Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 243–250
ISBN 978-83-60810-14-9
ISSN 1896-7094

Ontological Matchmaking in a *Duty Trip Support* Application in a Virtual Organization

Michal Szymczak,
Grzegorz Frackowiak,
Maria Ganzha and
Marcin Paprzycki
System Research Institute,
Polish Academy of Sciences,
Poland
Email: maria.ganzha@ibspan.waw.pl

Sang Keun Rhee,
Jihye Lee,
Young Tae Sohn,
Jae Kwan Kim
Yo-Sub Han and
Myon-Woong Park
Korea Institute of Science and Technology,
Seoul, Korea

***Abstract*—In our work, an agent-based system supporting workers in fulfilling their roles in a virtual organization has as its centerpiece ontologically demarcated data. In such system, ontological matchmaking is one of key functionalities in provisioning of personalized information. Here, we discuss how matchmaking will be facilitated in a *Duty Trip Support* application. Due to the nature of the application, particular attention is paid to the geospatial data processing.**

I. I

CURRENTLY, we are developing an agent-based system supporting resource management in a virtual organization. While it is often assumed that the notion of virtual organization should be applied when workers are geographically distributed ([1]), we do not make this assumption. Instead, we consider as "virtualization" process in which a real organization is "mapped" into a virtual one, where virtual can be understood as "existing electronically" (see, also [2]). In such virtual organization (1) its structure is represented by software agents and their interactions, while (2) the organization itself and its domain of operation are ontologically described. Here, we recognize need for (i) an ontology of an organization (e.g. specifying who has access to which resource, or which department does a given person work for), and (ii) a domain specific ontology (e.g. ontology of a car repair shop, specifying areas of expertise and skills of individual workers); see [3], [4], [5], [6] for summary of results obtained thus far.

One of main reasons for utilizing ontologies is that they allow for application of semantic reasoning. The aim of this paper is to describe how a specific type of such reasoning—ontological matching—can be used in the context of an applications currently under development. While in [7] we have considered matchmaking involved in the *Grant Announcement Support*, here we consider a *Duty Trip Support*. One of the key concepts that we will explore is geospatial matchmaking. To this effect, in the next section, we introduce the *Duty Trip Support* (*DTS*) application, following with an introduction to matchmaking processes that take place in the system. Next, we discuss specific matchmaking taking place in the *DTS*. In this section we also present in detail our ontological matchmaking algorithm.

II. *Duty Trip Support*

In our earlier work [5] we have introduced a scientist, Mr. Jackie Chan, employed in a Science Institute in Aberdeen, Hong Kong, China, who goes on a duty trip to Finland and will utilize a *Duty Trip Support* (*DTS*) application. It is expected that the *DTS* will be able to suggest to travelers places to stay and eat (based on their personal preferences and experiences of other travelers, e.g. from the same institution). This is an example of personalized information delivery that takes into account cultural and dietary differences between, for instance, Japan and Germany. A complete description of proposed functionalities and processes involved in the *DTS* application (including a complete UML sequence diagram) can be found in [8], [4], [5]. Here we focus our attention on ontological matchmaking utilized in the *DTS*. Note that majority of examples listed here (ontology classes and properties, in particular) are based on our previous work (see, [4], [5] for more details). Doing so, allows us to shed more light on these examples and matching processes taking place in the system, which were only outlined before, while preserving continuity and building a complete picture of the process.

Let us now reintroduce ontology class instance samples and start from a listing of *City* and *Country* instances which include geospatial information. Two countries (China and Finland) and three cities (Aberdeen, Oulu and Rovaniemi) are listed (each city located in one of the two countries).

```
geo:FinlandCountry a onto:Country;
        onto:name "Finland"^^xsd:string.
geo:ChinaCountry a onto:Country;
        onto:name "China"^^xsd:string.
geo:OuluCity a onto:City;
        onto:name "Oulu"^^xsd:string;
        onto:long "25,467"^^xsd:float;
        onto:lat "65,017"^^xsd:float;
        onto:isInCountry :FinlandCountry.
geo:RovaniemiCity a onto:City;
        onto:name "Rovaniemi"^^xsd:string;
        onto:long "25,8"^^xsd:float;
        onto:lat "66,567"^^xsd:float;
```

```
        onto:isInCountry :FinlandCountry.
geo:AberdeenCity a onto:City;
        onto:name "Aberdeen"^^xsd:string;
        onto:long "114,15"^^xsd:float;
        onto:lat "22,25"^^xsd:float;
        onto:isInCountry :ChinaCountry.
```

Next, let us recall the sample employee (Mr. Chan). As discussed in [4], [5], each employee is associated with several profiles. Note, however, that some resources introduced in this document have only a single profile assigned. In general, resource detailed information is stored in its profile in order to facilitate adaptability in the system, which assumes that profile extensibility, robustness and resource access control are required. Therefore, saving resource attributes within profiles allows to easily extend and adapt individual resource records (see, also [9]). Below we present a snippet based on the *Employee Profile*, which consists of a *Personal Profile* and an *Experience Profile*:

```
:Employee\#1 a onto:ISTPerson;
    onto:id "1234567890"^^xsd:string;
    onto:hasProfile (:Employee\#1PProfile,
                     :Employee\#1EProfile),
    onto:belongsToOUs (:GOU).
:ResearchOU a onto:OrganizationUnit;
    onto:name ''Researchers Organization
                        Unit''^^xsd:string.
```

