. Human agents (HA) can distribute their cognition to calculators, notebooks, tools, and so on, but software agents (SA) only to media accessible from the virtual space they reside in. Both forms, digital and physical, have their own benefits, as noted by (Dourish 2001). In principle also software agents could use physical structures for externalizing their cognition, for example by printing on paper. Moreover, should the software agent have access to machine vision equipment with pattern recognition capabilities, it could even internalize cognition from the piece of paper. These rare scenarios are depicted by the narrow arrow in Figure 1.

Distributing Cognition with Semantic Notes

We refer with the term *Semantic Note* to entities transmitting some meaningful piece of information, such as a definition of some complex concept, or instructions for completing a procedure. The domain of information in Semantic Notes is unrestricted, as mentioned in (Toivonen 2004). It means that a Semantic Note can contain a definition of a complex concept from any area. Therefore, the Semantic Note is better defined functionally as capturing the state of a cognitive pro-

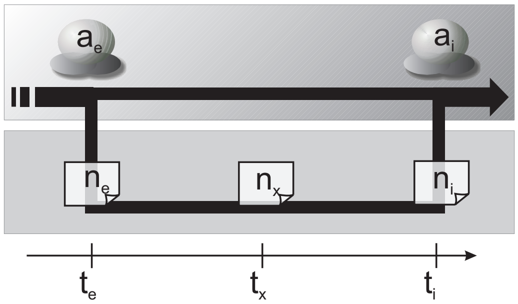

Figure 2: Semantic Note explicates the state of a cognitive process distributed into Web

cess's subprocess, which involves the Web. Figure 2 depicts this. After the externalization time, denoted by t_e, the Semantic Note (n) can go through changes which can have impact on its internalization at t_i. Considering these changes, however, is outside the scope of this paper.

A distinct feature of Semantic Notes is their purpose: they are often consumed and created in situations needing information for completing some *domain-specific* task. A Semantic Note can thus be conceived as a *tool*. More specifically, due to the domain-specificity, Semantic Notes can be considered as *tools for some task*, meaning that they do not have a purpose as such, but only in relation to the task they are used for (Dourish 2001; Heidegger 1927).

The Semantic Notes form a new "personalized" layer of information on top of other material in the Semantic Web. Semantic Notes can be characterized by comparing them with grocery checklists, which are created at home while checking what is missing from the refrigerator, and consumed in the grocery store either by the creator, or by someone else. Clearly this sort of analogy does not apply to all content in the Semantic Web. An important common denominator for Semantic Notes is that they are representations one or more entities, and therefore always refer to *something*. This fact entails that even though the Semantic Web dramatically restricts the kinds of objects for distributing cognition, it makes the usage of ontologies easier. All material in the Semantic Web is referred to with a URI, and this naturally applies to Semantic Notes as well.

Despite the fact that the same content carried by Semantic Notes should be accessible to both humans and software agents, we acknowledge that often these two kinds of agents have different roles, as they do in most Semantic Web systems. Humans make use of parts in Semantic Notes inaccessible to most software agents (for example content written in natural language), and vice versa. In contrast, software agents can provide decision support for humans operating in contextually rich environments. They can try to make use of the contextually meaningful parameters in the Semantic Notes, as well as trust-related factors, but unlike humans, pay no attention to the (unstructured) message contents.

Yet another extension to the traditional scope of distributing cognition is that releasing cognitive load is not the only motivation for doing it. We identify the need for information reuse and sharing as additional motivators for distributing cognition. Sharing Semantic Notes can be viewed as asynchronous broadcast where the agents deliver domain-specific information to each other. Having information sharing as a purpose also raises the emphasis on ontologies.

Determining the Usefulness of Semantic Notes

Although understanding the meaning of a piece of information is necessary for an agent to determine whether it is useful or not, it is still often insufficient as such. In this section we first present an ontology-based approach for determining how much of the information in a Semantic Note is understood. Subsequently, we formalize a way for an agent to determine whether some piece of content is relevant or not. Combining agent's understanding of the information with this relevance value enables it to determine the usefulness value of the information. This, in turn, can be used as a filter in personalized information retrieval. Some formalizations found in this section were originally presented in (Toivonen, Pitkäranta, & Riva 2005).

Semantic Notes and Ontologies

Semantic Notes consist of statements. Statements are opinions about states-of-affairs, such as *The Web site 'http://virtual.vtt.fi/virtual/proj2/dynamos/' is created by Santtu Toivonen*. The terms in a statement can be organized in the subject-predicate-object model of RDF, and conform to concepts in an explicitly defined ontology. This kind of machine-understandability is especially important for software agents in order to make use of the Semantic Notes. Using RDF, the above statement could be defined as follows:

```
<rdf:Description rdf:about=
  "http://virtual.vtt.fi/virtual/proj2/dynamos/">
  <dc:creator>Santtu Toivonen</dc:creator>
</rdf:Description>
```

Of the above RDF excerpt's terms, only the predicate (`dc:creator`) explicitly refers to an ontology, namely that of the Dublin Core metadata elements (Dub 1999). Combining the notion of statements with the approach adopted in (Williams & Ren 2001), an agent can be said to understand a statement found in a Semantic Note as follows: An agent (a) understands a statement (s), iff all the terms (t) constituting it conform to concepts (ϕ) found in an ontology (o), which is accessible to a. That is,

$$\mathit{understands}(a, s) \leftrightarrow \forall t : (t \in s \rightarrow \exists \phi : \\ (\mathit{conforms}(t, \phi) \wedge \phi \in o \wedge \mathit{access}(a, o))).$$

We assume that one statement is either understood or not understood by an agent. In principle a more specific definition could be given based on the understanding of the terms constituting the statement. In the above RDF excerpt, for example, the agent could be said to understand the meaning of the subject (a web page) and the predicate (creator) but not the object (literal "Santtu Toivonen"). This would entail that the agent understands two thirds of the statement. However, for our purposes a statement is on a more appropriate level of granularity. By applying a function und we assign the statements values, denoted by s_u, as follows:

$$und(s) = s_u = \begin{cases} 1 & \text{if all terms } (t \in s) \text{ are understood} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

n_u represents the agent's level of understanding of the Semantic Note (n). Let S_n be the set of statements included in n so that $s_1, s_2, ..., s_k \in n$, where $k = |S_n|$. n_u receives values between 0 and 1 based on the number of understood statements ($s_{u_1}, s_{u_2}, ..., s_{u_k} \in n$) divided by the number of all statements in the Semantic Note ($|S_n|$) as follows:

$$\begin{aligned} 0 \leq & n_u = \frac{1}{|S_n|} \sum_{i=1}^{|S_n|} s_{u_i} \leq 1 && S_n \neq \emptyset \\ & n_u = 0 && S_n = \emptyset \end{aligned} \quad (2)$$

Determining Information Usefulness

We define the information usefulness as a combination of understanding the information and considering it relevant. How the individual statements of the Semantic Note receive their relevance values, however, differs from the understanding. Assigning relevance to individual terms makes little or no sense. This means that the relevance of a statement, which in turn determines the relevance of a note, is formed based on some other criteria.

