Location based Reminder System with Reusable Ontology

Henrihs Gorskis and Arkady Borisov
Institute of Information Technology, Riga Technical University,1 Kalku Street, LV-1658 Riga, Latvia

Keywords: Application Ontology, Domain Ontology, Knowledge Reuse, Method Ontology, Reminder System.

Abstract: This paper provides a description of design of a reminder system that is based on location rather than on time. The system presented in this paper uses a reusable domain ontology model to access knowledge about a domain. The domain ontology is merged with a method ontology model in order to create an application specific ontology. This application ontology is used for communication with the user. It is proposed that using an ontology model enables the application to interpret the user input more flexibly. The reminder is triggered when a user is in close proximity to an establishment which is consistent with what was previously defined in the reminder application by the user with concepts from the domain ontology.

1 INTRODUCTION

The communication between a user and an application is often based on a checklist and is not quite dynamic or flexible regarding the input of the user profile. This paper proposes a system which uses ontology models for a location-based reminder system. This system lets the user define the reminder entry by using concept names from the ontology for a more dynamic type of user interaction. Using terminology from the ontology makes it possible for the system to search the concept model of the ontology to clarify the user intention. Doing so the system determines those facts which are applicable to the problem defined by the user. The system uses an application specific ontology as the knowledge base. The application ontology model can be created from any fitting and reusable domain ontology. This paper takes a look at a reminder system which reminds the user to buy something when he is near a shop. The type of product and shop are picked by the user, and the shops, which the system reminds the user of, is based on the user input and location. For this, an example domain ontology model for the city of Riga is used, which describes shops, stores, post offices and other businesses which can be of interest for the user. This way the system reminds the user to do something is based on the circumstances rather than time, which is different from the more classical approach to reminder systems.

2 REMINDER SYSTEM DESCRIPTION

The reminder system is based on the user creating reminder entries and the system constantly checking whether or not the conditions are met at the time of the check. This process begins with the user creating a reminder entry. This paper takes a look only at one type of reminder, but any number of reminder types can exist in such a system. The reminder is entered into the system by the user manipulating the linguistic sentence which describes the purpose of the reminder. The sentence is as follows:

"Remind me when I'm near a [business] where I can [action] [object]."

The words "business", "action" and "object" can be changed by the user in order to describe when the user wishes to be reminded. These words are directly linked to concepts within the application ontology model. They are either replaced by the most abstract concepts representing the meaning of each word or they are already the top most concepts for every branch of the taxonomy. The word "business" points to the most abstract concepts in the taxonomical branch for shops, post offices and other places where the user can perform the actions of which he wishes to be reminded of. Every action like "buy", "send", "order" etc. must be a sub concept of the concept to which the word "action" is linked to. The word "object" is linked to the top most concepts in the taxonomy, within the ontology, which describes

Gorskis H. and Borisov A..
Location based Reminder System with Reusable Ontology.
DOI: 10.5220/0005027501610167
In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development* (KEOD-2014), pages 161-167
ISBN: 978-989-758-049-9

those objects that are the target of the actions. The words, which the user can substitute, are based directly on these sub-taxonomies. This would be useful for preventing nonsense entries and for optimizing searches within the ontology. For example, let us say, the user chooses to create the following reminder sentence:

"Remind me when I'm near a [Store] where I can [buy] [bread]."

The concepts "Store", "buy" and "bread" have an ancestral link to the higher and more abstract concepts represented by "business", "action" and "object".

Once the user is done substituting the words with the names of the concepts, from the ontology which represent most closely the wishes of the user, the reminder system can begin to process the entry. The process begins by determining the concepts which are the closest to the given description. If the sentence given by the user about buying bread in a store is analyzed, it becomes clear that the ontology does not describe a store which sells bread. However, after searching the ontology and determining all the links between the concepts "Store" and "bread" and the property "buy" it can be determined that there exists the concepts "Food store" and "Super market" which in fact do sell "Food products". Knowing that both "Food store" and "Super market" are related to the concept "Store" and "bread" is related to "Food product", the system can determine that both locations are what the user is looking for.

