Keywords: ontology, semantics, semantic web, agent systems, RDF.

Minor GORDON[a], Aleksander KOWALSKI[b], Marcin PAPRZYCKI[c], TOMASZ PEŁECH[d], Michał SZYMCZAK[e], Tomasz WĄSOWICZ[b]

ONTOLOGIES IN A TRAVEL SUPPORT SYSTEM

The development of next-generation Internet content services is predicated on the ability to process information automatically on a semantically-rich level. This requires design of semantic languages, domain ontologies written in semantic languages, information described with these ontologies, and agents for exploiting the information. This paper explores the latter two requirements. We describe the basic concept of ontology (as used in informatics), summarize research directions in this area. In the second part of the paper, we present and discuss our initial design of a travel-related ontology for a software agent system and illustrate its RDF-based implementation.

1. INTRODUCTION

Future of software agent development depends largely on availability of "intelligent" information for software agents to work with, yet without software agents and semantic-level processing "intelligent" information is very difficult to arrive at. Thus the course of agent and information development must be managed simultaneously. Toward this end we have started to investigate issues related to semantic representation of travel-related information [1, 2]. This paper constitutes an extension of this project into an attempt at designing and implementing our travel ontology, with focus and on the class of *hotels*.

---

[a] Technische Universitaet, Berlin, Germany
[b] Technical University of Warsaw, Warsaw, Poland
[c] Oklahoma State University, Tulsa, USA and SWPS, Warsaw, Poland
[d] University of Science and Technology, Krakow, Poland
[e] Adam Mickiewicz University, Poznań, Poland

We proceed as follows. In the next section we introduce the concept of ontology as it is used in the world of computing. We follow with description of existing upper level ontologies and travel-related ontologies (in Sections 3 and 4). Finally, in section 5, we sketch design of our travel ontology, provide a general example of its usefulness and briefly illustrate its RDF-based implementation.

2. CONCEPT OF ONTOLOGY

The term *ontology* originates from philosophy, where it is defined as the branch of philosophy that is devoted to answering two basic questions: (a) what exists? and (b) if what exists is divisible into parts, then what are these parts, and what are the relationships between them? In the world of computing answers to these questions become a formally represented knowledge based on conceptualization of objects, concepts, and other entities that are assumed to exist in some area of interest and the relationships that hold among them [3]. This conceptualization is an abstract, simplified view of the world that we wish to represent for some purpose. In this sense, every knowledge base, knowledge-based system, or knowledge-level agent is committed to some conceptualization, explicitly or implicitly [4]. Such a conceptualization, when explicitly defined, is called an ontology.

Ontologies are often equated with taxonomic hierarchies of classes, class definitions, and the subsumption relation. However, such hierarchies are only a subset of a much broader class of conceptual entities. Furthermore, ontologies are not limited to definitions in the traditional logic sense, i.e. only introducing terminology and not adding any knowledge about the world [5]. In an ontology, definitions are associated with names of entities in the universe of discourse (e.g., the set of objects and their relations that can be represented) along with formal axioms that constrain the interpretation and well-formed use of these terms and possibly human-readable text describing what the names mean.

An ontology defines the basis of a vocabulary with which queries and assertions are exchanged between parties (such as software agents) [6, 7]. Ontological commitments are agreements by there parties to use the shared vocabulary in a coherent and consistent manner. Agents sharing a vocabulary need not share a knowledge base; each agent may know facts other agents do not, and an agent that commits to ontology is not required to answer all queries that can be formulated in the shared vocabulary. In short, a commitment to a common ontology is a guarantee of consistency, but not completeness, with respect to queries and assertions using the vocabulary defined in the ontology.

Let us now direct our attention to the existing ontologies. While the area is brimming with research activities, some ontologies are currently better known then others. It should be stressed that there are two basic approaches to developing ontologies. One is to try to create a “top-down” ontology that is as robust enough to encapsulate an extremely large number of “notions.” The second approach is to design “bottom-up” domain-specific ontologies and link them to form a whole (this is the vision of the Semantic Web [8]). We will now illustrate both approaches, briefly describing well-known attempts at developing general ontologies and presenting some travel ontologies we were able to locate.

