On Resource Profiling and Matching in an Agent-Based Virtual Organization*

Grzegorz Frąckowiak[1], Maria Ganzha[1], Maciej Gawinecki[1], Marcin Paprzycki[1], MichałSzymczak[1], Myon-Woong Park[2], and Yo-Sub Han[2]

[1] Systems Research Institute, Polish Academy of Science, Warsaw, Poland
{maria.ganzha,marcin.paprzycki}@ibspan.waw.pl
[2] Korea Institute of Science and Technology, Seoul, Korea
{myon, emmous}@kist.kist.re.kr

Abstract. In our work we are designing agent based support for workers in a virtual organization. In the system under development, all resources are to be ontologically demarcated and utilized through semantically-driven information processing techniques. This paper contains preliminary considerations concerning resource profiling and ontological matching methods which we intend to utilize within the system. Two applications that we are currently developing are used to illustrate the proposed approach.

1 Introduction

Our recent work is devoted to personalized worker support in an agent-based virtual organization. The basic assumption underlying our work is that emergent technologies such as software agents [25] and ontologies [19] should be utilized as a foundation around which the proposed system should be conceptualized. In particular: (i) organizational structure, consisting of "roles" played by various entities within the organization and interactions between them, should be represented by software agents and their interactions, and (ii) domain knowledge, resource profiles and resource matching should be based on ontologies and associated with them reasoning machinery. Thus far, in [20] we have outlined processes involved when a "project" is introduced into an organization (approached from the point of view of resource management). Later, in [4] we have conceptualized roles played by various entities identified in [20]. This allowed us to establish roles to be played by (a) software agent(s) alone, (b) by human(s), and (c) by human-agent team(s). We have also discussed agent interactions and introduced a specific application, a *Duty Trip Support* (*DTS*), to illustrate how the proposed top level design of the system can be utilized in practice. Separately, in [3] we have proposed how e-learning can be introduced into the system in support of adaptability of human resources. Finally, in [21] we introduced basic ontologies to be used in the system (including the concept of a generic resource).

*Work was partially sponsored by the KIST-SRI PAS "Agent Technology for Adaptive Information Provisioning" grant.

The aim of the current work is to discuss how we plan to introduce resource profiles and use them to perform simple matchmaking operations. The context of our work is provided by two applications that have been selected as initial test cases for our system. The first one is a rather simple *Grant Assistant System* (*GAS*), while the second one is the above mentioned *DTS*. Implementation and experiments with these two subsystems will provide us with experiences that will be generalized in two directions: (a) to support more involved reasoning, and as a result (b) to support a more general case of personalized information provisioning in an organization. Having said that, we start with providing a brief background of main features of the system under development.

2 System Overview

Our system is conceived as an agent-based virtual organization, which as its core provides support for project-oriented collaborative work [3]. Here, the notion of the project is very broad an includes installation of a satellite TV antenna as well as design and implementation of an intranet-based information system for a corporation. Structure of the organization and interactions between its participants are functionalized using software agents and their interactions. Domain knowledge and profiles of resources (human and non-human) are to be represented using ontologies [3, 4, 21] and overlay-based profiles (see below). Out of agents existing in the system, the *Personal Agent* (*PA*) and the *Organization Provisioning Manager* (*OPM*) play key roles in the context of this paper. The *Personal Agent* is associated with each human member of the organization. In addition to being the interface between its *User* and the system, the *PA* can support her in any role required by the organization (see [4] for an extended discussion). The *OPM*, on the other hand, is an entity responsible for organizational resource management. In this capacity the *OPM* has access to profiles of all resources present in the organization.

To focus our attention let us introduce two scenarios depicted in Figure 1. In the *Grant Assistant System* (*GAS*), the *OPM* of a university (or a research institute) receives a grant announcement and its role is to deliver it to these and only these *PA*s that represent *Users* that may be interested in exactly this announcement. In other words, the *OPM* has to decide who (which *PA*) should receive a given announcement, based on ontologically demarcated profiles describing faculty in the university (researchers in the institute). Here, we assume that the announcement is a resource that has already an assigned profile based on the internal domain ontology (note that specifying entity inside, or outside, of the system that performs profile demarcation is of no importance here).