We apply a rule-based approach for determining the information relevance. The information content, of which the relevance is to be determined, is connected with user context via general preference rules specified by the user. The user context describes some essential details about the user's current situation, for example her location and current activity. Both the information content (that is, the Semantic Notes) and the user context are realized as sets of statements. If there exists a term (t_{ctx}) in a statement found in the user context, as well as a term (t_n) in a statement found in the Semantic Note so that both of those conform to respective concepts ($\phi_{ctx,n}$) which are navigable from the concepts (ϕ_{r_1,r_2}) found in the rule (r), the rule is said to be applicable (r_a). That is,

$$\exists t_{ctx} : \textit{conforms}(t_{ctx}, \phi_{ctx}) \wedge \exists t_n : \textit{conforms}(t_n, \phi_n) \wedge \textit{navigable}(\phi_{r_1}, \phi_{ctx}) \wedge \textit{navigable}(\phi_{r_2}, \phi_n) \rightarrow r_a$$

where *navigable*(ϕ_x, ϕ_y) means that there exists a network of concepts and relationships, realized as one ontology or several connected ontologies, that enables navigating between ϕ_x and ϕ_y. A positive match indicates that an applicable rule is found, as well as suitable values to satisfy it. Negative match means that there exists an applicable rule, but that the statements plugged in it do not have suitable values. In order to assign relevance values for the Semantic Notes utilizing the applicable rules, we define the following abstract function:

$$\textit{app}(r_a) = r_m = \begin{cases} 1 & \text{positive match} \\ 0 & \text{negative match} \end{cases} \quad (3)$$

The function *app* is realized as various concrete rules, that determine the relevance assignment (r_m, where m denotes "match"). The applicable rules (r_a) as well as the match value (r_m) are utilized in the relevance equation for Semantic Notes. Let R_a be the set of applicable rules, $r_{a_1}, r_{a_2}, ..., r_{a_k}$, where $k = |R_a|$. The Semantic Note relevance (n_{rel}) can receive values between 0 and 1 as the ratio between the sum of the match values ($r_{m_1}, r_{m_2}, ..., r_{m_k}$) and the number of applicable rules ($|R_a|$):

$$\begin{aligned} 0 \leq & n_{rel} = \frac{1}{|R_a|} \sum_{i=1}^{|R_a|} r_{m_i} \leq 1 && R_a \neq \emptyset \\ & n_{rel} = 0 && R_a = \emptyset \end{aligned} \quad (4)$$

Depending on the case, more fine-grained equations for relevance determination could be given. For example, some part of the relevance determination could be context-sensitive, that is, dependent on the current situation in which the relevance determination task is taking place, whereas other factors could hold across all possible contexts of the relevance evaluator. One could also consider the impact that trust assigned by the relevance evaluator has on the determination process. Trust can be aimed directly to the content, or to the creator of the content (Golbeck & others 2003). Moreover, context can actually influence the trust assignment and therefore the relevance determination, as discussed in (Toivonen & Denker 2004; Toivonen, Lenzini, & Uusitalo 2006). In this work, however, we do not decompose Equation 4 into more specific parts.

Finally, the information usefulness (n_{use}) of a Semantic Note is defined as a weighted combination of understanding (n_u) and considering it relevant (n_{rel}). We assign weights a and b to the individual parts of this equation, since in some application areas the relevance determination plays little or no role, whereas in other areas it is more important.

$$n_{use} = a * n_u + b * n_{rel} \quad (5)$$

where $0 \leq a + b \leq 1$ and $a, b \in \mathbf{R}^+$.

Processing Semantic Notes for Relevance

This section discusses the rules which are applied when determining the relevance of a Semantic Note. General processes of going through both the available Semantic Notes and the statements within one Semantic Note are presented, as well as some rule kinds that are used. Examples of these rules are given in a pseudo code. However, should there emerge a need for publishing these rules, they could be expressed in rule language such as SWRL, the Semantic Web Rule Language (Horrocks & others 2004).

Figure 3 depicts the states of going through the set of available Semantic Notes in order to determine their relevance. For scalability, some pre-filtering can be done for example based on the rules that apply across all possible contexts. The user's profile rules that apply with regard to her current context are applied. Note that the user can define whether some rule is mandatory or not. Should there exist mandatory rules, or strict preferences (Wellman & Doyle 1991), they have to be satisfied by all notes that go through the relevance determination process.

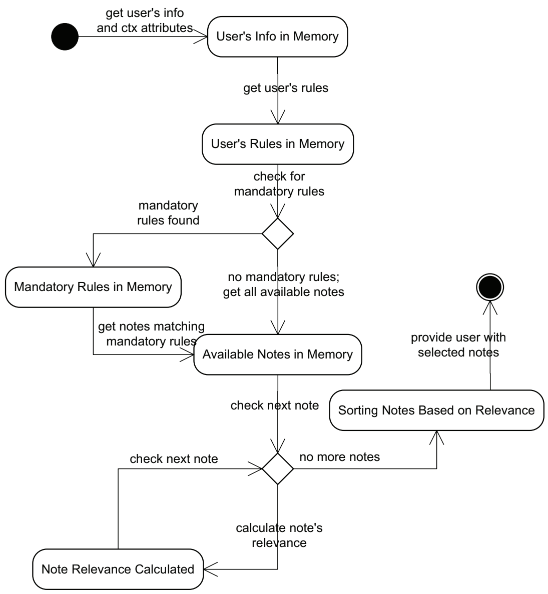

Figure 3: States of going through available Semantic Notes for determining their relevances