The system described in this paper is location based. How exactly the location of the user is determined is not directly part of this paper. It can be assumed that an analysis of GPS data is performed and the distances between the user and the different locations stored in a database are calculated. Also, whether the database is first searched by location and then the remaining data entries are checked by the description, or if the database is first searched by the description and only then the distances to the fitting entries are calculated, is left to the system design. The important aspect of the database is that it contains data about the individuals of the ontology. Each individual has a class, a description of the individual and location data. The class of the individual links the individual to its parent concept and through that the ontology as a whole. The description of the individual is also based on the description of the parent concept. If the parent concept is a specific store type and the ontology states that such a store sells food item, then the individual of such concept can specify which specific food items are sold at the specific store. The data must be testable against the ontology it belongs to and the location of the user.

The system performs these tests constantly. Once the system has determined that a business which fits the description given by the user is located nearby, it notifies the user. Similarly to a traditional time-based reminder the user can choose to postpone the reminder if the situation of the user is unfitting at the time of the reminder. The delay can be temporal or based on other factors. The user could also supply additional data to the system which can be taken into account for later reminders. For example, the user could choose to ignore the specific store for all future searches. This can be useful if the store does not physically exist anymore or if the user dislikes the specific store for reasons undeterminable by the existing ontology or search sentence.

Once the user has performed the action and the reminder served its purpose, the user can choose to delete or keep the reminder in an inactive state for a later time when the reminder entry will be needed again.

There already exist some reminder systems that use ontology models and are context aware. One of such systems is called "Nama" (Kwon, O., Choi, S., Park, G., 2005). It is a context-aware multi-agent based web service. The main idea behind this personalized reminder system is that it tries to proactively identify the user needs. The reminder system presented in this paper differs from that system. "Nama" uses it's ontology to model user profiles, for explanation and prediction purposes. The system in this paper uses a domain ontology model for the purposes of interpretation.

Another field reminder systems are used in is medical care (Paganelli, F., Giuli D., 2007). Such systems also use ontology knowledge to model user profiles, medical care processes and guidelines. Again, this differs from the system presented in this paper, since medical reminder system mostly models the relations between different steps of the care process and use existing rule based knowledge with very little need to interpret situations, except for those cases where a patients data is analysed using a disease ontology (Buranarach, M., Chalortham, N., 2009). From this it is clear, that the idea of using other factors than time for a reminder system is not new and an existing field of interest (Ludford, P., Frankowski, D., 2006). However, the system description provided in this paper is different from those in the related works, because it tries to explain the exact relationship between the user input and the