3. GENERAL UPPER-LEVEL ONTOLOGIES

The most well known attempts at creating an upper-level ontology defining the most general and most often used concepts are as follows:

- The most comprehensive ontology available today is **Cyc**, a proprietary system under development since 1985, which consists of foundation ontology and several domain-specific ontologies (called microtheories). A subset of Cyc (written in OWL) has been released for free under the name OpenCyc [9, 10].
- **WordNet** is a freely available database that was originally based on psycholinguistic principles and later expanded with definitions becoming a general dictionary for use in Natural Language Processing. WordNet includes most general concepts as well as more specialized concepts, related to each other not only by subsumption relations, but by other semantic relations such as part-of and cause. Unlike Cyc, WordNet has never been axiomatized to make logical relations between concepts precise [11].
- **Suggested Upper Merged Ontology** (**SUMO**) is another attempt at defining an upper level ontology. It was created by an IEEE working group (predominantly by a group at Teknowledge) and is freely available. SUMO is written in the SUO-KIF language and has been mapped to the WordNet lexicon. Furthermore, it includes language generation templates for Hindi, Chinese, Italian, German, Czech and English [12, 13].
- The **SENSUS** project is an extension of the **WordNet** and is focused on creating a semantic thesaurus required for deeper (semantic) understanding of texts, to be applied, among others, in Machine Translation, Summarization, and Information Retrieval. Currently SENSUS consists of 70,000-node taxonomy and a framework into which additional knowledge can be added [14].

4. DOMAIN SPECIFIC – TRAVEL ONTOLOGIES

The “bottom-up” approach to developing ontologies focuses on describing the objects and relationships in a specific domain in detail. This approach is usually driven by the need for a workable vocabulary (e.g. to facilitate communication between software agents) rather than a comprehensive view of the universe of discourse. Furthermore, ontologies designed this way are typically built with an application in mind and in this way it is the application that “drives” development of ontology. It should be noted, that these requirements are more manageable and ultimately more pragmatic.

Here we concentrate on ontologies for the travel domain, for which we ourselves are developing an ontology (described in Section 5) and a software agent system that works with that ontology [15].

4.1. OTA SPECIFICATION

The Open Travel Alliance (OTA) specifications have been designed to serve two purposes: (a) as a common language for travel-related terminology and (b) a mechanism for exchange of information between travel industry members. The OTA specification is an attempt to create, from the business perspective, a possibly complete ontology for the “world of travel”. While the word *ontology* itself is not used in the description of the project [16], it is possible to view the OTA specifications as a comprehensive ontology, defining concepts such as AirSchedule, GolfCourseReservation, HotelContentDescription, HotelPreferences, etc. The OTA specification has already been utilized in some travel-related AgentCities projects.

4.2. MONDECA

Mondeca´s [17] tourism ontology defines tourism concepts based on the WTO thesaurus [18]. These include, among others, terms for tourism object profiling, tourism and cultural objects (place, museum, restaurant, housing, transportation, events...), tourism packages and tourism multimedia content. Mondeca created a proprietary system ITM that is used to manage its travel ontology.

4.3. TAGA ONTOLOGY

The Travel Agent Game in Agentcities (TAGA) is an agent framework for simulating the global travel market on the Web. Its purpose is to demonstrate Agentcities [19] and Semantic Web technologies. TAGA works on FIPA-compliant platforms within the Agentcities Environment [20]. In addition to the FIPA content language ontology, TAGA defines two domain ontologies to be used in simulations. The first TAGA ontology covers basic travel concepts such as itineraries, customers, travel services, and service reservations. The second ontology is devoted to auctions and defines different types of auctions, roles the participants play in them, and the protocols used etc. Unfortunately, TAGA ontologies are limited by their usability (only defining very broad concepts, in not much detail) and fairly unrealistic due to the nature of TAGA simulations.