In the *Duty Trip Support* (*DTS*) scenario, the *OPM* undertakes the role of a *Travel Assistant*. As can be seen in [4] (specifically, see the sequence diagram presented there) the *OPM* is contacted two times while the *PA* helps its *User* during the Duty Trip related activities. First, when the traveler requests a list of other institutions/persons she can visit during her trip. Second, when a specific advice is sought as far as accommodations are concerned (e.g. hotel and/or

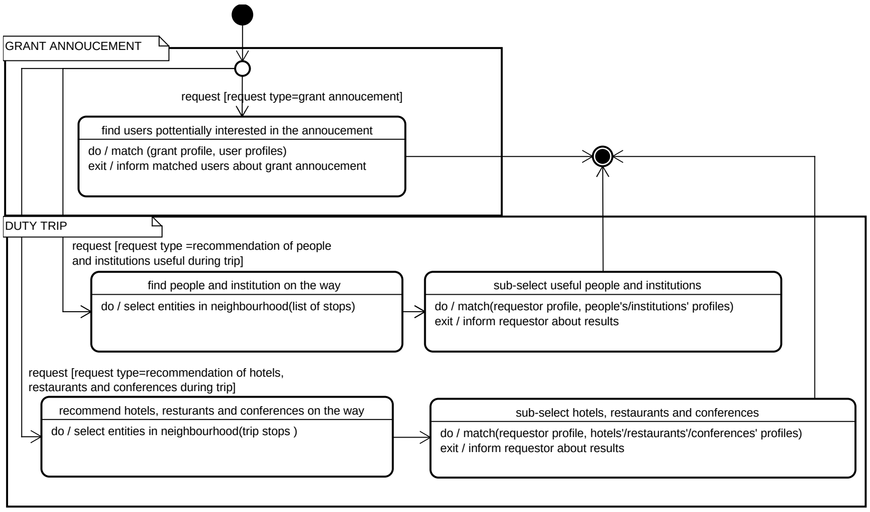

Fig. 1. Matching scenarios in the proposed system

restaurant advice). In both cases the *OPM* has to search the (local) database of ontologically demarcated data to find the pertinent information. For instance, let us assume that a researcher is to travel to a conference that is going to take place in Poznan, Poland. In this case the first question is: which universities/individuals located "close-enough" to Poznan, she may want to visit during her trip. The second question is: where should she stay and eat while in Poznan.

Let us now describe how the desired results are to be obtained in both scenarios. Before we proceed let us stress that we assume that all data *within* a given organization is demarcated utilizing a common (for that organization) ontology. Therefore, in what follows we do not have to deal with matching differing and potentially incompatible (external) ontologies. All that we are interested in is: how to establish "distances between resources" within a single ontology and how to use this information in the above described scenarios.

3 Defining profile of a resource

In the proposed system every entity is conceptualized as a *resource*; regardless of it being an employee of the organization, or a laptop computer. Since we use two ontologies (organizational and domain specific), resource profile is very likely going to utilize both of them. The organizational ontology allows to define structure of the organization and place a resource within this structure, while domain specific ontologies allow to place the resource within the "area of expertise" of the organization [21]. For instance, a mass spectrometer, may be assigned to a given research laboratory (with a specific list of human resources that have access to it), but also be demarcated utilizing an "ontology of chemistry." Let us note that most profiles defined in our system are likely not only to be demarcated in terms of organizational and domain ontologies, but also to have an "access specification." Here, consider the fact that while for most (grant) announcements anyone should be able to read them, access to modify (write or delete) grant announcements should be restricted to authorized entities.

When considering the resource profile we also follow our work in the design of the Travel Support System (TSS) [7]. There, a profile of a human was defined as a set of data about a given individual and his opinions which reflect his attitude toward certain features of objects, or toward objects demarcated within the system [5]. This approach allowed us to deduce the potential interest of a person in a Korean restaurant knowing that she likes meat dishes and dislikes vegetarian cuisine. While, here, we consider both human and non-human resources, for the time being we opt for conceptualizing profiles similarly to the way it was done in the TSS. However, it is still possible that our approach will become readjusted as we continue to analyze project requirements and various development options.