In this example the *Employee#1PProfile*—the *Personal Profile*, presented next—describes the "human resource (HR) properties" of an employee. In what follows we use only basic properties: *fullname*, *gender* and *birthday*; as well as the *belongsToOUs* property, which indicates Mr. Chan's position in the organization (the *Organizational Unit* he works for). Let us stress that a complete list of HR properties is organization-dependent and is instantiated within the ontology of a given organization.

```
:Employee\#1PProfile a onto:ISTPersonalProfile;
  onto:belongsTo :Employee\#1;
  person:fullname ''Yao Chan''^^xsd:string;
  person:gender person:Male;
  person:birthday
        ''1982-01-01T00:00:00''^^xsd:dateTime.
```

The second profile of *Employee#1* (Mr. Chan) that we have introduced, is the *Experience Profile* that demarcates his specialization in terms of fields of knowledge and project experience. Here, codes for the fields of knowledge specification originate from the KOSEF (Korea Science and Engineering Foundation) [10]. Obviously, *any* classification of fields of knowledge/expertise could be applied here (appropriately represented within the ontology of an organization).

```
:Employee\#1EProfile a onto:ISTExperienceProfile;
  onto:belongsTo :Employee\#1;
  onto:doesResearchInFields
    scienceNamespace:Volcanology-13105,
    scienceNamespace:Paleontology-13108,
    scienceNamespace:Geochronology-13204;
  onto:knowsFields
    [a onto:Knowledge;
    onto:knowledgeObject
        scienceNamespace:Volcanology-13105;
    onto:knowledgeLevel "0.25"^^xsd:float],
    [a onto:Knowledge;
    onto:knowledgeObject
        scienceNamespace:Paleontology-13108;
    onto:knowledgeLevel "0.15"^^xsd:float],
    [a onto:Knowledge;
    onto:knowledgeObject
        scienceNamespace:Geochronology-13204;
    onto:knowledgeLevel "0.90"^^xsd:float];
  onto:managesProjects (:Project1).
```

According to the *ISTExperience profile*, Mr. Chan specializes in *Volcanology*, *Paleontology* and *Geochronology*. Level of knowledge in each of these areas is expressed as a sample real value; respectively: 0.25, 0.15, 0.9. Here, we assume that values describing the level of knowledge in specific fields are a result of self-assessment of an employee. However, in [11] we have proposed mechanisms for human-resource adaptability, which can be utilized to automatically (or semi-automatically) adapt level of knowledge on the basis of, for instance, work and training history of the employee. Furthermore, note that professional test (existing in most fields) can also be directly used as a method for knowledge level assessment. Now, *Employee#1* who is described with that profile manages project *Project1*. It is a scientific project in *Volcanology* (see below).

```
:Project1 a onto:ISTProject;
 onto:managedBy :Employee\#1;
 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:Volcanology-13105;
 onto:projectTitle ''Very Important Volcanology
                Scientific Project''^^xsd:string.
```

To be able to illustrate matching processes taking place within the *Duty Trip Support* application, we will now introduce instances of a *Contact Person* (*:ContactPerson#1*) and a *Duty Trip Report* (*:DTR#1*); see [5], [7] for additional details).

```
:ContactPerson\#1 a onto:ContactPerson;
  onto:hasProfile :ContactPersonProfile\#1.
:ContactPersonProfile\#1
                a onto:ContactPersonProfile;
  person:fullname
            ''Mikka Korteleinen''^^xsd:string;
  person:gender person:Male;
  person:birthday
          ''1967-11-21T00:00:00''^^xsd:dateTime;
  onto:doesResearch science:Paleontology-13108,
                    science:Volcanology-13105;
  onto:locatedAt geo:RovaniemiCity;
  onto:belongsTo :ContactPerson\#1.
:ContactPerson\#2 a onto:ContactPerson;
  onto:hasProfile :ContactPersonProfile\#2.
:ContactPersonProfile\#2
                a onto:ContactPersonProfile;
  person:fullname
            ''Juno Viini''^^xsd:string;
  person:gender person:Male;
  person:birthday
          ''1957-01-15T00:00:00''^^xsd:dateTime;
  onto:doesResearch science:Geochronology-13204;
  onto:locatedAt geo:RovaniemiCity;
  onto:belongsTo :ContactPerson\#2.
:DTR\#1 a onto:ISTDutyTripReport;
  onto:hasProfile (:DTRProfile\#1).
:DTRProfile\#1 a onto:ISTDutyTripReportProfile;
  onto:destination geo:OuluCity;
  onto:traveler :Employee\#1;
  onto:status dtStatusNamespace:Application;
  [a onto:Period;
  onto:from ''2008-06-07T00:00:00''^^xsd:dateTime;
  onto:to ''2008-06-19T00:00:00''^^xsd:dateTime.];
  onto:stayedAt hot:OuluRadisonSAS
  onto:expense [a onto:SingleCost;
        ''4000''^^xsd:float;
```

```
onto:expenseCurrency ''USD''^^xsd:string.]
onto:purpose ''Conference''^^xsd:string;
onto:belongsTo :DTR\#1.
```