4.4 HARMONIZE ONTOLOGY

Harmonize is an attempt at ontology-mediated integration of tourism systems following different standards [21, 22]. Its goal is to allow organizations to exchange information without changing data structures. The Harmonize project also involves sub-domains that are only partially related to the world of travel: geographical and geo-spatial concepts, means of transportation, political, temporal, activity/interest, gastronomy etc. These sub-domain concepts can be used within the travel system (as needed) or incorporated into the ontology constructed for the system. A comprehensive guide was prepared within the E-Tourism project. Here, it is claimed that next generation of "eTourism" will be powered by the Semantic Web technology (resulting in an eTourism Semantic Web portal which will connect the customers and virtual travel agents from anywhere at anytime).

4.5. DAML-BASED ONTOLOGIES

A number of "minimalist" travel ontologies can be found within the DAML language portal [23]. For instance, the Itinerary-ont is a simple ontology for representing travel itineraries [24]. It was actually defined as an itinerary-ont.n3 and translated to DAML (and later OWL) using CWM. It reuses the airport codes ontology and involves definitions of only the most basic terms like Aircraft, Class, Flight etc. Another example is the Trip Report Ontology [25] that defines terms like Airfare, Amount, Date, etc.

4.6. OAS WORKSHOPS AND THEIR OUTCOME

Finally, one of the important academic workshops on ontologies and agents is the series of Ontologies in Agent Systems (OAS) Workshops organized within the framework of the Autonomous Agents and Multiagent Systems (AAMAS) conferences [26]. The OAS workshops are devoted to: (1) practical experience and considerations in designing applications where interactions are based on ontologies, and the infrastructural support required for their effective use, (2) discussion of the dependencies between ontologies, their supporting technologies and other aspects of agent systems such as agent architectures, and interaction mechanisms (coordination, communication, etc.), (3) comparison of different ontology representation approaches for use in agent systems. The most interesting aspect of the OAS workshops is the challenge problem of designing and (preferably) implementing multi-agent system in the domain of world of travel. Note that the focus of the challenge was not the ontology itself, but rather how it is used during the process of design, implementation and execution of agent-based system [27].

Despite proliferation of travel-related ontologies, we have yet to find a clean and complete ontology of basic entities that exist in the *world of travel* such as, for instance, a *hotel* or a *pub*, or a *movie theater*. In response to this situation, in the second part of our paper we present our initial attempt at defining domain-specific travel ontology, with focus on the definition of the class of *hotels*.

5. TRAVEL ONTOLOGY

Before starting the ontology development we need to speify requirements for our project. Following list presents those we set up at the very beginning (here we follow suggestions presented in [28, 29, 30]):

- Understandability to a person who wishes either to describe documents with the ontology or to develop software for mining web with its use.
- Bringing computational benefits with describing, still in bulk, not semantically described travel-related segment of Internet stored documents.
- Consistency, accuracy and completeness which result in easiness and effectiveness of working with the ontology to non-human as well as human users.

In case of our project, the proposed ontology will be the centerpiece around which an agent based travel support system will be build. The proposed system will provide prospective users with a range of services related to broadly understood "needs of a traveler" (some of the examples of available services have been illustrated in section 5.3). To achieve this goal we will use our travel ontology to describe/demarcate and manage data that's already available on the Web, but is not semantically described.

5.1. ONTOLOGY DESCRIPTION

Ontology of travel/tourism can be divided into two parts: one that describes concrete, real-world objects and "relationships between them", and one that defines more intangible business aspects of the world of travel. Following this delineation, the most general view of the proposed ontology is depicted in Figure 1.