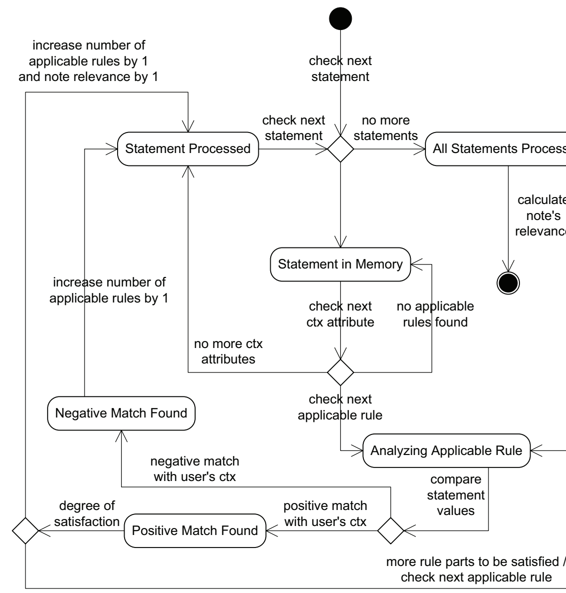

Figure 4: States of going through statements of a Semantic Note for determining its relevance

Figure 4 depicts going through the statements of a Semantic Note for determining its relevance. It can be embedded in bottom-left corner of Figure 3 (TRANSITION: CALCULATE NOTE'S RELEVANCE and STATE: NOTE RELEVANCE CALCULATED). This approach is simple in the sense that all statements are considered equally important. However, they could also be assigned weights and ordered, for example as in (Wellman & Doyle 1991; Fishburn 1999).

There are number of ways for specifying the relevance of a statement in a Semantic Note so that the current context of the user and her preferences are taken into account. Figure 5 depicts the general matching of a Semantic Note with a user context. Each statement in a Semantic Note is compared with each statement in the user context until either a positive match is found, a negative match is found, or there is no match and the next statement is taken into consideration. A positive match results in increasing both the numerator (r_{m_i}) and the denominator ($|R_a|$) of Equation 4 by one and the negative match increasing only the denominator by one. No match results in leaving the Equation 4 unaffected.

Typically the matching between Semantic Notes and user contexts depends on the values of the statement objects. The matching can be based on definitions and relationships in an ontology (for example finding out exact or close matches in the concept hierarchy/network), or on datatype comparisons (like string matching or numerical comparisons). One rule can concern one or more statements of a Semantic Note, and also one or more statements of the user context. These distinctions will be shown by the subsequent examples. A common thing for all the rules is that the connection between Semantic Notes and user contexts is based on matching, which is stored in the user profiles accessible to the system.

A Case Example: Delivering Personalized Information to Recreational Boaters

This section gives a case example of applying the information usefulness determination in practice by implementing it as a set of rules, which are applied when filtering content for the users. We first give an overview of the project in which this work is conducted, then consider the actual rules, and finally present the information usefulness values generated by applying these rules.

DYNAMOS Overview

We have applied the above information usefulness determination approach in a research project DYNAMOS[1]. DYNAMOS combines two areas in context-aware research, namely business-to-consumer (B2C) systems providing customers with 3rd party information, and consumer-to-consumer (C2C) systems for users to share information in a contextual manner.

In DYNAMOS, mobile users are dynamically provided with relevant descriptions of available services potentially of interest to them given their current situations. This entails taking into account the users' contextual details such as time, location, and activity. In addition, and more importantly with regard to the information filtering and distribution of cognition, DYNAMOS enables the users to annotate the service descriptions and share them with each other. The users of the DYNAMOS system can create and share also content that is not related to services. Therefore, in total there are three kinds of content, namely *service descriptions*, *service*

[1]Dynamic Composition and Sharing of Context-aware Mobile Services, http://virtual.vtt.fi/virtual/proj2/dynamos/

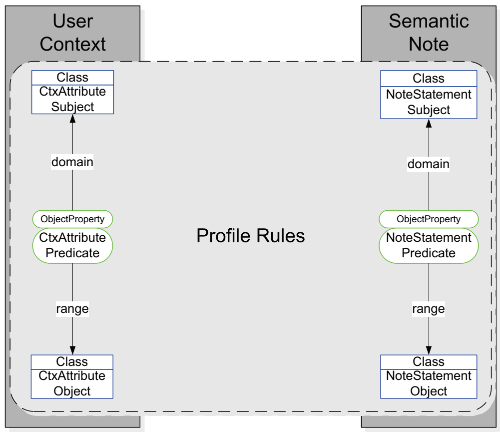

Figure 5: A general depiction of matching a Semantic Note with a user context

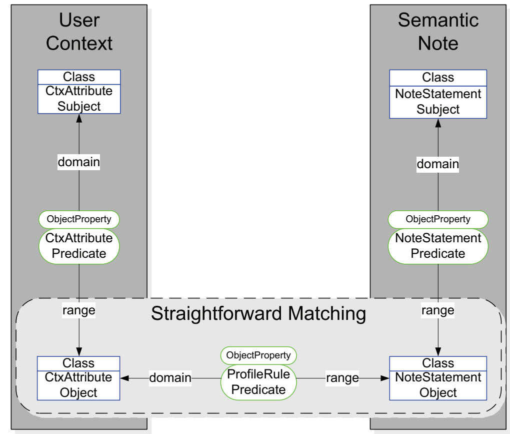

Figure 6: The simplest means for navigating from a Semantic Note statement to the user context

annotations, and *user notes*. A more extensive description of DYNAMOS can be found in (Riva & Toivonen 2006).

Recreational boaters were chosen as the user group to be studied in DYNAMOS. An example of a useful service description to that user segment would be an advertisement of a guest harbor. A typical annotation might be a rating of that guest harbor either in natural language and/or by giving "stars", and a user note could be a warning about the potentially dangerous route to the guest harbor during low tide. Relating this with the terminology used in this paper, all these three kinds of content can be considered as Semantic Notes. Even though it is not feasible to think that a service provider is "releasing its cognitive load" by creating service descriptions, such descriptions can assist the users of the system to perform cognitive tasks.

As is typically the case in context-aware information provision systems, the users' contextual details are either automatically estimated (such as location with GPS) or manually entered by the users themselves (such as activity, that is, what the user is currently doing). Based on this kind of information the users are provided with appropriate content, for example nearby guest harbors. In DYNAMOS the users have profile files containing information characterizing them. This information is in the form of rules, which are used to provide the users with appropriate Semantic Notes.