As we will see, *Contact Person*s will be suggested (though for a different reason) by our system as someone who Mr. Chan should visit. First, Mikka Korteleinen, is defined through the *ContactPerson#1* and the *ContactPersonProfile#1* objects. The latter object defines Mr. Korteleinen's field of specialization to be *Paleontology* and *Volcanology*. Here, the level of expertise of Mr. Korteleinen is not specified, because it is assumed that such data is a result of self-assessment of the person, or processes taking place internally in her/his organization (see above); and as such is not available to the *DTS* of Mr. Chan's organization. However, note that the very fact that a potential contact person is in a system is very likely going to be a result of a personal meeting with her/him, followed by an employee introducing the contact information into the system. Such employee, could potentially assess not only area(s) of expertise, but also level of knowledge in each one of them. This possibility will be explored in the future. Now, we can observe that the profile of Mr. Korteleinen informs us that he can be reached in Rovaniemi, Finland. Second, the profile of Mr. Juno Viini was defined. According to this example he can be found in Rovaniemi as well, while his research interest is *Geochronology*. Separately, note that the *Duty Trip Report* is a basic resource associated with any *Duty Trip*. It is defined through the *DTR#1* and the *DTRProfile#1* objects and represents a required set of information associated by the organization with a *Duty Trip*. It describes travel details details, such as:

```
destination: Oulu, Finland,
status: application,
purpose: a conference.
```

Here, we can see that Mr. Chan plans to travel to Oulu, Finland to a conference and he has applied for this *Duty Trip* (in the case his travel is approved, the field *status* will change its value to *approved*). Our aim in this paper is to show how ontological matchmaking can be used to find cities near-by the city where Mr. Chan is to travel to, and person(s) that he may want do consider visiting during his trip.

III. M

A. General idea

What is needed to achieve our goal is to be able to establish measure of distance (similarity) between two (or more) instances of ontologically demarcated data (here, between researchers considered in the context of cities they reside in). Before proceeding, let us note first that in our work we have made an important simplifying assumption. Across all currently developed applications (which are to work *within* an organization), a single ontology is used. Therefore, we do not have to deal with problems related to *ontology matching/integration* (where an attempt is made to establish "common understanding" between two, or more, ontologies; see, for instance [12], [13], [14]). Since our goal is to establish if specific instances of an ontology are "close enough," our objective should be to define a measure of distance representable as a single number (among others, for ease of comparison). This number can be then compared against a threshold to make a decision if objects are relevant to each-other. Note also that for each application there is a specific *Matching Criteria* that represents the "focus" of the matching process (e.g. research interests, eating preferences, or location). In [7] we have described in general terms process of measuring closeness of objects—*Calculating Relevance*, and presented it in the context of the *Grant Announcement Application*. Here a modified version of that algorithm will be proposed. However, we can use the same point of departure and define the *Matching Criteria* as a tuple (quadruple in this case) $\langle x, q, a, g\rangle$, where:

- x is the selected ontology class instance (source object)
- q is a SPARQL query ([15]) which defines a subset of objects that are considered potentially relevant (this is the above mentioned focus of the matchmaking process) and will be matched against the source object x
- $a \geq 0$, specifies threshold of closeness between objects to be judged actually relevant to each-other
- g is a sub-query processed by the GIS subsystem (in general, this parameter can be omitted—if there is no geospatial query involved; or it can be replaced by one or more different criteria; thus the notion that the *Matching Criteria* is a tuple); this part of the system is responsible for finding cities which are located within a specified distance to a specified city; this sub-query is a triple $\langle gr, gc, ga\rangle$, where:
  - gr is an operator which allows to either limit returned number of cities of possible interest (*AMOUNT* condition) or to specify the maximum distance between the gc and the returned cities (*RADIUS* condition)
  - gc is an URI of a city demarcated with properties of the *City* class of the system ontology
  - ga is the parameter of the gr operator ($gr(gc, ga)$); it either specifies the limit of the number of returned cities or the maximum distance between the gc and the returned cities

B. Relevance graph

Calculation of "distance" between instances of ontology is based on a graph structure that represents the underlaying *Jena Ontology Model*. Specifically, this *Model* is interpreted as a directed graph $G = (V, E)$ such that (here, reflexive relations are ignored):

V : set of nodes, representing all instances,

E : set of edges, representing all object properties.

Upon creating edges, the value of the annotation property *voPropertyWeight* of each object property becomes the label of the edge, representing the "distance between two nodes" ("importance of the relationship between two nodes") in the ontology. This concept comes from work of S. Rhee and collaborators (see, [16], [17] for more details), where it was shown how

connections between nodes in the graph representing concepts in the ontology can be weighted according to the importance of their relationships. For instance, suppose for a given researcher we have two relations `doesResearchInFields` and `worksForProjects`; and the system attempts to recommend an additional contact person based on those two relations. Here, `worksForProjects` can be regarded to be more important than `doesResearchInFields` because the fact that two persons work at a given time for a similar project can be considered more important for recommending them to eachother than the situation when two persons declare that they have same (or similar) research interests. Currently, in our system we do not have an automatic way of assigning weights to edges in an ontology, and thus weights equal to 1 will be initially used as a default (which is also the approach suggested in [16], [17]). However, we assume that different weights can (and will in the future) be used in the graph representing the proposed ontology. It is also worthy stressing that such values allow us to naturally deal with concepts that are not directly connected in a directed graph (see, [6] for more details). For the sake of precision it can be thus stated that the relevance graph $G = (V, E)$ becomes $G = (V, E, W)$,

V : set of nodes, representing all instances,

E : set of edges, representing all object properties.