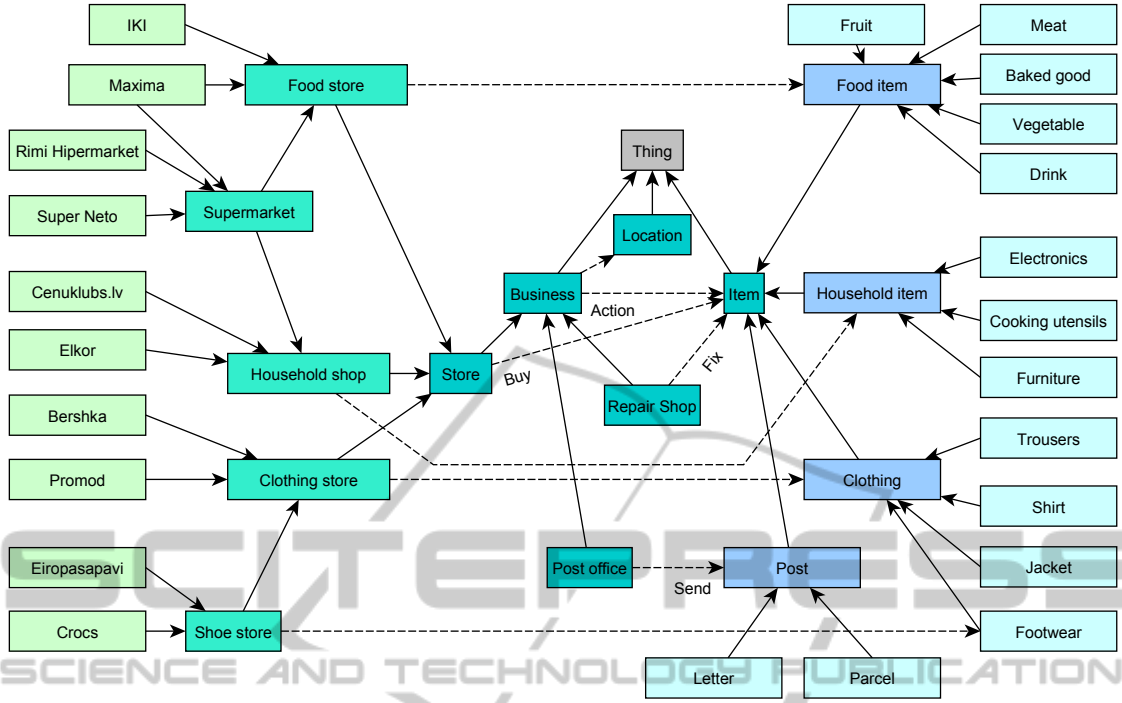

Figure 1: The domain ontology for businesses in Riga.

system functionality, through the use of an ontology model.

2.1 Task Statement

In order to present the design of the system and show its potential capabilities this paper will do the following:

- Describe the example domain ontology which provides information about the city of Riga and some businesses in it;
- Describe the user profile and method ontology for combining the system with an fitting reusable domain ontology;
- Describe the process of merging these two ontology models and show it on the example of the Riga ontology;
- Define the process of interpreting a user's reminder entry and finding more fitting concepts for the same request;
- Define the process of determining which individuals from the database are applicable for a given request;
- Show an example for a reminder based on the given city ontology.

These points are addressed successively in the upcoming sections of this paper.

3 THE DOMAIN ONTOLOGY

The domain ontology describes the basic knowledge of the system. For this paper we use an example domain ontology model. In this case the knowledge describes the city of Riga and the types of shops and other businesses located in it. The domain ontology defines the taxonomy of shops in a way which makes it possible to determine shops from a very abstract form to a more specific one. For the purposes of this paper the ontology is essentially simplified. The taxonomy for shops and items are quite basic. However, the processes and approaches described in the next sections are also applicable to more complex structured ontology models.

As can be seen in Figure 1 the ontology describes the existence of an abstract business concept which performs actions with items. In this ontology there are stores which sell items, post offices which send items and repair shops which fix items. The sub concepts of "store" are "Food store", "Household shop", "Clothing store", "Shoe store" and indirectly "Supermarket". The supermarket is a food store and a household item shop at the same time. The shoe store is a specialized type of clothing store. Every store has a property connection to a sub concept of item which indicates the type of item sold in this

type of store. The only exception of this, in the given ontology, is the supermarket concept which inherits these connections from its parent concepts. These properties are part of a hierarchy. The property between “Food store” and “Food item” is named “Buy Food” and is a sub property of the more abstract “Buy”. The same thing applies to the properties “Buy Household Item”, “Buy Clothing” and “Buy Footwear”. It is worth noting, that these are not the only properties in the domain ontology. Some properties are not shown. This is especially true for inverse properties. A Store, of course, does not buy items, it sells them. Selling is the inverse of buying. The ontology is defined by what a store sells; however, the user of the reminder system will be looking for a store where he can buy certain items. The user will be searching for triplets of the type {Store, Buy, Item}, as is defined by the reminder sentence, rather than {Store, Sell, Item}.

4 USER PROFILE AND METHOD ONTOLOGY

A user profile can include several reminder entries as well as some user preferences and some information about the user. The location data can be user-based or device-specific. Every reminder entry is linked to concepts within the application ontology and describes situation as defined by the user. Since the system relies on reusable domain ontology there exists a need to make sure that the application specific ontology, which is a product of the domain ontology, is valid and usable by the application. To solve this problem, an ontology model is created that describes all important concepts and properties of the functions within the system in a domain independent way. In this paper this ontology will be called method ontology. The purpose of the method ontology is to ensure that after merging it to the domain ontology, the resulting application ontology

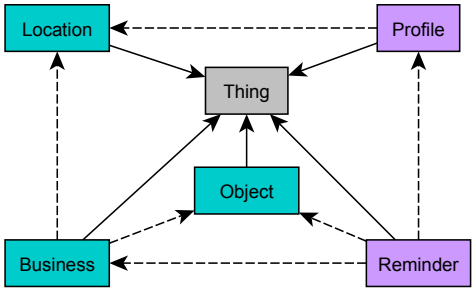

Figure 2: The discussed method ontology for this application.

contains all the concepts and properties which are referenced by the systems functions. For this application the method ontology has to provide a set of concepts which are the basis for connecting the domain ontology.