Matching Content Based on Profile Rules

As an initial requirements study, we conducted interviews, in which we asked twenty recreational boaters several questions related to their boating habits[2]. 70% of the interviewed boaters thought that a system like DYNAMOS would assist them in daily boating routines in general. However, one of the findings was that the interests and information needs of the boaters vary a lot based on their current activities. Generally, the information needs are the biggest when on the move (90%), approaching harbor (80%), or departing harbor (65%). In contrast, only 5% of the boaters mentioned having information needs after departing harbor. Typically the boaters either plan their trip before departing the harbor, or later on after the departure activities have ceased, and therefore have not much information needs right after departure.

Based on the interview results which show that information needs vary while engaged in different activities, we enabled the DYNAMOS users to specify activities in their profile files. More specifically, the activities can be associated with various interests. While sailing, a user can be interested for example in information related to guest harbors. A simple rule like this follows the structure depicted in Figure 6. In order to save space, the following RDF/OWL excerpts lack the namespace definitions. In reality, the user context, Semantic Note, and the rules for connecting them are defined in separate files denoted by respective namespace declarations. First, the Semantic Note has the following statement, which informs the category of the service it describes:

```
<Service rdf:ID="GuestHarborX">
  <hasCategory rdf:resource="#GuestHarbor"/>
</Service>
```

The user's context description has a statement informing the user's current activity, which is in our current implementation manually inserted by the user, because it is very difficult to detect the user's activity automatically. We acknowledge the fact that having users to manually change their activities is not desirable, and that automatically retrieved context attributes are much more user-friendly. However, given the user interview results which state that the activities are very important with regard to information needs, we have settled for the manual insertion of the activity by the user.

```
<User rdf:ID="UserCtxI">
  <currentActivity rdf:resource="#Sailing"/>
</User>
```

Finally, the user's profile rules have the following definitions for connecting the Semantic Note and the user context:

[2] The DYNAMOS boater interview skeleton is accessible at http://virtual.vtt.fi/virtual/proj2/dynamos/interviews/

```
<owl:ObjectProperty rdf:ID="interestedIn">
  <rdfs:domain rdf:resource="#Activity"/>
  <rdfs:range rdf:resource="#ServiceCategory"/>
</owl:ObjectProperty>
...
<Activity rdf:ID="Sailing">
  <interestedIn>
    <ServiceCategory rdf:resource="#GuestHarbor" />
  </interestedIn>
</Activity>
```

The following is a simple pseudo code representation of the general rule, which maps the service category of a service described by a Semantic Note to the current context of the user with the help of the connecting predicate `interestedIn`, found in the user's profile:

```
[Category rule:]
1 IF ctx:Ctx, ctx:hasCurrentActivity, ctx:Activity
2    ctx:Activity, prof:interestedIn, srv:Category
3    IF srv:Srv, srv:hasCategory, srv:Category
4       THEN note.relevance += 1
5            note.apprules += 1
6       ELSE note.apprules += 1
```

The namespace definitions `ctx`, `prof`, and `srv` indicate user context, user profile, and service, respectively. The above rule states, that should there be a specific activity declared in the user context (1st line) which is in the profile mapped to a specific service category (2nd line), if the current Semantic Note includes a description about a service belonging to that category (3rd line), the n_{rel}, denoted in the rule by `note.relevance`, is increased by 1 (4th line). $|R_a|$, denoted in the rule by `note.apprules`, is also increased by 1 (5th line). If that service belongs in some other category, `note.relevance` remains unaffected and only `note.apprules` is increased by 1 (6th line). If the Semantic Note has no statement about the service category, the second IF clause is not reached, and the Semantic Note's relevance value remains unaffected.

Suppose that the user has instead expressed her interest in boating in general, while she is at harbor. The service category for guest harbors is modeled as a subcategory of boating. This means that in order to provide the user with the Semantic Note describing *GuestHarborX*, navigating in the category hierarchy is inevitable.

```
<ServiceCategory rdf:ID="GuestHarbor">
  <subCategoryOf>
    <ServiceCategory rdf:ID="Boating"/>
  </subCategoryOf>
</ServiceCategory>
...
<Activity rdf:ID="AtHarbor">
  <interestedIn>
    <ServiceCategory rdf:resource="#Boating" />
  </interestedIn>
</Activity>
```

In addition to guest harbors, boating can have other subcategories such as boating supply stores and floating docks, and services matching these would also be provided to the user expressed interest in boating.

Our system currently makes no difference between whether it finds an exact match, or a match somewhere else in the category hierarchy. For us this solution is suitable, since the service database we are currently utilizing[3] has simple two-layered categorizations for services, such as archipelago/accommodation. However, a potential future development for the matching would be to take the distance in the hierarchy into account when matching, for example as proposed in (Stojanovic & others 2001).

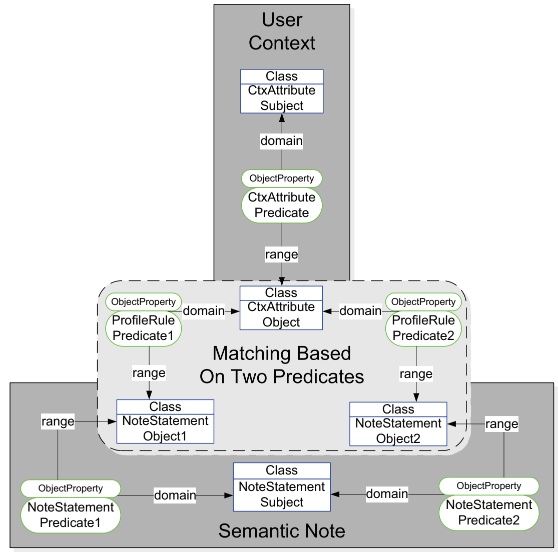

Figure 7: Matching based on two profile rule predicates

The user can also define multiple constraints to be satisfied by a Semantic Note. This is depicted in Figure 7. For example, the user can state that while sailing, she wants to receive "official" information about guest harbors that is created by service providers. Such information can be for example facility listings, directions and approaching maps. Afterwards, while already at harbor, she can be interested in any boating-related information, as mentioned. In addition, she can state that during this activity she is more interested in content created by the users of the system. Examples are opinions and ratings about various boating-related services around the harbor area.

```
[Double predicate rule:]
IF ctx:Ctx, ctx:hasCurrentActivity, ctx:Activity
   ctx:Activity, prof:interestedIn, srv:Category
   ctx:Activity, prof:prefCreator, srv:Creator
   IF srv:Srv, srv:hasCategory, srv:Category
      srv:Srv, srv:hasCreator, srv:Creator
      THEN note.relevance += 1
           note.apprules += 1
      ELSE note.apprules += 1
```

We also acknowledge the cases where someone wishes to receive information about services that have been rated high enough by the users. As described above, the users can give numerical ratings (or "stars") about the services provided to them and include these ratings in service annotations.