W : "importance weights" assigned to all edges

Let us now observe that while weight values assigned to each edge represent the distance between two nodes in the *Model*, some instances connected by a certain relation (i.e. having the same "importance weight") may not have the same "importance" on the level of individuals. For example, a person may have knowledge/interest in three different subject areas, while his/her knowledge/interest level in each area may be different (see the *ISTExperience profile* example in Section II).

Therefore, we can distinguish two levels of "scaling" of importance of ontological relationships. The first one is "within the ontology" and involves relationships between concepts (nodes in the relevance graph). The second one is on the "level of instances" and specifies importance of specific properties to an individual. Therefore, when the ontological distance is calculated, first we have to take into account ontological distance between concepts and, second, to scale it according to individual "interests" of resources involved in matching. As an example of such process, delivery of personalized information is depicted in figure 1.

Here, we can see an employee within an organization. The ontology of an organization and the domain ontology provide us with a (weighted) relevance graph ($G = (V, E, W)$). At the same time, an ontological instance—the employee profile—allows us to scale specific relations in the ontology according to the employees' interests. Both the relevance graph and the individual profile, together with resources, closeness to which is to be established (selected according to the *Matching Criteria*), are the input to the matching algorithm. As an output we obtain list of resources that are relevant to the employee.

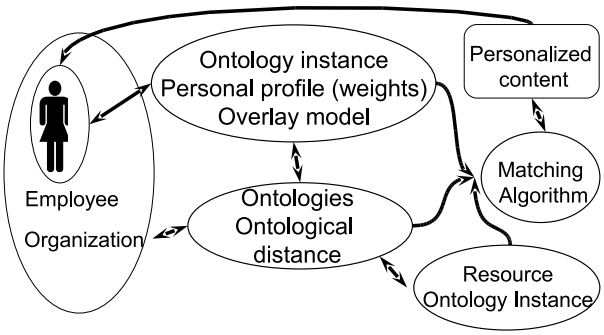

Fig. 1. Top level overview of matchmaking

C. Geospatial Information Subsystem

Before proceeding to describe in detail the matchmaking algorithm and its utilization in the *Duty Trip Support* application, let us make the following comment on the (optional) GIS sub-query. In general, the first four terms of the *Matching Criteria*, above, allow one to find objects relevant to the given *source object* assuming that there is a (directed) path between these objects in the *Relevance Graph* (these objects are linked with each other). However, one of basic requirements of the system under development is that it will provide geospatial information-based recommendations. Observe that the *Relevance Graph* does not provide natural support for distance calculation in terms of geographical localization (as it is represented in our ontology). Specifically, the *Relevance Graph* is built on the basis of nodes that represent ontology class instances, which in turn are linked by edges that represent properties. In our ontology, geospatial attributes are: *longitude*, *latitude*, *altitude* and *inCountry*. While meaning of the first three is obvious, the latter is a relation between instances of the *Country* and the *City* classes that allows us to model "cities being in a country" property.

SPARQL seems to be most recommended technology for querying the RDF demarcated data and it provides support for computing results of mathematical calculations on *Datatype Properties* values of objects defined in the RDF [18]. The latter functionality is necessary for finding RDF demarcated *City* class instances which meet certain distance criteria in terms of geospatial localization. However, designing full GIS support to be performed by the SPARQL engine would require full information about countries and cities of the world to be stored in the semantic storage. Such an approach would overload the semantic storage and severely influence other SPARQL operations which have to be performed on the RDF demarcated data. On the other hand, distances between nodes of the *Relevance Graph* correspond to (scaled) weights of ontology properties that reflect the semantic distance of certain concepts. This distance is not the geospatial distance. Therefore, in order to avoid semantic storage replication/clustering, which would be necessary if we stored all GIS information in the semantic storage, we decided, for the time being, that the GIS sub-query is going to be processed by a dedicated subsystem that allows us to select only related objects which meet criteria defined in

the sub-query. This subsystem returns as a result a list of *City* object URIs which represent cities that meet the geographical localization criteria and, in the case that a particular *City* object does not exist in the semantic storage, it creates the necessary RDF statements. This subsystem is simple and independent and it reduces the volume of the RDF demarcated data stored in the system. Please note that this solution is temporary as we experiment with an alternative that allows to utilize full computational possibilities of SPARQL and keeps RDF data volume as small as possible.

IV. *Duty Trip Support-* M E

As noted, one of important functionalities of the *Duty Trip Support* subsystem is to suggest optional activities of an employee who plans a duty trip. In order to give Mr. Chan advice about possible extensions of his duty trip, first, it is necessary to define appropriate *Matching Criteria*. Next, the delivered advice is a result of matching between the *DTR#1* object and instances of the *ContactPerson* class ([19], [7]). Note that in the future such matching will involve also information about food and accommodations (while other features can also be naturally selected). Overall, the matching process in the *DTS* involves the following steps:

1) Construct Matching Criteria $\langle x, q, a, g\rangle$:
   a) $x = DTR\#1$
   b) $q =$

```
PREFIX onto:
<http://rossini.ibspan.waw.pl/
        Ontologies/KIST/KISTVO>
SELECT ?person
WHERE {?person isa onto:ContactPerson.}
 FILTER (onto:locatedAt
          temp:gisResults-multi).
```

   c) $a = \frac{1}{40}$
   d) $g = [gc, gr, ga]$:

$$gc = OuluCity,$$
$$gr = RADIUS,$$
$$ga = 200.$$

   The *Criteria* defined above can be stated as: find a potentially interesting (in terms of professional interests) person who resides not further than 200 km away from Oulu.
2) Execute the GIS query g. To do this, invoke the GIS interface method which returns references to objects which represent (known to the system) cities located within 200 km distance from the city of Oulu. Results are chosen from all cities for which at least one RDF object in the semantic storage exists. The result in our example is: RovaniemiCity for which the distance from *OuluCity* equals to 173.168 km.
3) Execute an appropriate SPARQL query. In case of the duty trip based advisory [5] this query should limit sought objects only to *ContactPerson* class instances (obviously, in the general case, such a query could seek other entities, e.g. golf courses, or historic castles). An additional SPARQL filter is applied according to the respective *Matching Criteria* part: *onto:locatedAt temp:gisResults-multi*. The matching request processing engine transforms the GIS sub-query results to a valid SPARQL filter and executes the query. In our example the final SPARQL query has the following form:

```
PREFIX onto:
<http://rossini.ibspan.waw.pl/
        Ontologies/KIST/KISTVO>
SELECT ?person
WHERE {?person isa onto:ContactPerson.}
FILTER (onto:locatedAt :RovaniemiCity).
```

   In our example, results of this query are *ContactPerson#1* and *ContactPerson#2* object references. Note that the proposed order of the GIS and the SPARQL query execution may change in the final version of our system, as it largely depends on results of our experiments with designing optimal SPARQL support for the, described above, GIS calculations.
4) Having merged results returned by the GIS component and the SPARQL engine, the relevance can be calculated. Note that the above proposed threshold value $R = \frac{1}{40}$ is a sample value only and is used to illustrate the process; an actual value will be a result of experimental calibration of the system. Specifically, to be able to actually establish a reasonable threshold value, a number of experiments have to be performed. Such experiments require a complete implemented system running and providing explicit and/or implicit user feedback. However, the question of tuning the performance of the proposed approach to a given institution is out of scope of this contribution. Thus the matching process involves:
   a) source instance $URI = DTR\#1$
   b) target objects $URI's = [ContactPerson\#1, ContactPerson\#2]$
   c) relevance threshold: $R = \frac{1}{40}$.

   If the relevance value for any object is above the relevance threshold, such object(s) will be suggested.

A. Calculating relevance

Let us start from considering Figure 2 which presents the overview of relations between *Employee1* and two contact persons *ContactPerson#1* and *ContactPerson#2* via three research fields: *Volcanology*, *Paleontology*, *Geochronology*. These relations are represented in our ontology, as shown in Figure 3 (note that a figure which would include all relations would be too complex to be explanatory).

Figure 2 includes scaling factors (related to professional interests of the *Employee#1*), represented as $W1$, $W2$, and $W3$, which in the proposed algorithm, are used in order to calculate relevance between objects. The values of D represent the ontology property weights for each relation, defined by an annotation property `voPropertyWeight`. On the other hand, Figure 3) presents in some detail links between the *Employee#1* and *ContactPerson#1* resources depicted from the perspective of ontology concepts.

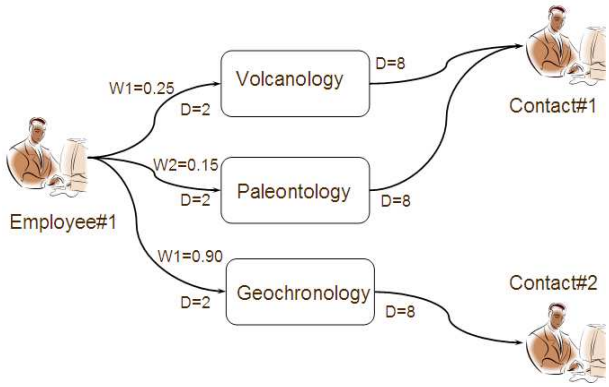

Fig. 2. Employee and foreign contacts

Therefore, based on Figure 2 and the above discussion, we can define three paths from *Employee#1* to the selected contact persons:

Path 1: *Employee#1* → *Employee#1Profile* → *Volcanology* → *ContactPerson#1Profile* → *ContactPerson#1*
Path 2: *Employee#1* → *Employee#1Profile* → *Paleontology* → *ContactPerson#1Profile* → *ContactPerson#1*
Path 3: *Employee#1* → *Employee#1Profile* → *Geochronology* → *ContactPerson#2Profile* → *ContactPerson#2*

Let us assume (see above) that ontology property weights are defined as follows:

$$voPropertyWeight(doesResearchInFields) = 2$$
$$voPropertyWeight(isResearchedBy) = 8$$
$$voPropertyWeight(hasProfile) = 1$$
$$voPropertyWeight(belongsToResource) = 1$$