As shown in Figure 2, there are several concepts which serve as intermediaries for connecting the method to the domain. To some extent the method ontology mirrors the structure of the highest level concepts from the domain ontology. After merging these concepts they can be left in the application ontology or be fully replaced. From the method ontology we can see that the domain ontology must necessarily have a location concept. This is important for getting access to the location data in the individuals. Next, it can be observed that there is a concept describing an abstract business. A business must have a location and must be in some way connected to an object through an action. In the context of our domain ontology this will be shops selling products, repair shops fixing items or post offices sending items. The method ontology also shows how the reminder entry is liked to a business. The purpose of this ontology is to provide an access point to the reminder system. It describes necessary concepts, properties and restriction for use in the system. Any domain ontology which can satisfy these requirements or can be modified to satisfy these requirements can be used with the system.

The system contains rules or methods or functions which contain references to the concepts of the method ontology. Once the method ontology is used for the creation of the application ontology these links must be updated. Whether the concepts from the method ontology were taken into the application ontology, merged or replaced by concepts from the domain ontology does not matter as long as the references are maintained. By calling classes from the application ontology the system is capable of performing searches and other necessary actions. At this point merging is considered to be done manually by an expert. The result of merging is an ontology model which has all the necessary traits to be used with the application.

5 MERGING OF TWO ONTOLOGY MODELS

Before the system can use the domain ontology which contains the data of the city of Riga for understanding and creating the users reminder entry, the domain ontology has to be merged with the method ontology, thereby creating the application

ontology (Rothenfluh T.E., Gennari J.H., 1994). In the given example of the domain ontology we can see that the concepts “Business” and “Location” are the same. Merging in this case is very easy, since only the concepts “Profile” and “Reminder” have to be added to the domain ontology, as well as the property connections between the concepts. There is also one instance of replacing a concept from the method ontology with a concept from the domain ontology. The concepts “Item” and “Object” are in this case synonyms for the same thing. For the given example, the concept “Object” will be replaced and only the concept “Item” will be used. In more complex ontology models all these concepts could be added or replaced by other concepts from the domain ontology. The most important part is that the application ontology cannot be smaller than the method ontology and every concept and property from the methods ontology has to be used in order to create a link between the system and the application ontology.

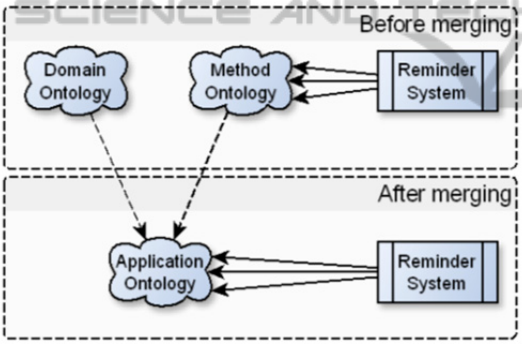

Figure 3: Visualization of the process of merging.

As we can see from Figure 3, the reminder system uses direct references to the concepts originally defined in the method ontology for the interpretation of the concepts from the domain ontology. Without this, the system would not be capable to determine which concepts are applicable to the processes in the system. In case of a more complex domain ontology where there would be more than one “business” concept, the concept of business or a renamed version of this concept could be added to the domain ontology in order to create this overarching concept of all businesses required for the reminder system. Also it is important to remember that in this specific case we are only interested in businesses which perform actions with items. The expert merging these ontology models must remember this, since in other domains businesses can be defined differently.
During the process of merging it is important to maintain the references from to system to the ontology. The way the system keeps track of the references is specific to the system, but we can think of it as a list of terms. For example, if the system is referencing the concept “Object” in the method ontology, but in the process of merging the expert chooses to keep the equivalent concept “Item” from the domain ontology replacing “Object”, the reference to “Object” from the system must be updated to correspond to this change. From then on any functions and actions, which search for or consider references to “Objects”, must do this with “Items”. Another part of merging, that needs to be considered, is the formatting of the location data. The given example stored location data as individuals of the concept “Location” with data properties of the type “double” for longitude and latitude. Another system might store this data directly within the individuals of businesses. The reminder system needs to know how to access the location data for comparison.