[3]Provided by Turku Touring: http://www.turkutouring.fi

Matching like this is based on datatype comparison, which is depicted in Figure 8. Say that the user wishes to receive information about services that have been rated four stars or more by other users. The next service annotation therefore satisfies this rule:

```
<ServiceAnnotation rdf:ID="HarborAnnotationY">
  <creator rdf:resource="#UserB"/>
  <rating rdf:datatype=
    "http://www.w3.org/2001/XMLSchema#int">4</rating>
  <refersTo rdf:resource="#GuestHarborX"/>
</ServiceAnnotation>
```

The following rule captures the actual matching based on restrictions like the one above. Two namespaces are introduced here: ann refers to concepts used in annotations created by users, and xsd indicates XML Schema, which contains datatype definitions used in OWL. Note that the below excerpt makes no restrictions on who has produced the annotation and the rating. Should the user wish to consider only ratings created by a specific user or a group of users, the creator information of the above excerpt could be utilized in the rule base. Here the rating rule is applied to one specific activity only, but it could be applied across activities and service kinds.

```
[Rating rule:]
IF ctx:Ctx, ctx:hasCurrentActivity, ctx:Activity
   ctx:Activity, prof:interestedIn, srv:Category
   ctx:Activity, prof:minRating, xsd:Int1
   IF ann:Ann, ann:refersTo, srv:Srv
      srv:Srv, srv:hasCategory, srv:Category
      ann:Ann, ann:hasRating, xsd:Int2
      xsd:Int2 >= xsd:Int1
      THEN note.relevance += 1
           note.apprules += 1
      ELSE note.apprules += 1
```

As mentioned above, the user can state whether she wishes to retrieve content created by service providers or by users of the system. Service annotations are created by users, but they always refer to services descriptions, which are created by service providers. If the user has declared the preferred content creators in her profile, the matching process takes that into account. Based on the above definitions, while in at harbor activity, *UserA* would receive the this service annotation. While in sailing activity, in contrast, she would receive the service description, to which this annotation refers. This shows one of the hybrid content provision model's strengths and novel features: Even if one wishes to receive commercial content, one can filter it based on user opinions.

Another important matching that is based on datatypes takes place when filtering content that is either referring to or created at a location spatially close enough to her. In our system the service descriptions and user contexts can have coordinates[4] associated to the services they describe. The following excerpts show the locations of a guest harbor and a user:

```
<Service rdf:ID="GuestHarborX">
  <serviceLocation rdf:datatype=
    "http://www.w3.org/2001/XMLSchema#string">
    lat60.185926926long24.82605626
  </serviceLocation>
</Service>

<User rdf:ID="UserCtxA">
  <currentLocation rdf:datatype=
    "http://www.w3.org/2001/XMLSchema#string">
    lat60.180798553long24.818225408
  </currentLocation>
</User>
```

[4]We describe coordinates using the WGS84 notation: http://www.wgs84.com

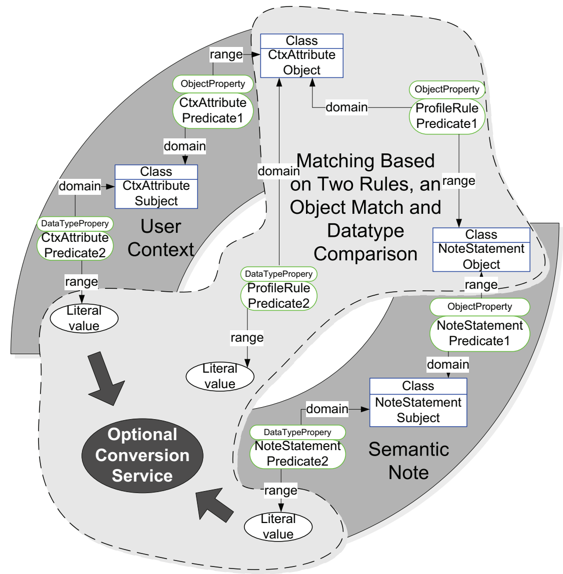

Figure 8: Matching based on object match and datatype comparison

Here the coordinates are given as one string that is parsed by the system. Another option would be to ground the longitude and latitude values to a GIS ontology, for example as described in (Chen & others 2004; Dürr & Rothermel 2003). Suppose that the user wishes to receive information about services that are no further than five kilometers from him. After converting the locations to a form that enables comparing their distance, the following expression can capture the matching process:

```
[Location rule:]
IF ctx:Ctx, ctx:hasCurrentActivity, ctx:Activity
   ctx:Activity, prof:interestedIn, srv:Category
   ctx:Ctx, ctx:hasLongitude, xsd:Float1
   ctx:Ctx, ctx:hasLatitude, xsd:Float2
   ctx:Activity, prof:maxRange, xsd:Float3
   IF srv:Srv, srv:hasCategory, srv:Category
      srv:Srv, srv:hasLongitude, xsd:Float4
      srv:Srv, srv:hasLatitude, xsd:Float5
      sqrt((xsd:Float1 - xsd:Float4)^2 +
           (xsd:Float2 - xsd:Float5)^2) <=
            xsd:Float3
      THEN note.relevance += 1
           note.apprules += 1
      ELSE note.apprules += 1
```

Evaluation: Assigning the Relevance Values

In order to demonstrate the impact of profile rules to relevance determination, suppose that we have two fictional users, *Bob* and *Jane*. Mandatory rules are omitted from this example in order to better illustrate the impact of statement relevances. *Bob* has the following rules in his profile:

1. When in sailing activity, interested in guest harbors.
2. When in sailing activity, interested in textual content.
3. When in at harbor activity, interested in boating.
4. When in any activity, maximum distance to content is 5 kilometers.

Jane has a richer rule base:

1. When in sailing activity, interested in guest harbors
2. When in sailing activity, interested in content created by service providers
3. When in sailing activity, maximum distance to content is 9 kilometers
4. When in at harbor activity, interested in boating
5. When in at harbor activity,
   (a) interested in content created by users AND
   (b) interested in content with minimum rating of 4 stars
6. When in at harbor activity, maximum distance to content is 2 kilometers
7. When in at harbor activity, interested in multimedia content.