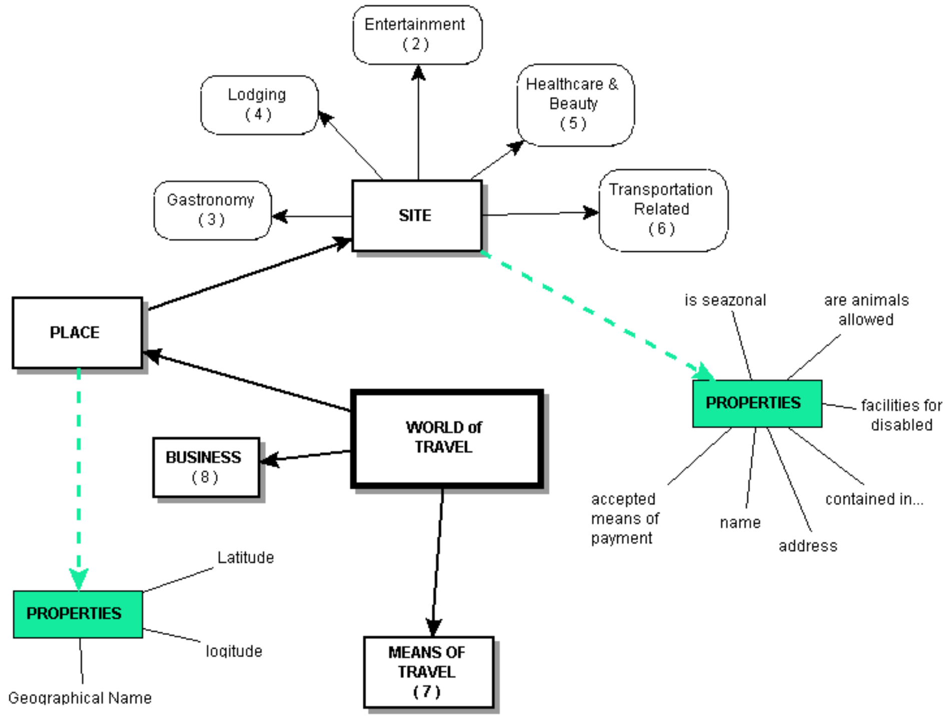

Fig. 1: Travel ontology

5.1.1. REAL-WORLD PART

The first and most general notion in this part of the ontology is the class *Place* with its properties (which include, among others, a complete set of geospatial characteristics necessary to support the GIS component of the proposed system). This class is as a very general notion covering geographical locations we can find on the map, such as mountains, valleys, cities etc. The *Site* class inherits *Place*'s properties and has its own properties thanks to which it describes locations more specifically. Figure 2 illustrates details brought by subclasses of the *Gastronomy* node.