6 INTERPRETATION OF THE USER REMINDER REQUEST

Before the user can create a reminder entry the system has to have a connected application ontology model. Once the reminder entry is defined the system can start to process it. The created description of a business could be directly used to determine how fitting an individual from the data in the domain ontology is, for example, if the created reminder stated:

“Remind me when I’m near a [Store] where I can [buy] [bread].”

The system could look through every individual and try to determine whether it has a direct or indirect relative of the concept “Store” and whether the property individual “to buy bread” is applicable to it. However since the concepts used in defining the reminder are part of an ontology, it can be useful to use the available concept hierarchy in order to determine the intent and meaning of the user. A specialized search can be performed to determine which related concepts and properties best fit the description. This can be done by passing a triple to every related concept and property, determining the distance for all the parts of the triplet and passing the result to a result collector. This collector only keeps the same data once (a correct substitute triplet and its calculated distance from the concepts or property from which the original triplet was passed to). Once this is all done the system keeps only the top valid triplets.

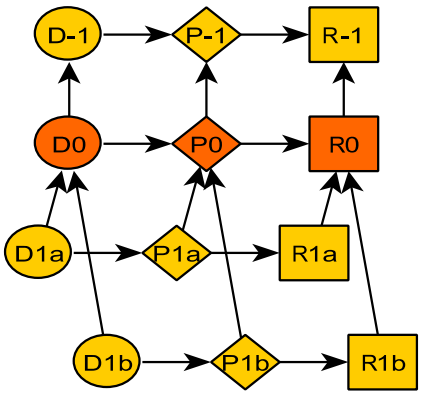

Figure 4: Visualization of concept search.

Figure 4 shows a visualization of this process. The elements with the index 0 represent the original triplet. The circular element is the domain for the triplet, the rhombus represents the property and the rectangle represents the range of the triplet. The search encompasses all related concepts and properties. For example, if the user has defined that he is looking for a triplet {Store, Buy, Item}, the search references the concept "Store" and determines that store has in fact a property "Buy" and the range of the property is "Item". In this case the user request was directly found within the ontology and no further interpretation is required. However, if the user has created a request for a triplet {Shop, Buy, Footwear}, the system could determine that the concept "Shop" has in fact a property "Buy", but the range of this property is not "Footwear" but "Item". In this case further searching is required. Since "Footwear" is a type of "Clothing" and "Clothing" is an "Item" the triplet given by the user is correct. It can be proposed that more specific concepts should be held in higher regard than more abstract concepts, because it requires more work for the user to pick a specific concept. Keeping this in mind, "Footwear" should become the centre of attention for this search. Therefore rather than simplifying the triplet to {Store, buy, item}, which is also a viable option, the search should specify the concept "Store" and the properties related to "Buy" and determine the possibilities depicted in Table 1. Any of the triples shown in Table 1 are valid for the final search for store individuals in the database.
Since the user specifically requested "Footwear" and {Shoe store, Buy Footwear, Footwear} is consistent with the original triplet, it can be interpreted that the user is in fact looking for a shoe store.

Table 1: Inferred triplets.

Domain	Property	Range
Store	Buy	Item
Clothing store	Buy	Item
Clothing store	Buy	Clothing
Clothing store	Buy Clothing	Clothing
Shoe store	Buy	Item
Shoe store	Buy	Clothing
Shoe store	Buy Clothing	Clothing
Shoe store	Buy Clothing	Footwear
Shoe store	Buy Footwear	Footwear

However, since the user did not specify "Shoe store" to be the store of his choice, it cannot be completely dismissed that the user is looking for any type of store which happen to also sell shoes.
Besides analyzing the user preferences, interpretation can be used to limit the count of active concepts for a given task. If the ontology would have been of a very large size and constantly searching it to determine applicable individuals would be resource intensive, interpretation could be viewed as a type of optimization.