3.1 Defining human resource profile

Let us now consider a sample profile of a human resource. Specifically, a professional experience related fragment of a profile of an employee of a Research Institute in East Asia could be defined as (note that names of disciplines and their codes are standardized for a given Organization and Country, e.g. considering a South Korean Institute–as in the example below, it is likely to be the standard of the Korea Science and Engineering Foundation [12]):

```
:SamplePersonalProfile a onto:ISTPersonalProfile;
  onto:belongsTo resourceNamespace:someHumanResource;
  onto:id "1234567890"^^xsd:string;
  person:name "John Doe"^^xsd:string;
  person:gender person:Male;
  person:birthday "1982-01-01T00:00:00"^^xsd:dateTime.
:SampleProfile a onto:ISTExperienceProfile;
  onto:belongsTo resourceNamespace:someHumanResource;
  onto:doesResearchInFields
    scienceNamespace:Aerodynamics-30501,
    scienceNamespace:Marine_Hydrodynamics-30505,
    scienceNamespace:Turbulence-30412;
  onto:knowsFields
    [a onto:Knowledge;
    onto:knowledgeObject scienceNamespace:Aerodynamics-30501;
    onto:knowledgeLevel "0.75"^^xsd:float],
    [a onto:Knowledge;
    onto:knowledgeObject scienceNamespace:Marine_Hydrodynamics-30505;
    onto:knowledgeLevel "0.40"^^xsd:float],
    [a onto:Knowledge;
    onto:knowledgeObject scienceNamespace:Turbulence-30412;
    onto:knowledgeLevel "0.90"^^xsd:float];
  onto:managesProject :project1.
:project1 a onto:ISTProject;
  onto:managedBy resourceNamespace:someHumanResource;
  onto:period
    [a onto:Period;
    onto:from "2008-06-01T00:00:00"^^xsd:dateTime;
    onto:to "2009-05-31T00:00:00"^^xsd:dateTime];
 onto:fieldsRef scienceNamespace:Aerodynamics-30501;
 onto:projectTitle ''Very Interesting Aerodynamics Project''^^xsd:string.
```

From the example we can tell that a person identified as *someHumanResource* specializes in *Aerodynamics* and his *level of knowledge* is identified as 0.75 (for more info about assigning levels of skills, or more generally "temperature" to a feature, see [5, 7, 9]), *Marine Hydrodynamics* (level of knowledge identified as 0.4), and *Turbulence* (level of knowledge 0.9). Additionally, this person is scheduled to manage a project entitled: "Very Interesting Aerodynamics Project," which starts on June 1st, 2008 and ends on May 31st, 2009.

Let us note, that the scientific interests of a given employee (*onto:knowsFields* in the above example) can be replaced by professional skills describing worker in any discipline. For instance, they could as well be used to specify that a given programmer has knowledge of Java (level 0.7), C++ (level 0.5), Oracle (level 0.65), etc. In this way the proposed approach is both robust and flexible.

3.2 Defining non-human resource profile

To illustrate how we plan to introduce and utilize profiles of non-human resources, we present a fragment of a profile of a Duty Trip Report utilized within the *DTS* scenario (note that, as any document in the organization, the Duty Trip Report is also a resource with its own profile).

```
:SampleDTProfile a onto:DTProfile;
        onto:belongsTo resourceNamespace:SomeDT;
        onto:employee resourceNamespace:someHumanResource;
        onto:status dtStatusNamespace:Application
        onto:destination geo:AmesdaleCity;
        onto:period
                [a onto:Period;
                onto:from "2008-02-01T00:00:00"^^xsd:dateTime;
                onto:to "2008-02-12T00:00:00"^^xsd:dateTime].
```