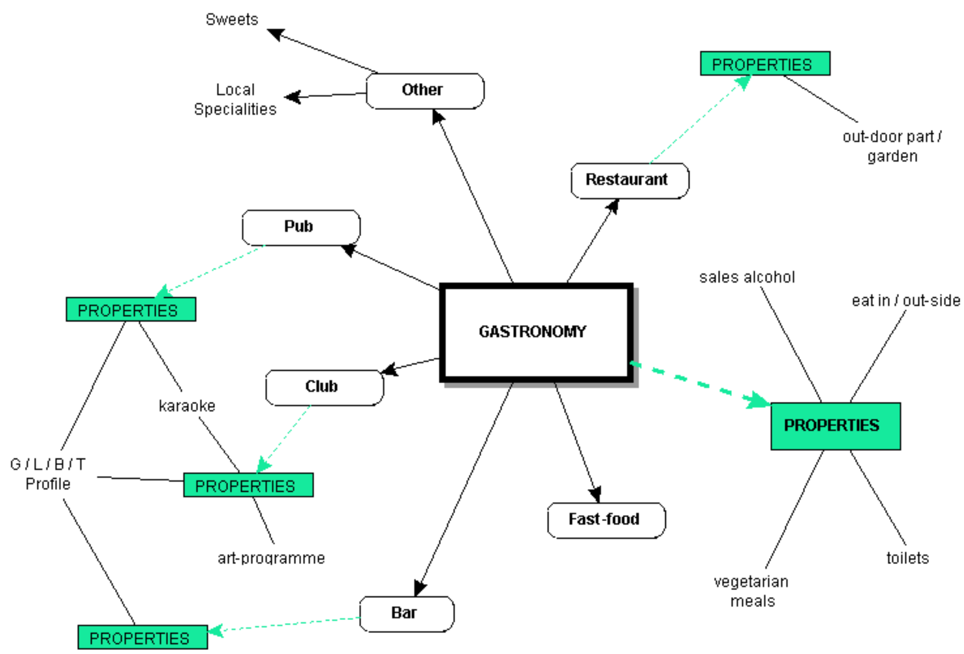

Fig. 2: *Gastronomy* with its subclasses and properties.

This part of the ontology handles also means of travel. We have distinguished three of them: *LandTransportation, AirTransportation* and *SeaTransportation*. Figure 3 depicts our approach to means of transportation.

5.1.2. BUSINESS PART

For the business-related part of the ontology, we have two basic entities: a *Customer* and a *ServiceProvider*. While the *Customer* does not require further explanation, the *ServiceProvider* is any kind of business that provides widely understood *Services* (in the context of travel). Any *ServiceProvider* contains a collection of *Services* it provides. The ontology includes both generic *Service* description objects and more specific classes of services for various types of *Sites*. By an *EntertainmentService* we mean a single entry to some event, like a movie screening, while by a *GastronomyService* we mean both a dish, i.e. a *MenuItem*, and a *Table* (needed for *Reservation* purposes).

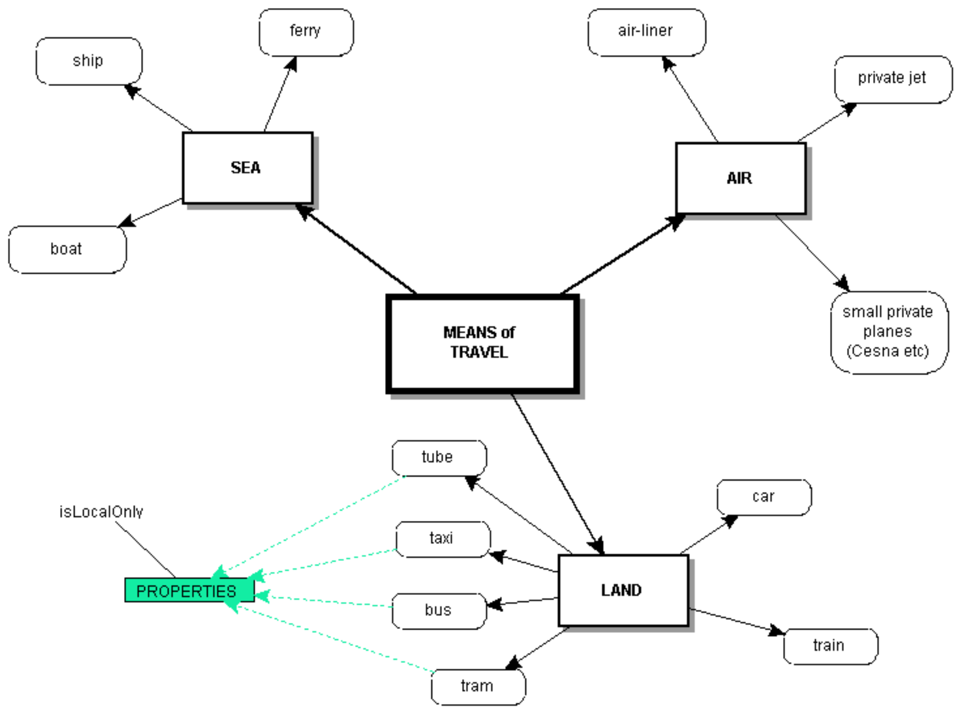

Fig. 3: *Transportation* with its subclasses and properties.