Now, let us recall the fact that the *Employee#1* has different level of knowledge of these research fields, and the knowledge level (i.e. *weight*) can be applied to the `voPropertyWeight(doesResearchInFields)` value (i.e. *distance*) to obtain a scaled distance value for each individual. Precisely, the scaled relevance value is obtained by multiplying the individual weight value by the inverse of the distance value (i.e. the relevance value):

$$newRelevance = relevance \times weight$$

Since the *relevance value* between two nodes is the inverse of the *distance value*, the new distance is as follows:

$$newDistance = (\frac{1}{distance} \times \texttt{weight})^{-1}$$

This scaling provides personalized relevance results for the *Employee#1*. Here, scaled distance values from *Employee#1* to the three research fields (*Volcanology*, *Paleontology*, *Geochronology*) are 8, $\frac{40}{3}$, $\frac{20}{9}$, respectively. Now, the relevance value for each path is calculated as follows:

$$Rel_{path} = (\sum_{k=1}^{n}(k \times D_k))^{-1},$$

where n is the number of edges in the path and D_k—distance of k-th edge. Thus the relevance result for each path is:

$$Rel_{path1} = \frac{1}{45} = 0.022$$
$$Rel_{path2} = \frac{3}{167} = 0.018$$
$$Rel_{path3} = \frac{9}{301} = 0.030$$

Both Rel_{path1} and Rel_{path2} represent the relevance between *Employee#1* and *ContactPerson#1*, hence we can establish the final relevance value between the two objects by adding relevances of each path. Therefore, the final value of relevance between *Employee#1* and the two contact persons is:

$$Rel_{ContactPerson\#1} = Rel_{path1} + Rel_{path2} = 0.040,$$
$$Rel_{ContactPerson\#2} = 0.030.$$

Note that the calculation process is presented in a simplified way in this paper, however, the detailed algorithm can be found in [16], [17], [20]. Based on the result and the proposed *Criteria*=$\{R \geq \frac{1}{40}$, where R is the relevance threshold$\}$, both *ContactPerson#1* and *ContactPerson#2* will be recommended for *Employee#1*. Let us stress that *ContactPerson#2* is recommended via a single research field (*Geochronology*) whereas *ContactPerson#1* is recommended via combined strength of two research fields, even though the relevance value of each single path is below the threshold. Thus, the relevance measure considers not only a single research field match but also the "multidisciplinary" case. In the future we may consider making the threshold for the multi-pathway relevance to be different than the single-pathway one; as a single strong link is more important than a number of weaker links, but this will be done as a part of system calibration. Overall, as a result of the matching, the following additional duty trip activity is to be suggested:

```
:AdditionalDuty\#1 a onto:ISTDuty;
        onto:destination geo:RovaniemiCity;
        onto:madeContact :ContactPerson\#1.
:AdditionalDuty\#2 a onto:ISTDuty;
        onto:destination geo:RovaniemiCity;
        onto:madeContact :ContactPerson\#2.
:DTRProfile\#1 onto:duty :AdditionalDuty\#1.
:DTRProfile\#1 onto:duty :AdditionalDuty\#2.
```

The *:AdditionalDuty#1* and *:AdditionalDuty#2* objects define activities which are suggested to Mr. Chan who is planning his duty trip represented in the system as the *DTR#1*. Figures 3 and 4 present relations between objects included in the example in this section. Note that these figures omit *ContactPerson#2* and *AdditionalDuty#2* due to the fact that relations of these resources with *Employee#1* are analogical to the relation between *Employee#1* and *ContactPerson#1* and *AdditionalDuty#1*. In Figure 3 we present a path between the *Employee#1* and the *ContactPerson#1* and including them would only make both figures less legible. Finally, in Figure 4 we depict the final relations between the *Employee#1*, the *ContactPerson#1* and the *DutyTrip#1*, after the suggested *AdditionalDuty* is accepted.