In the listing above *SampleDTProfile* is a profile of resource represented by *SomeDT*. In our example the latter is a Duty Trip Report resource. The employee who this profile directly refers to is represented by the *someHumanResource*. Hence, we can tell that a person represented in the system as *someHumanResource* applied for a duty trip (*SomeDT*). The current status of that Duty Trip Report is *Application* and the trips destination is Amesdale, Canada. The researcher intends to stay there for twelve days (from February 1st till February 12th, 2008). Again, properties of the *DTProfile* refer to the system schemas (organization and domain ontologies) and fulfill the data model requirements set by the Duty Trip Support System which we develop (see also [21]). Please note that, in order not to overly complicate the example, the snippet above does not cover all properties of the *DTProfile* class defined in [21]. Following listing introduces context of the *SampleDTProfile* instance and serves as geospatial information data for the purpose of our example:

```
geo:CanadaCountry a onto:Country;
        onto:name "Canada"^^xsd:string.
geo:QuebecArea a onto:Area;
        onto:name "Quebec"^^xsd:string;
        onto:isInCountry :CanadaCountry.
geo:OntarioArea a onto:Area;
        onto:name "Ontario"^^xsd:string;
        onto:isInCountry :CanadaCountry.
```

```
geo:AnnavilleCity a onto:City;
        onto:name "Annaville"^^xsd:string;
        onto:long "-72.433"^^xsd:float;
        onto:lat "46.217"^^xsd:float;
        onto:isInCountry :CanadaCountry;
        onto:isInArea :QuebecArea.
geo:AnjouCity a onto:City;
        onto:name "Anjou"^^xsd:string;
        onto:long "-73.533"^^xsd:float;
        onto:lat "45.6"^^xsd:float;
        onto:isInCountry :CanadaCountry;
        onto:isInArea :QuebecArea.
geo:AmesdaleCity a onto:City;
        onto:name "Amesdale"^^xsd:string;
        onto:long "-92.933"^^xsd:float;
        onto:lat "50.017"^^xsd:float;
        onto:isInCountry :CanadaCountry;
        onto:isInArea :OntarioArea.
```

Here we defined *CanadaCountry* as an instance of the *Country* class, *QuebecArea* and *OntarioArea* as an instances of the *Area* class; both of them are regions of the *CanadaCountry*. Additionally, three instances of the class *City* were defined: *AnnavilleCity*, *AnjouCity* and *AmesdaleCity*. The first two are located in the *QuebecArea* whereas the last one is in the *OntarioArea*. Please note that only coordinates of cities are given. On the other hand, countries are represented in our system as sets of regions (areas) and these are in turn represented as sets of cities and are located in a particular country. Let us note that we do not claim that such a representation of data in our geospatial information support subsystem is the most efficient solution to the problem, but we may assume that it is sufficient enough for the purpose of our information provisioning system and the Duty Trip Support application.

In order to illustrate the way that our system is going to work, let us introduce one more example of a non-human resource, which is an information about a conference.

```
:SampleConferenceProfile a onto:ConferenceProfile;
  onto:belongsTo resourceNamespace:SomeConference;
  onto:primaryFields
        scienceNamespace:Aerospace_Ship_and_Ocean_Engineering-30500,
        scienceNamespace:Mechanical_Engineering-30400;
  onto:specialSessionField
        scienceNamespace:Acoustics_and_Noise-30405;
  onto:takesPlaceAt geo:AnnavilleCity;
  onto:period
        [a onto:Period;
        onto:from "2008-02-07T00:00:00"^^xsd:dateTime;
        onto:to "2008-02-09T00:00:00"^^xsd:dateTime. ].
```

The conference introduced here, is going to be held in the second week of February 2008 in Annaville, Canada. Here, the location of the conference (object of the *takesPlaceAt* property) references the geospatial information which was introduced in the previous listing. As far as the conference is concerned, its main topics are *Aerospace*, *Ship and Ocean Engineering* and *Mechanical Engineering*. Furthermore, a special session is planned to cover *Acoustics and Noise*.

Let us now propose how the above introduced resource profiles can be utilized in our system.