7 SEARCHING THE DATA

Once a reminder is added and the system interpreted the request, the reminder system needs to determine when to signal the user. If the data set is small enough, as it is in this paper, it is possible to constantly check every data entry for businesses in the city. However, a more full ontology of very different types of businesses and data for every store, post office and other establishments in the city would be very resource intensive to search this way. There are many ways to optimize a database. One possibility would be to index the data by concept. By using the most important concepts determined by the interpretation step, only the applicable individuals could be pulled from the database and tested further. Further searching would also include determining the proximity of the store. However, if there are many stores of any given type, or if interpretation did not narrow the applicable concepts down enough, this approach would still not be very effective. Another solution would be to store the data spatially in the database. This way the test for proximity would be the first to be performed and only the closest businesses would be tested for validity with the user request.

Determining the validity of an individual business from the concepts involved in the creation of the reminder entry is done by testing concept-

individual relations, required property relation existence and disjoint property absence. This means, that only those individuals, who are directly or indirectly related to the business concept given, are eligible. The individuals in question also need to have the object properties, which are defined in the reminder. If the reminder is looking for a Supermarket which sells bread, the individual of a supermarket concept has to possess this property. Some such properties can be implied to some extent. If a supermarket by definition sells bread or food in general, such a property relation can be deducted. An individual also must not contain any disjoint properties which negate the necessary property. For example, if an individual is of the concept "Food store" and is therefore assumed to sell all kinds of foods, but contains a disjoint property "does not sell" to the food type the reminder is looking for, this individual cannot be offered to the user. The result of such a search would be a list of fitting businesses in the area. This means that the reminder system is also, to some extent, a recommendation system. It provides the user with a list of stores that are fitting with the request he made and are sorted by proximity to the user.

8 CONCLUSIONS

This paper has proposed and described a reminder system that reminds the user to perform some tasks while he is moving through a defined space with predefined locations in it. This reminder system relies on its ability to search and use an ontology model for the purposes of providing the possibility of flexible input to the user. This gives the user more freedom and makes it easier to define what the user is looking for, by letting him use a form which is closer to natural language. The user is capable of presenting the reminder system a request which may be quit abstract. This makes it possible for the user to define an idea rather than a specific task. However, by constraining the user input to the concepts existing in the application ontology errors are minimized and direct access to the ontology is possible.

The described system is capable of using different domain ontology models. This is useful for having different implementations in different cities or for the use in completely different domains, as long as the underling structure is similar enough. The merging process, which is done before the reminder system is ready for use, can use any fitting domain ontology. This way the reminder system is not constrained to just one problem domain. The example described in this paper showed a simplified merging problem, since most of the concepts were already present in both the domain and the method ontology. The reminder system depends mostly on the hierarchical structure of concepts, which makes it quit easy to perform merging. The expert needs to pick out the most abstract concepts which represent the places and actions a user might be looking for.

The system described in this paper is a work in progress. The description of this system and the steps and processes stated in this paper are a foundation for future implementation. Future work will be focused on building a prototype using Protégé 4 and geographical data collected about business locations in the city of Riga.

ACKNOWLEDGEMENTS

We would like to thank Dr.sc.ing. Yuriy Chizhov (Department of Modelling and Simulation, Riga Technical University) for his contribution and assistance in developing the task for a reminder system based on geographical location using ontology elements.

REFERENCES

Kwon, O., Choi, S., Park, G., 2005. NAMA: a context-aware multi-agent based web service approach to proactive need identification for personalized reminder systems. In *Expert Systems with Applications 29*. ELSEVIER

Paganelli, F., Giuli D., 2007. An Ontology-based Context Model for Home Health Monitoring and Alerting in Chronic Patient Care Networks. In *National Inter-University Consortium for Telecommunications, Dept. of Electronics and Telecommunications*. University of Florence

Ludford, P., Frankowski, D., Reily, K., Wilms, K., Terveen, L., 2006. Because I Carry My Cell Phone Anyway: Functional Location-Based Reminder Applications. In *Proceedings of CHI 2006.*

Buranarach, M., Chalortham, N., Chatvorawit, P., Thein, Y., Supnithi, T., 2009. An Ontology-based Framework for Development of Clinical Reminder System to Support Chronic Disease Healthcare. In *Proceedings of ISBME2013*.

Rothenfluh T.E., Gennari J.H., Eriksson H., Puerta A.R., Tu S.W., Musen M.A., 1994. Reusable Ontologies, Knowledge-Acquisition Tools, and Performance Systems: Protégé-II Solutions to Sisyphus-2. In *International Journal of Human-Computer Studies*. ELSEVIER