In addition, we have two user contexts, *Ctx I* and *Ctx II*. In order to compare the impact of the rules, assume that both users can be in both of these contexts. The contexts have only two statements, which concern the user's current activity and location. For the sake of readability, the longitudes and latitudes are here combined into a single number denoting location:

- *Ctx I*
  - current activity = sailing
  - current location = 14
- *Ctx II*
  - current activity = at harbor
  - current location = 1

Finally, we have three content: one service description (*GuestHarborX*), one service annotation (*HarborAnnotationY*), and one user note (*NoteZ*):

- *GuestHarborX*
  - creator class = service provider
  - location = 7
  - service category = guest harbor
  - text content = "GuestharborX is open from mid-March to..."
  - multimedia content = approachmap.jpg
- *HarborAnnotationY*
  - creator class = user
  - location = 2
  - refers to = *GuestHarborX*
  - rating = 3
  - text content = "GuestharborX has a nice atmosphere. There you can..."
- *NoteZ*
  - creator class = user
  - location = 4
  - text content = "Be careful in this narrow passage when the wind is..."

Table 1: The usefulness values assigned to Bob and Jane based on their rules with equal weights (0.5) assigned for understanding the Semantic Notes (n_u) and their relevance (n_{rel}).

		Guest-HarborX	Harbor-AnnotationY	NoteZ
Bob	Ctx I	0.47	0.73	0.50
	Ctx II	0.55	0.65	0.58
Jane	Ctx I	0.80	0.73	0.33
	Ctx II	0.55	0.53	0.33

By combining the above information we can come up with the usefulness values depicted in Table 1. We assume that every statement apart from the content (multimedia or textual) is understood by the system. For example, *HarborAnnotationY*, which has five statements, of which four are understood by the system, has the n_u value $(1+1+1+1+0)/5 = 0.8$. However, should the content be structured and machine-readable, also the statements in it could be analyzed by the system and taken into account in the understandability determination.

For the sake of example, let us consider how the n_{use} for *HarborAnnotationY* is constructed for *Jane* when she is in the context *Ctx II*. Of the seven rules in her rule base, the first three are not considered as applicable (r_a), since they only apply when the current activity is "sailing". The rules (4) - (7), instead, are applicable, and therefore the denominator ($|R_a|$) for Equation 4 in this case is four.

Of the three applicable rules, rule (4) receives a positive match, since *HarborAnnotationY* refers to *GuestHarborX*, which is a service belonging in a subcategory of "boating". Rule (6) is not satisfied, even though the distance between the location where the annotation was created and the location of *CtxII* is smaller than the range *Jane* has assigned. This is because the matching of is based on the location of the service, which the annotation refers to, in this case to location "7". Finally, the only composite rule, namely rule (5), is not satisfied. Part (a) of rule (5) is satisfied, because *HarborAnnotationY* indeed is created by a user of the system. However, because the rating of the annotation (3 stars) is below the minimum rating (4 stars) assigned by *Jane*, part (b) of the rule is not satisfied. The conjunction denoted by the "AND" operator causes the whole rule (5) not to be satisfied. Note that rules can be composed using other operators too, such as disjunctions and substitutions, which might have different effects on the relevance calculation (Koutrika & Ioannidis 2005).

By summing up the relevance information we can conclude that the relevance value for *HarborAnnotationY* for *Jane* in the context *Ctx II* is $(1 + 0 + 0 + 0)/4 = 0.25$.

Table 2: The usefulness values assigned to Bob and Jane based on their rules with 0.8 weight assigned for relevance and 0.2 for understanding.

		Guest-HarborX	Harbor-AnnotationY	NoteZ
Bob	Ctx I	0.39	0.69	0.40
	Ctx II	0.52	0.56	0.53
Jane	Ctx I	0.92	0.69	0.13
	Ctx II	0.52	0.36	0.13

Table 3: The usefulness values assigned to Bob and Jane based on their rules with 0.2 weight assigned for relevance and 0.8 for understanding.

		Guest-HarborX	Harbor-AnnotationY	NoteZ
Bob	Ctx I	0.55	0.77	0.60
	Ctx II	0.58	0.74	0.63
Jane	Ctx I	0.68	0.77	0.53
	Ctx II	0.58	0.69	0.53

Finally, we can combine this with the above-mentioned understandability value and the equal weights (0.5 for both) using the Equation 5, and receive the value visible also in the Table 1, as follows: $0.5 * 0.8 + 0.5 * 0.25 \approx 0.53$.

Depending on the threshold, we can filter different content for *Bob* and *Jane* in different situations. For example, if the threshold is 0.6, *HarborAnnotationY* would be relevant for *Bob* when he is either in *Ctx I* or *Ctx II*. As for *Jane*, *GuestHarborX* and *HarborAnnotationY* would be considered relevant when she is in *Ctx I*, and nothing when she is in *Ctx II*.

The values can be amplified by emphasizing either the understanding or the relevance. Tables 2 and 3 depict this effect. If relevance is emphasized with 0.8 weight, as in Table 2, the 0.6 threshold yields providing *HarborAnnotationY* to *Bob* when he is in *Ctx I*, and nothing when he is in *Ctx II*. *Jane* would receive *GuestHarborX* and *HarborAnnotationY* when in *Ctx I* and nothing when in *Ctx II*. Since more statements are generally understood by the system than considered relevant based on the profile rules of users, emphasizing the relevance factor more than the understanding factor causes greater differences between the Semantic Notes. With 0.8 weight assigned to relevance causes the Semantic Note values to have the variance of 0.05 and standard deviation (SD) of 0.22, whereas with equal weights these numbers are: variance 0.02 and SD 0.15. With 0.2 weight assigned for relevance the valueas are: variance 0.01 and SD 0.09.

Related Work

An interesting research area for the theory of distributed cognition since its beginning has been to find out how computers and computer programs can assist humans in distributing cognition. The impact of computers on the distribution of cognition has been investigated for example in learning situations (Perkins 1993), when flying an airplane (Hutchins & Klausen 1996), onboard an aircraft carrier (Hutchins 1996), and in adaptive furnishing and interior design (Kirsh 1998). Our approach differs from these in the sense that computer programs and computers are not only objects participating in distributed cognitive processes of humans, but that certain computer programs can actually be the subjects of distributing cognition. In addition, we recognize other reasons besides the acute need for releasing cognitive load that motivate distributing cognition.