4 Calculating distances between resources

From the scenarios described above and summarized in Figure 1, we can easily see the need for resource matching (finding distances between two or more resources). To focus our attention, let us present a few examples of types of resource matching operations that have to be implemented in our system (this list is not intended to be exhaustive, but rather to point to some classes of needed resource matching and/or distance calculations):

1. computing distance between two geographical locations, to be able to establish if a given location is close-enough to the place where the employee is to travel,
2. matching a non-human resource (e.g. a grant, hotel, restaurant, conference) with a human-resource, to find if a person who is planning a trip could be interested in a given nearby located conference, or if an employee is potentially interested in a grant announcement,
3. matching two human resources to find out who are the researchers that a person planning a trip may be interested in visiting.

Upon further reflection it is easy to notice that the way the distance between resources should be calculated depends on types of objects which are arguments of calculations. For example, the distance between value of *onto:destination* property of the *SampleDTProfile* and the value of the *onto:takesPlaceAt* of the *ConferenceProfile* class instance will be calculated in a different way than the distance between values of *onto:period* properties of the *SampleDTProfile* and the *ConferenceProfile* class instances. The following object types that appear in our work can be distinguished, based on different approach to calculate their distance (examples of specific calculations involve the above presented ontology snippets):

1. Objects which represent geographical locations—distance between the *onto:destination* property range of the *onto:DTProfile* class and the *onto: takesPlaceAt* of the *ConferenceProfile* class.
2. Numeric objects—distance between *onto:long* property values.
3. Date objects—distance between *onto:from* and the *onto:from* (or the *onto: to*) property values.
4. Enumerable objects—distance between *onto:primaryFields* property values/ range of the *ConferenceProfile* and the range of *onto:doesResearchInFields* property of the *HRProfile* class.

Let us now discuss proposed approaches to distance calculations/resource matching for the four distinguished classes of properties.

4.1 Location based calculations

City, *country* and *area* are classes designed to represent geographical locations which may be visited by the *User*. These classes have properties which allow to build a tree structure of countries, areas and cities. For instance, *CanadaCountry*, *QuebecArea*, *AnnavilleCity*, *AmesdaleCity* and *AnjouCity* were samples of geo-locations introduced above. They represent a part of an administrative division of Canada. First level in our structure is a country, the second level is an area and finally city (Annaville and Anjou) is the third one. Available properties allow to query for neighbor (adjacent) instances of the same class. This approach requires access to administrative divisions of the world data, otherwise it may be of little value in terms of facilitating a location based advice. Apart from the administrative division tree, these classes allow to describe actual geo-coordinates of objects. Location based advising can be performed by calculating object's distances using the general formula (*long* - longitude, *lat* - latitude, *alt* - altitude):

$$\sqrt{(long_0 - long_1)^2 + (lat_0 - lat_1)^2 + (alt_0 - alt_1)^2}$$

Note that in most business travel scenarios the altitude (alt_0 and alt_1) is of little relevance and can be omitted. In the case of searching for a nearby conference or an organization which may be visited during a Duty Trip, performance issues have to be considered. Calculating distance according to the above formula from data in a semantic (Jena-based [11]) repository is currently very likely to be inefficient. However, we have chosen the PostgreSQL database for the persistence layer of the GIS information [16]. This database engine supports GIS processing and allows to perform geospatial calculations with a designated component. Instead of calculations involving directly the semantically described data, we propose to create a link between (a) semantically described objects which represent geographical locations and have been described using their coordinates, and (b) the copy of the object's geo-information represented within, and calculated using, the PostgreSQL GIS engine. Obviously, similar calculations can be performed not only for conferences and/or institutions, but also for all other geo-objects (e.g. restaurants and hotels) as their coordinates are described in the same way as cities (hotel, restaurant and city are subclasses of the *Spatial Thing* class in our travel ontology; see [7, 22]). Therefore, the *DTS* system will be able to provide at least the following geo-info-based advice:

1. Location notes and tips (textual information about a location which was added to Duty Trip Reports - class in the ontology: *Location Specific Notes*),
2. Organizations and people that can be visited (objects of *Organization Contact* and *Contact Person* classes, these objects are created by the employees during the Duty Trip Report's creation),
3. Information about nearby conferences of possible interest (based on location of the trip and the conference as well as on the personal interest and conference topics),
4. Hotels and restaurants (based on the *Hotel* and *Restaurant* TSS ontology classes),
5. Car rental and golf courses (ontology extensions based on the OTA specification [15]).