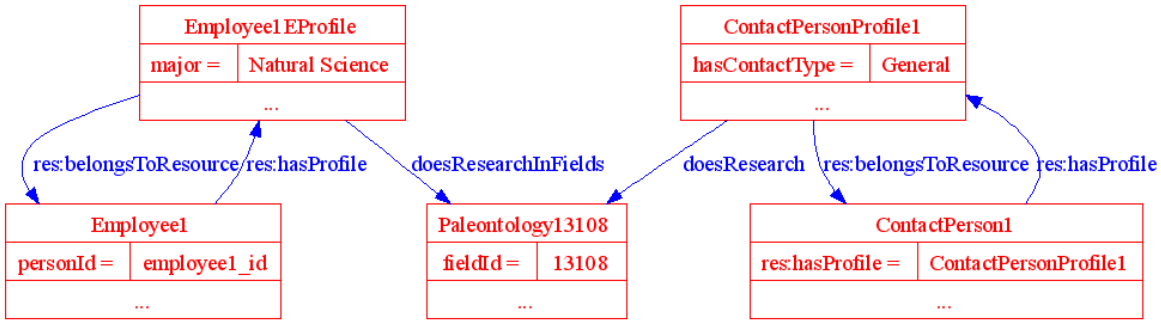

Fig. 3. Employee and foreign contact

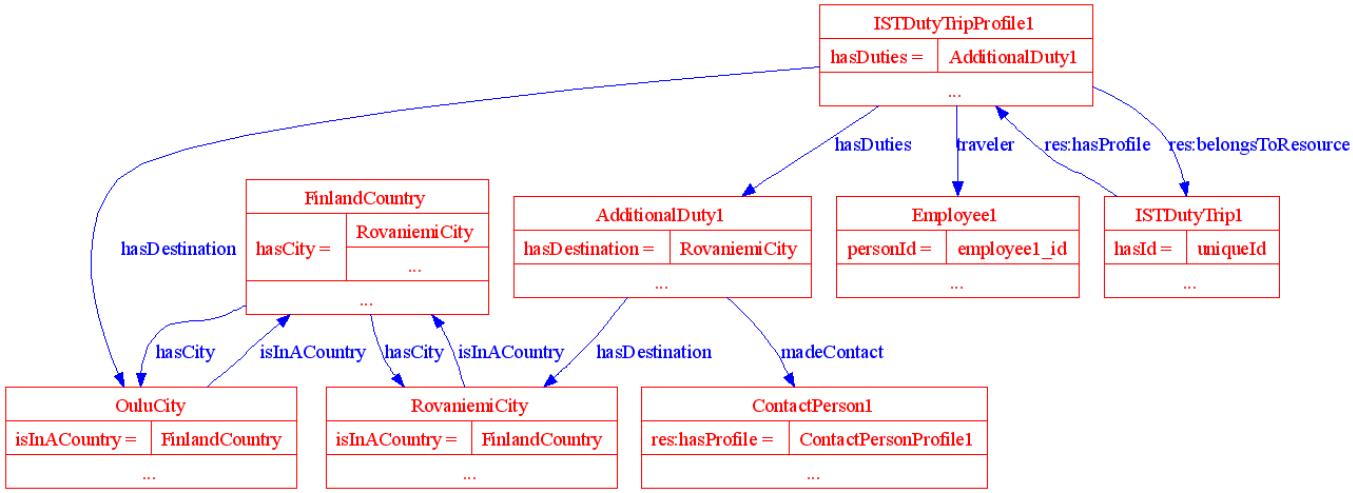

Fig. 4. Duty Trip related objects

V. S

In [7] we have described in some details two main building blocks of the system: the *Relevance Calculation Engine* and the *Relevance Calculation Interface*. Therefore, here, we discuss only the *GIS subsystem*.

A. The GIS Subsystem

In [6], [5], [7] we have outlined utilization of the GIS module—it is queried in case objects which have geospatial location properties that are involved in the matching operation. The state of the art research has shown that we can provide a reliable geospatial backend for our system by using the following components: (1) the GeoMaker [21] for collecting geographic coordinates of cities in the world, (2) the PostgreSQL database [22] for storing that information and for caching the result, and (3) Java *GIS—coordinates and distance cache* for calculating distance between cities, populating distance calculation results cached in the PostgreSQL database and interfacing the GIS module with the rest of the system. In Figure 5, which represents the GIS subsystem, these elements are placed on its bottom.

In the system that is supposed to communicate with the *GIS component*, the *GIS data consumer* is an interface that is responsible for requesting new data from the *GIS component*. On the system side, data is going to be stored in the semantic data storage that is based on the *JENA Model* [23]. We assume that the semantic data storage and *GIS component* share city instance identifiers in order to communicate in an optimal way in terms of performance.

For the purpose of calculating distance between two cities we utilize an implementation of the *Great Circle Distance Formula* [24]. This formula uses spherical trigonometry functions. Although relatively high precision of this method is not required in the system for the purpose of calculating distance between cities we apply it because the distances are calculated only once for each pair of cities.

$$result = 69.1 * \frac{180}{\pi} * \arccos\big(\sin LAT1 * \sin LAT2 + \\ + \cos LAT1 * \cos LAT2 * \cos(LONG2 - LONG1)\big)$$

As it was pointed in previous sections, the *GIS component* is still under development and changes in its design (performance-related changes in particular) may be introduced as a result of further experiments.

VI. C

In this paper we have presented a novel ontological matching algorithm in which we have combined matching based on

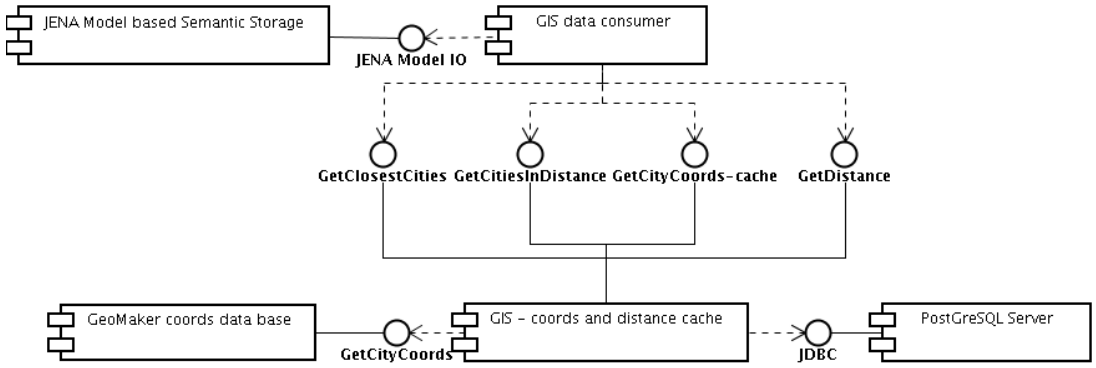

Fig. 5. The GIS subsystem; UML component diagram

ontological distance with filtering based on individual profiles. The proposed algorithm was illustrated in the context of a *Duty Trip Support* application. Across the paper we have identified a number of research questions, especially those related to an efficient implementation of geospatial data processing. We are currently implementing the proposed algorithm and the GIS subsystem as a part of the *DTS* application. Completing the initial implementation will allow us to start experimentally investigating all efficiency related questions. We will report on our progress in subsequent publications.

A

This work is partially sponsored by the KIST-SRI PAS "Agent Technology for Adaptive Information Provisioning" grant.