We can attach *ServiceProviders* to various real-world *Sites*. Namely, an instance of a *Cinema,* which is a type of *Site*, is at the same time an instance of the *ServiceProvider* class, therefore it has its bag of *Services* which reflect current repertoire. We found it useful to define another entity called *Ticket*, which reflects additional, customer dependent, properties of the *Service* of interest. For example, the price of an *EntertainmentTicket* will depend on the screening time and the day of the week as well as on whether the *Customer* is entitled to a discount. A *ConnectionTicket* concept comes in handy when we think of a *Connection* (*TransportationService*) from city A to city B and goes through *itineraryPlaces* C and D, but the *Customer* wants to travel only from A to C – the *ConnectionTicket* will carry that information, as well as optional data concerning the seat, *comfortClass* etc.

It is also useful to define a notion of *Reservation*, which points to a given *Service* or a proper type of a *Ticket* in case of *Entertainment* and *Transportation Services*. *Reservation* has some common properties – it carries information about the *timeOfTheEvent, expiryDate, ServiceProvider,* the *Owner* (who is a *Customer*), and the cost of the

Reservation itself (as opposed to the *price* of the given *Service*). There are also some type-specific properties, e.g. *TransportReservation* would point to a *ConnectionTicket*, while a *GastronomyReservation* would carry the information about the *numberOfPeople* we reserve a table for and whether the table is *non-Smoking* or not. For illustration of some of just described considerations, see Figure 4.

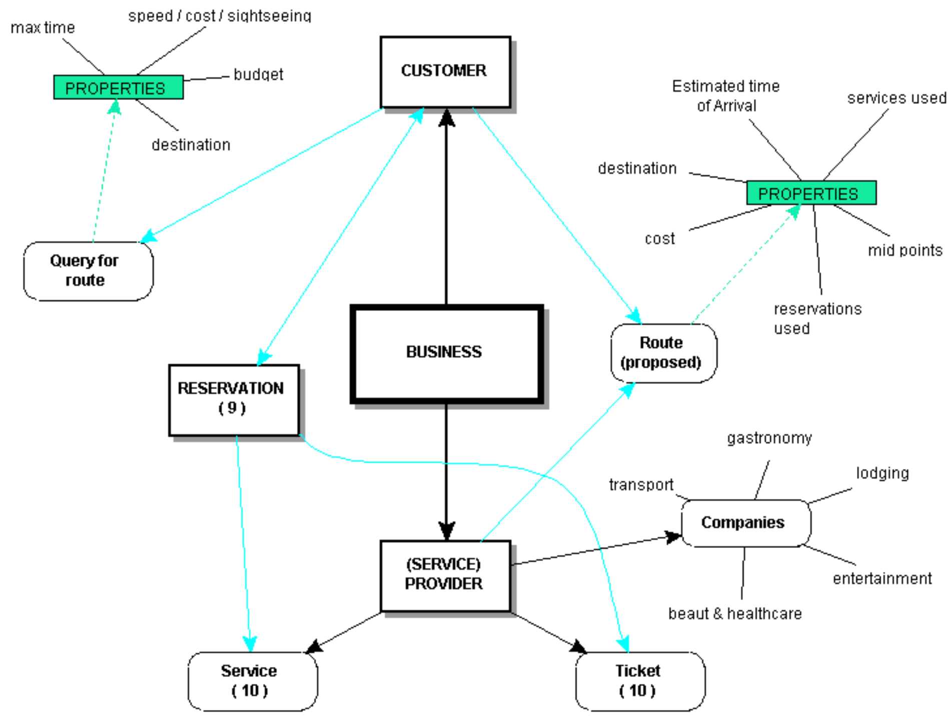

Fig. 4: Business part of the travel ontology

5.1.3. HOTEL ONTOLOGY

The hotel concept is a specific branch of our ontology that illustrates the use of the abstractions described above. Since our ontology will be used primarily to collect and manage information available on the Internet it is this very information that has to drive the development of out *hotel* ontology. We cannot start from theorizing and lexicon-based definitions as the resulting ontology may not match the information that is actually available on the Web. Therefore, in our attempt at designing *hotel* ontology, we started by analyzing the better known travel-related web-sites:

- http://www.travelocity.com

- http://www.venere.com
- http://www.hotelclub.net
- http://www.hrs.de
- http://www.web-hotels.com
- http://www.travelweb.com
- http://www.travelciti.com

which constitute the top-10 of Google's response to the *hotel reservation* query. We looked at description of hotels as presented by these sites. In order to make our study a bit more focused, we have looked each time at the same hotel (Warsaw Le Royal Meridien Bristol) to see how it is represented/described. We have found that – disregarding the differences in layout – all services carry practically the same information about hotels:

Table 1: Found information about hotels

Element of description	**Did we include it**
Name	Yes
Address	Yes
Rating (number of stars)	No
Total number of rooms	Yes
Room types with their prices	Yes
Amenities	Yes
Way of reaching the place	Yes
Dining	Yes
Nearby attractions	No
Transportation	Yes
Accepted means of payment	Yes
Accepted currencies	No
Check in/out time	Yes
Booking and cancellation policy	No
Guaranteed rates & other	No
Pets accepted	Yes
Photos of the property	No
City map with hotel location	No

Let us make a few comments about the characteristics that we have identified and used:

(1) omitted characteristics can be easily added later, e.g. nearby attractions may be just an additional set of Site class elements etc.,

(2) since the rating system is not uniform among the sites, some of them use their own ratings while others use the 'international star system' or information provided by hotels themselves we decided, at least for the time being to skip this category,

(3) as far as the information about rooms and pricing was concerned, there also was no consistency; in most places a simple info was provided, e.g. deluxe room

with king bed; in some other cases only an overall written comment about standards was provided; furthermore, in some places, a distinction between hotel's amenities and room amenities was made, while in other cases it was not; finally, in some cases not all types of room were listed and we're not quite sure if this is a result of the fact that only a part of them could be booked online or some other reasons (specific contracts between the hotel and the website, for instance).

In Figure 5 we depict the general structure of the *hotel ontology*.