4.2 Numeric and date object calculations

Computing distance between numeric and date object is rather obvious. The distance will be represented by the result of difference operation on these objects. In the first case, the result will be a number, in the latter case the result will be a time period (e.g. of a stay in a given place). Note that currently most major programming languages provide date calculation support hence we believe this issue should not be discussed in more detail (assuming there are no problems with date representation and deserialization).

4.3 Enumerable object calculations

In case of ontology, enumerable values can be more complex than *enums* known from popular programming languages. In an ontology, class instances can also be enumerable values. In that case complex structures can be constructed, representing relations between objects. For instance, presented above *ScientificField* class falls under the OWL *oneOf* restriction, however each instance of that class has property values which refer to other instances of that class. This results in a graph-like structure of enumerable values.

To calculate distance between two object of enumerable type, let us note first that if the structure of *enum* values is flat (plain list with no relations between objects) it can be assumed that the distance is 0 if the values match, otherwise it equals to 1. An example of such simple enumerable is the *Gender* property, which is utilized in the human resource profile. Here we have two values: *Male* and *Female* and if they match the distance is 1, and 0 otherwise.

Let us now present a method for calculating distance between class instances which involve transitive, non-symmetric properties, which is a simplified case of method introduced in [17]. Here, a path in a directed graph is calculated for all relations. Let us assume that R is such a transitive, not symmetric relation (*property* in the OWL notation). Then the distance between two vertices of a graph of relation R: v_0 and v_k ($dist_R(v_0, v_k)$) is calculated according to the following algorithm:

1. If there exists $path_R(v_0, v_k)$ in the graph of relation R, then the shortest one can be found and
$$dist_R(v_0, v_k) = length(shorthestPath_R(v_o, v_k));$$
otherwise go to 2nd step.
2. Let $X = \{x : path_R(x, v_0) \text{ and } path_R(x, v_k) \text{ exist}\}$. Find such $y \in X$, that $length(path_R(y, v_0))$ is minimal among all vertices belonging to X (i.e. this is the shortest path):
$$dist_R(v_0, v_k) = 10^{length(path_R(y,v_0))} + length(shorthestPath_R(y, v_k))$$

Let us now describe calculation of distance between research interests of a human resource and conference topics, while utilizing examples introduced above. According to the proposed algorithm the following distance values can be found (here we calculate all-against-all distance values):

```
$dist_SF = path_isSubfieldOf$
$dist_SF$(Aerodynamics-30501, Aerospace_Ship_and_Ocean_Engineering-30500)=10
$dist_SF$(Aerodynamics-30501, Mechanical_Engineering-30400)=101
$dist_SF$(Aerodynamics-30501, Acoustics_and_Noise-30405)=102
$dist_SF$(Marine_Hydrodynamics-30505, Aerospace_Ship_and_Ocean_Engineering-30500)=10
$dist_SF$(Marine_Hydrodynamics-30505, Mechanical_Engineering-30400)=101
$dist_SF$(Marine_Hydrodynamics-30505, Acoustics_and_Noise-30405)=102
$dist_SF$(Turbulence-30412, Aerospace_Ship_and_Ocean_Engineering-30500)=101
$dist_SF$(Turbulence-30412, Mechanical_Engineering-30400)=10
$dist_SF$(Turbulence-30412, Acoustics_and_Noise-30405)=11
```

These values allow us to utilize a number of strategies to establish "closeness" of two resources. The simplest one would be, if for any two properties the distance is below a certain threshold, then a conference should be recommended as potentially interesting. Note that distance could be also scaled by the level of knowledge of the specialist in the field. However more involved considerations are also possible. In this context let us note that values of the $dist_R(v_0, v_k)$ function allow us to specify how far are the graph nodes located from each other in terms of a transitive, not symmetric relation R. In the case of research specialization modeling relation we can assume that the maximum length of $path_R(v_0, v_k)$ is 9. In our ontology an example of such relation is the *isSubfieldOf* property of the *ScientificField* class, where the maximum length of $path_SF(v_0, v_k)$ is 2. Additionally, infinite distance is not considered. With such assumptions we are able to distinguish following groups of conclusions which can be drawn from the function values:

1. If $dist_R(v_0, v_k) = 0$, then $v_0{=}v_k$
2. If $dist_R(v_0, v_k) = n$ and $0 < n < 10$, then $R(v_0, v_k) = true$ and v_k is n-deep specialization of v_0
3. If $dist_R(v_0, v_k) = n$ and $n = 10^k, k > 0$, then $R(v_0, v_k) = false$ and v_0 is k-deep specialization of v_k
4. If $dist_R(v_0, v_k) = n$ and $10^k < n < 10^(k + 1), k > 0$, then $R(v_0, v_k) = false$ and v_0 is $n - 10^k$-deep specialization of k-deep specialization of v_k

For instance, if

$$dist_SF(Turbulence - 30412, Acoustics\_and\_Noise - 30405) = 11,$$

we may say that there is a node of which both *Turbulence* and *Acoustics and Noise* are direct specializations.

These observations allow us to develop a number of reasoning scenarios that utilize not only information about numerical distance, but also following form it knowledge about the structure of relations. Developing a reasoning engine utilizing this information is next step in our work.

Let us note that the proposed algorithm is a specific case of the method proposed in [17]. Basic difference between them is as follows. The method proposed in [17] assumes existence of multiple relations linking any class instance (node) from a single node and merging edges which represent relations of the same direction between the same nodes; thus, the distance is computed including all properties (relations) of classes (concepts). The algorithm presented above, on the other hand, is restricted to one selected property (relation) of a class (concept) and an inverse of the selected property. This pair represent generalization and specialization relations between concepts. We claim that the algorithm presented here can be substituted for the algorithm of [17] by adjusting appropriate weights to concepts relations. Specifically, used here weirths of 1 for specialization and 10 for generalization.

5 Concluding Remarks

In this paper we have presented preliminary results concerning resource matching in a virtual organization. Specifically, we have considered comparisons between specific values within ontologically demarcated resources (geographical distance, date and numeric objects, as well as enumerated types). In the next phases of our work we plan to apply these methods in automatic resource matching. Specifically, we will develop a matching engine that will use rules of resource relevance and as a result will be able to calculate the distance between them depending on the matching query context (i.e. scenario and types of compared resources).

An important problem which has to be taken into account is the performance of proposed operations. This is especially the case as we are already aware of deficiencies of the *Jena* persistence and query engine. This is one of the reasons for the planned mixed environment in which the PostgreSQL GIS engine is to be used. However, the final determination of the implementation details will have to be made on the basis of experiments performed with realistic datasets.

References

1. Biesalski, E., Abecker, A.: Human Resource Management with Ontologies. In: *Wissensmanagement. Professional Knowledge Management, Third Biennial Conference, WM 2005, Kaiserslautern, Germany, 2005, Revised Selected Papers*. LNAI 3782, Springer, 2005, 499–507.
2. Bizer, C. Heese, R., Mochol, M., Oldakowski, R., Tolksdorf, R. and Eckstein, R.: The Impact of Semantic Web Technologies on Job Recruitment Processes. In: *Proc. International Conference Wirtschaftsinformatik (WI 2005), Bamberg, Germany*, 2005.
3. Ganzha, M., Paprzycki, M., Popescu, E., Bădică, C., and Gawinecki, M.: Agent-Based Adaptive Learning Provisioning in a Virtual Organization. In: *Advances in Intelligent Web Mastering. Proc.AWIC'2007*, Fontainebleu, France. *Advances in Soft Computing* 43, 25–40, Springer, 2007
4. Ganzha, M., Paprzycki, M., Gawinecki, M., Szymczak, M., Frackowiak, G., Bădică, C., Popescu, E., Park, M.-W.: Adaptive Information Provisioning in an Agent-Based Virtual Organization—Preliminary Considerations. In: *Proceedings of the SYNASC Conference*, IEEE CS Press, in press.
5. Gawinecki, M., Modelling User on the Basis of Interactions with a WWW Based System, Master Thesis, Adam Mickiewicz University, 2005
6. Gawinecki, M., Gordon, M., Paprzycki, M., Vetulani, Z.: Representing Users in a Travel Support System. In: Kwasnicka, H. et. al. (eds): *Proceedings of the ISDA 2005 Conference*, IEEE Press, 2005, 393–398.