Personalization and user profiling has been around around before the Web (Rich 1979; Kobsa 1993). However, the vast expansion of the Web has boosted the interest on the subject and multiplied the number of research efforts concentrating on personalized information filtering. Some examples are (Sieg & others 2003; Liu & others 2004; Srivastava & others 2000). These have also influenced commercial applications like Amazon. Typically the approaches on personalized information retrieval are based either on automatic monitoring of the user's behavior for deducing the probable preferences, or on manual insertion of the preferences by the user (Koutrika & Ioannidis 2005). The third option is a combination of these. Our current approach makes use of manual user profiles, most importantly with regard to the notion of "activity", but nothing prevents us from either replacing the manual user profiles or alternatively complementing them with automatically deduced preferences. For example, after monitoring the user's behavior long enough, the system could deduce that with certain probability the user's activity during weekdays between 9AM and 17PM is "working", and provide the user with material she has typically showed interest towards when engaged in "working" activity.

Commenting existing Web content for one's own purposes, as well as sharing those comments with others, has been investigated for example in terms of the Annotea project (Kahan & others 2002). The notion of Web annotations has also been applied in CSCW when working on shared documents (Cadiz & others 2000). We extend the approach of these work in the sense that we include contextual parameters of the content creator in the annotation, and later on use those parameters in information filtering.

In our case example we combine two research traditions in context-aware systems, namely B2C approaches providing customers with commercial or other 3rd party information (see for example WebPark[5], (Mostefaoui & Hirsbrunner 2004; Yang & Galis 2003)), and C2C approaches for users to share information in a contextual manner (see for example (Pascoe 1997; Espinoza & others 2001; Burrell & Gay 2002; Busetta & others 2004)). The novelty of our approach with regard to these systems and research efforts is its combinatorial nature via the hybrid service provision model.

Conclusions and Future Work

We presented and interactive approach for information filtering. The approach is part of a larger effort of extending the theory of distributed cognition to new virtual environments

[5]The WebPark project:
http://www.geodan.nl/uk/project/WebPark/WebPark.htm

such as the Semantic Web, where the information creators and consumers can be software agents in addition to human beings. More specifically, information filtering is an important component of the internalization process carried out by an agent who wishes to make use of external elements while carrying out a cognitive task related to some domain.

We gave formalizations for determining information usefulness, which can be used in information filtering. In particular, we concentrated on understanding a piece of content, as well as considering it relevant with regard to the current task/context. We defined a thinking tool called Semantic Note for capturing a piece of information assisting when carrying out domain-specific tasks. The applicability of the approach was explicated by defining several concrete rules for determining the information relevance of Semantic Notes, and discussing them via a case example involving users in a contextually rich domain, namely recreational boating.

Potential development areas for the information usefulness determination include a mechanism for calculating the "distance" between concepts in a network, and use that as a component for relevance determination. In addition, we are considering structured contents in Semantic Notes and how to use them when calculating usefulness. One area for studying this is content related to weather conditions. 95% of the boaters we interviewed mentioned the use of mobile phone as a device for retrieving information about weather conditions while they are offshore. Should the weather information have structured content for wind direction and speed, temperature, humidity, and so on, the system could use that information in relevance determination. A user could define a rule stating "on sunny weather, interested in outdoor restaurants". The actual weather information could be delivered by a 3rd party or alternatively from other boaters equipped with the needed sensors delivering weather information to the system. Assigning weights to the statements in a Semantic Note is another interesting future research theme.

The information usefulness determination could be applied in other context-sensitive domain areas. In a popular sports event, the spectators could filter the information about the ongoing game based on their preferences—for example based on the membership of a certain fan club. It is envisaged that the framework would work also in professional domains. As an example, consider a service description referring to manufacturing equipment. Various actors working with it could attach annotations revealing about the current status of the equipment. These annotations would be filtered to the actors based on their profiles and contexts.

Acknowledgements

The authors would like to thank Heikki Helin for his contribution to the paper. Thanks also to Turku Touring for providing the project with a service database for test use. The DYNAMOS Project is jointly funded by TEKES, ICT-Turku, Suunto, TeliaSonera, and VTT.

References

Albrechtsen, H., et al. 2001. Affordances in activity theory and cognitive systems engineering. Technical Report Risø-R-1287(EN), Risø National Laboratory, Roskilde, Denmark.

Berners-Lee, T., et al. 2001. The semantic web. *Scientific American* 284(5):34–43.

Burrell, J., and Gay, G. 2002. E-Graffiti: Evaluating real-world use of a context-aware system. *Interacting With Computers: Special Issue on Universal Usability* 14(4):301–312.

Busetta, P., et al. 2004. K-trek: P2P knowledge management in wireless mobile networks. In Zaihrayeu, I., and Bonifacio, M., eds., *Working notes of the First International Workshop on Peer-to-Peer Knowledge Management (P2PKM)*, CEUR Workshop Proceedings.

Cadiz, J. J., et al. 2000. Using web annotations for asynchronous collaboration around documents. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work (CSCW '00)*, 309–318. New York, NY: ACM Press.

Chen, H., et al. 2004. Soupa: Standard ontology for ubiquitous and pervasive applications. In *1st Annual International Conference on Mobile and Ubiquitous Systems (MobiQuitous 2004), Networking and Services, 22-25 August 2004, Cambridge, MA, USA*, 258–267. IEEE Computer Society.

Dey, A., et al. 2001. A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. *Human-Computer Interaction (HCI) Journal* 16((2-4)):97–166.

Dourish, P. 2001. *Where the Action Is: The Foundations of Embodied Interaction*. Cambridge, MA: MIT Press.

Dublin Core Metadata Initiative. 1999. *Dublin Core Metadata Element Set, Version 1.1*. Specification identifier: http://dublincore.org/documents/2004/12/ 20/dces/.

Dürr, F., and Rothermel, K. 2003. On a location model for fine-grained geocast. In Dey, A. K., et al., eds., *Proceedings of the Ubiquitous Computing (UbiComp 2003), 5th International Conference, Seattle, WA, USA, October 12-15, 2003*, volume 2864 of *Lecture Notes in Computer Science*, 18–35. Springer.

Espinoza, F., et al. 2001. GeoNotes: Social and Navigational Aspects of Location-Based Information Systems. In *Proceedings of the International Conference on Ubiquitous Computing (Ubicomp 2001)*, 2–17. Springer.