R

[1] http://en.wikipedia.org/wiki/Virtual_enterprise, 2008.
[2] http://www.businessdictionary.com/definition/virtual-organization.html, 2008.
[3] M. Ganzha, M. Gawinecki, M. Szymczak, G. Frackowiak, M. Paprzycki, M.-W. Park, Y.-S. Han, and Y. Sohn, "Generic framework for agent adaptability and utilization in a virtual organization—preliminary considerations," in *Proceedings of the 2008 WEBIST conference*, J. Cordeiro *et al.*, Eds. INSTICC Press, 2008, pp. IS–17–IS–25.
[4] M. Szymczak, G. Frackowiak, M. Gawinecki, M. Ganzha, M. Paprzycki, M.-W. Park, Y.-S. Han, and Y. Sohn, "Adaptive information provisioning in an agent-based virtual organization—ontologies in the system," in *Proceedings of the AMSTA-KES Conference*, ser. LNAI, N. Nguyen, Ed., vol. 4953. Heidelberg, Germany: Springer, 2008, pp. 271–280.
[5] G. Frackowiak, M. Ganzha, M. Gawinecki, M. Paprzycki, M. Szymczak, M.-W. Park, and Y.-S. Han, *Considering Resource Management in Agent-Based Virtual Organization*, ser. Studies in Computational Intelligence. Heidelberg, Germany: Springer, 2008, in press.
[6] ——, "On resource profiling and matching in an agent-based virtual organization," in *Proceedings of the ICAISC'2008 conference*, ser. LNCS. Springer, 2008.
[7] M. Szymczak, G. Frackowiak, M. Ganzha, M. Paprzycki, M.-W. Park, Y.-S. Han, Y. T. Sohn, J. Lee, and J. K. Kim, "Infrastructure for ontological resource matching in a virtual organization," in *Proceedings of the IDC Conference*, ser. Studies in Computational Intelligence, N. Nguyen and R. Katarzyniak, Eds., vol. 134. Heidelberg, Germany: Springer, 2008, pp. 111–120.
[8] M. Ganzha, M. Paprzycki, M. Gawinecki, M. Szymczak, G. Frackowiak, C. Badica, E. Popescu, and M.-W. Park, "Adaptive information provisioning in an agent-based virtual organization—preliminary considerations," in *Proceedings of the SYNASC Conference*, ser. LNAI, N. Nguyen, Ed., vol. 4953. Los Alamitos, CA: IEEE Press, 2007, pp. 235–241.
[9] M. Szymczak, G. Frąckowiak, M. Ganzha, M. Gawinecki, M. Paprzycki, and M.-W. Park, "Resource management in an agent-based virtual organization—introducing a task into the system," in *Proceedings of the MaSeB Workshop*. Los Alamitos, CA: IEEE CS Press, 2007, pp. 458–462.
[10] "Korea science and engineering foundation," http://www.kosef.re.kr/english_new/index.html.
[11] C. Badica, E. Popescu, G. Frackowiak, M. Ganzha, M. Paprzycki, M. Szymczak, and M.-W. Park, "On human resource adaptability in an agent-based virtual organization," in *New Challenges in Applied Intelligence Technologies*, ser. Studies in Computational Intelligence, R. K. N.T. Nguyen, Ed., vol. 134. Heidelberg, Germany: Springer, 2008, pp. 111–120.
[12] H. S. Pinto and J. P. Martins, "Ontology integration: How to perform the process," in *Proceedings do Workshop Ontologies and Information Sharing, realizado durante a conferência International Joint Conference in Artificial Intelligence, IJCAI2001*, Seattle, Washington, USA, 2001, pp. 71–80.
[13] A. Gangemi, D. M. Pisanelli, and G. Steve, "Ontology integration: Experiences with medical terminologies," in *Proceedings of Formal Ontology in Information Systems, FOIS'98*, N. Guarino, Ed. IOS Press, 1998, pp. 163–178.
[14] J. Euzenat and P. Shvaiko, *Ontology Matching*, ser. Studies in Computational Intelligence. Heidelberg, Germany: Springer, 2007.
[15] "Sparql query language for rdf," http://www.w3.org/TR/rdf-sparql-query.
[16] S. Rhee, J. Lee, and M.-W. Park, "Ontology-based semantic relevance measure," *CEUR-WS*, vol. 294, no. 1613–0073, 2007.
[17] ——, "Riki: A wiki-based knowledge sharing system for collaborative research projects," in *Proceedings of the APCHI 2008 Conference*, ser. LNCS. Springer, 2008.
[18] "Extensible sparql functions with embedded javascript," http://online-journals.org/proceedings/article/view/232/164.
[19] G. Frąckowiak, M. Ganzha, M. Gawinecki, M. Paprzycki, M. Szymczak, C. Bădică, Y.-S. Han, and M.-W. Park, "Adaptability in an agent-based virtual organization," *Internetional Journal Accounting, Auditing and Performance Evaluation*, 2008, in press.
[20] S. Rhee, J. Lee, and M.-W. Park, "Semantic relevance measure between resources based on a graph structure," in *Proceedings of the IMCSIT'08 Conference*, 2008, in press.
[21] "Geomaker," http://pcwin.com/Software_Development/GeoMaker/index.htm.
[22] "Postgis:home," http://postgis.refractions.net/.
[23] "Jena—a semantic framework for java," http://jena.sourceforge.net, 2008.
[24] http://www.meridianworlddata.com/Distance-calculation.asp, 2008.