7. Gawinecki M., Gordon M., Nguyen N.T., Paprzycki M., Szymczak M., RDF Demarcated Resources in an Agent Based Travel Support System. In: M. Golinski et. al. (eds.), Informatics and Effectiveness of Systems, PTI Press, Katowice, 2005, 303-310
8. Gawinecki, M., Gordon, M., Nguyen, N.T., Paprzycki, M., Zygmunt Vetulani, Z.: Ontologically Demarcated Resources in an Agent Based Travel Support System. In: R. K. Katarzyniak (ed.): *Ontologies and Soft Methods in Knowledge Management*, Advanced Knowledge International, Adelaide, Australia, 2005, 219–240.
9. Greer, J., McCalla, G., Student modeling: the key to individualized knowledge based instruction. NATO ASI Series F, 125, 1993
10. HR-XML Consortium. http://www.hr-xml.org/.
11. Jena Semantic Web Framework, http://jena.sourceforge.net/
12. Korea Science and Engineering Foundation http://www.kosef.re.kr/english_new/index.html
13. Mochol, M., Wache, H., and Nixon, L.: Improving the Accuracy of Job Search with Semantic Techniques. In: *Business Information Systems*. LNCS 4439, Springer, 2007, 301–313.
14. M. Montaner, B. López, and J. L. de la Rosa, A taxonomy of recommender agents on the Internet, *Artif. Intell. Rev.*, 19(4), 2003, 285–330.
15. Open Travel Alliance, http://www.opentravel.org/
16. PostGIS : HOME, http://postgis.refractions.net/
17. Rhee, S. K., Lee, J., Park, M. W.: Ontology-based Semantic Relevance Measure, In: S. Kim et.al. (eds.), Proceedings of the The First International Workshop on Semantic Web and Web 2.0 in Architectural, Product and Engineering Design, 6 pages, 2007 http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-294/
18. Schmidt, A., Kunzmann, C.: Towards a Human Resource Development Ontology for Combining Competence Management and Technology-Enhanced Workplace Learning. In: *OTM Workshops (2). On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops*. LNCS 4278, Springer, 2006, 1078–1087.
19. Semantic Web. http://www.w3.org/2001/sw/.
20. Szymczak, M., Frackowiak, G., Ganzha, M., Gawinecki, M., Paprzycki, M., Park M.-W.: Resource Management in an Agent-based Virtual Organization—Introducing a Task Into the System. In: *Proc. of the WI/IAT 2007 Workshops*, IEEE CS Press, Los Alamitos, 2007, in press.
21. Szymczak, M., Frackowiak, G., Ganzha, M., Gawinecki, M., Paprzycki, M., Park M.-W.: Adaptive Information Provisioning in an Agent-Based Virtual Organization—Ontologies in the System, submitted for publication
22. Szymczak, M., Gawinecki, M., Vukmirovic, M., Paprzycki M., Ontological Reusability in State-of-the-art Semantic Languages, in: C. Olszak et. al. (eds), Knowledge Management Systems, PTI Press, Katowice, 2006, 129-142
23. Training. http://en.wikipedia.org/wiki/Training.
24. Tzelepis, S., Stephanides, G.: A conceptual model for developing a personalized adaptive elearning system in a business environment. In: *Current Developments in Technology-Assisted Education*, Formatex Publishing House, vol.III, 2006, 2003–2006.
25. Wooldridge, M.: *An Introduction to MultiAgent Systems*, John Wiley & Sons, 2002.