Fishburn, P. 1999. Preference structures and their numerical representations. *Theoretical Computer Science* 217(2):359–383.

Fjeld, M., et al. 2002. Physical and virtual tools: Activity theory applied to the design of groupware. *Computer Supported Cooperative Work* 11:153–180.

Golbeck, J., et al. 2003. Trust networks on the semantic web. In Klusch, M.; Ossowski, S.; Omicini, A.; and Laamanen, H., eds., *Proceedings of Cooperative Intelligent Agents (CIA) 2003*, 238–249. Helsinki, Finland: Springer.

Heidegger, M. 1927. *Sein und Zeit*. Tübingen, Germany: Niemeyer.

Hollan, J., et al. 2000. Distributed cognition: Toward a new foundation for human-computer interaction research. *ACM Transactions on Computer-Human Interaction* 7(2):174–196.

Horrocks, I., et al. 2004. Swrl: A semantic web rule language combining OWL and ruleml. Technical report, W3C. http://www.w3.org/ Submission/SWRL/.

Hutchins, E., and Klausen, T. 1996. Distributed cognition in an airline cockpit. In Engeström, Y., and Middleton, D., eds., *Cognition and Communication at Work*. New York, NY: Cambridge University Press. chapter 2, 15–34.

Hutchins, E. 1996. *Cognition in the Wild*. Cambridge, MA: MIT Press.

Kahan, J., et al. 2002. Annotea: an open RDF infrastructure for shared web annotations. *Computer Networks* 39(5):589–608.

Kirsh, D. 1996. Adapting the environment instead of oneself. *Adaptive Behavior* 4(3/4):415–452.

Kirsh, D. 1998. Adaptive rooms, virtual collaboration, and cognitive workflow. In Streitz, N., ed., *Proceedings of the Cooperative Buildings, Integrating Information, Organization, and Architecture, First International Workshop, (CoBuild'98), Darmstadt, Germany, February 1998*, volume 1370 of *Lecture Notes in Computer Science*, 94–106. Springer.

Kobsa, A. 1993. User modeling: Recent work, prospects and hazards. In Schneider-Hufschmidt, M.; Kühme, T.; and Malinowski, U., eds., *Adaptive User Interfaces: Principles and Practice*. Amsterdam, The Netherlands: North-Holland. 111–128.

Koutrika, G., and Ioannidis, Y. 2005. A unified user-profile framework for query disambiguation and personalization. In Brusilovsky, P., et al., eds., *Proceedings of the Workshop on New Technologies for Personalized Information Access (PIA 2005)*, 44–53.

Leont'ev, A. 1978. *Activity. Consciousness. Personality*. Englewood Cliffs, NJ: Prentice-Hall.

Liu, F., et al. 2004. Personalized web search for improving retrieval effectiveness. *IEEE Transactions on Knowledge and Data Engineering* 16(1):28–40.

Mostefaoui, S., and Hirsbrunner, B. 2004. Context aware service provisioning. In *Proceedings of the IEEE/ACS International Conference on Pervasive Services (ICPS 2004)*, 71–80. IEEE.

Pascoe, J. 1997. The stick-e note architecture: extending the interface beyond the user. In *Proceedings of the 1997 International Conference on Intelligent User Interfaces*, 261–264. ACM Press.

Perkins, D. 1993. Person-plus: a distributed view of thinking and learning. In Salomon, G., ed., *Distributed Cognitions: Psychological and Educational Foundations*. New York, NY: Cambridge University Press. chapter 3, 88–110.

Rich, E. 1979. User modeling via stereotypes. *Cognitive Science* 3(4):329–354.

Riva, O., and Toivonen, S. 2006. A hybrid model of context-aware service provisioning implemented on smart phones. In *Proceedings of the 3rd IEEE International Conference on Pervasive Services 2006 (ICPS06)*. Lyon, France: IEEE.

Sieg, A., et al. 2003. Concept based query enhancement in the arch search agent. In Arabnia, H., and Mun, Y., eds., *Proceedings of the 4th International Conference on Internet Computing, Las Vegas, NV, 2003*, 613–619. CSREA Press.

Srivastava, J., et al. 2000. Web usage mining: Discovery and applications of usage patterns from web data. *SIGKDD Explorations* 1(2):12–23.

Stojanovic, N., et al. 2001. Seal: A framework for developing semantic portals. In *K-CAP 2001: Proceedings of the international conference on Knowledge capture*, 155–162. New York, NY: ACM Press.

Toffler, A. 1970. *Future Shock*. New York, NY: Bantam Books.

Toivonen, S., and Denker, G. 2004. The Impact of Context on Trustworthiness of Communications: An Ontological Approach. In *Proceedings of the Trust, Security, and Reputation on the Semantic Web workshop, held in conjunction with the Third International Semantic Web Conference (ISWC 2004)*.

Toivonen, S.; Lenzini, G.; and Uusitalo, I. 2006. Context-aware trust evaluation functions for dynamic reconfigurable systems. In *Proceedings of the WWW2006 workshop on Models of Trust for the Web (MTW'06)*.

Toivonen, S.; Pitkäranta, T.; and Riva, O. 2005. Determining information usefulness in the semantic web: A distributed cognition approach. In *Proceedings of the Knowledge Markup and Semantic Annotation Workshop (SemAnnot 2005), held in conjunction with the Fourth International Semantic Web Conference (ISWC 2005)*, 113–117.

Toivonen, S. 2004. Distributing cognition in the semantic web. In Calejo, M., ed., *Proceedings of the Doctoral Consortium on Enterprise Information Systems (DCEIS'04)*, 24–29.

Vygosky, L. 1978. *Mind in Society: The Development of Higher Psychological Processes*. Cambridge, MA: Harvard University Press.

Wellman, M., and Doyle, J. 1991. Preferential semantics for goals. In *Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91)*, volume 2, 698–703. Anaheim, CA: AAAI Press/MIT Press.

Williams, A., and Ren, Z. 2001. Agents teaching agents to share meaning. In *AGENTS '01: Proceedings of the fifth international conference on Autonomous agents*, 465–472. Montreal, Quebec, Canada: ACM Press.

Yang, K., and Galis, A. 2003. Policy-driven mobile agents for context-aware service in next generation networks. In Horlait, E.; Magedanz, T.; and Glitho, R., eds., *Mobile Agents for Telecommunication Applications, 5th International Workshop (MATA 2003)*, volume 2881 of *Lecture Notes in Computer Science*, 111–120. Marrakesh, Morocco: Springer.