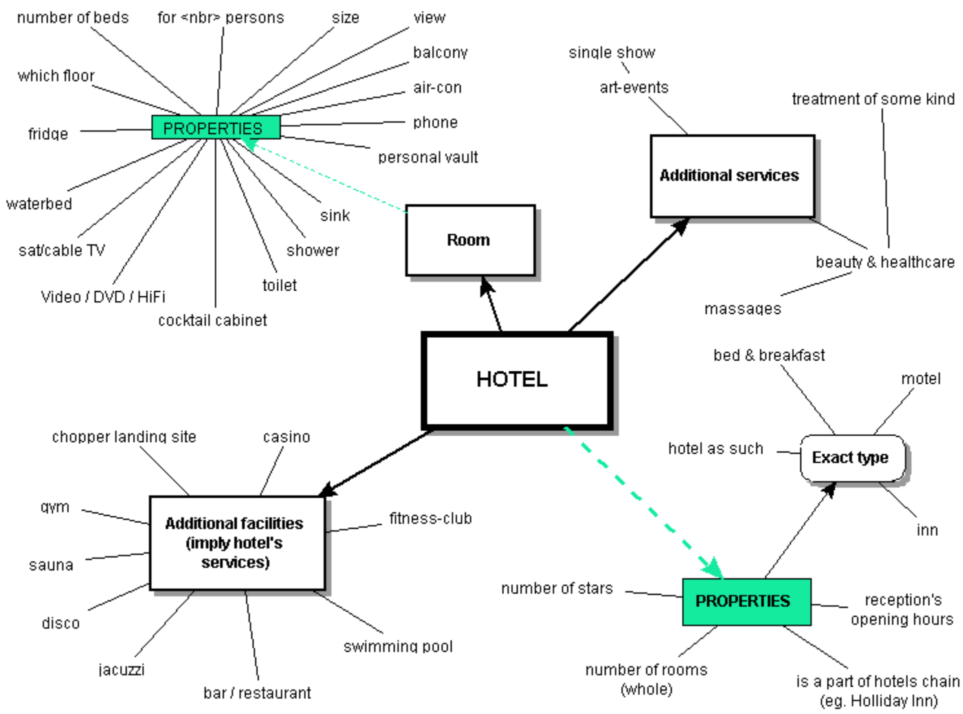

Fig. 5. *HotelAlike* with subclasess and properties

5.2. EXAMPLE OF APPLICATION

Let us assume we have a *customer* who wants to (*query for route*) get from Warsaw (*Place*) to Krakow (*Place; query for route-> destination*) as fast as possible (*query for route->optimize for speed*) having 600zl for the trip (*query for route->budget*). It is also crucial that he reaches the place within two days (*query for route->max time*). Such a *query for route* is sent to the processing system which begins with checking the *Place* called Krakow for *Transportation related Sites*. Having found a list of those he creates a list of *Service Providers* of type *Transportation* whose *Transportation Services* are listed as *Lines/Firms hosted* by those *Transportation related Sites*. Which is important is the fact that our ontology takes into account that we might need to use only a short part (*Itinerary Places*) of the *Transportation Service* which, by default, may connect two more distant *Places*. Now we can choose among *Means of Travel* to choose the one which suits the best the assumptions of *Query for route*. Let us say that in *Place* Krakow we have 3 *Transport related Places*: a *Bus Station*, an *Airport* and a *Railway Station*. This means we can use *Land Means of Transport* such as a *Car* a *Train* and a *Bus* as well as *Air Means of Transport* such as a *Private Jet* or an *Airliner* (of course only if the *Airport* has those kinds of planes listed on its *Supported Sizes* list). The agents systems now can connect to the *Service Providers* of type *Transportation* to get some *Reservations* for specific *Connections*. A *Reservation* features its own *Cost* (in case if there's an additional fee for booking a place), a *Provider* (of type *Service Provider*) whose *Service* we will use, the *Owner* (of type *Client*) of the reservation, the *Expiry date*, *Time of the Event* (in this case of the journey) and number of *Services* of given kind (using the assumption that buying two tickets in the same train we book twice the same *Service*). In the notion of *Transportation Service* there is also a notion of *Place In Transportation* which details the *Connection Involved*, the *Comfort Class* and the *Place Number* so we have a complete description of the transportation we will use. Nevertheless we also considered the possibility that we will not be able to book (*Reservation*) some kind of a *Service* (for example Polish PKS does not provide such possibility and we buy tickets on place hoping for some free places) so our *Proposed Route* which is one of possible answers to the *Query for Route* consists of both the *Reservations* for *Services* (*Reservations Used*) and the *Services* as such (*Services Used*). The *Route Proposed* also features the *Cost* of such a trip (being the sum of *Prices* of *Services* and *Cost* of the *Reservations*), *Estimated time of Arrival* (to see if we fit within the *Query for Route->max time*) the *Destination* (of type *Place*) and *MidPoints* (also of type *Place*). The reason for separate *MidPoints* (separate in the meaning that it doubles the *Itinerary Places*) is the fact that the *Transportation Service* lists only *Sites* of type *Transportation Related Places* (such as *Railway Stations*) but not geographical *Places* such as mountains, valleys or other more general notions. We thought this might be interesting for a *Customer* not only to know the *Railway Stations* but also the fact that on his way he will cross for example the Rocky Mountains.

The *Customer* most probably will also be interested in getting a *Reservation* for a *Lodging Service* to book a *Hotel Room* in a *Hotel alike place* of some standard (*Hotel alike place->number of stars*; properties of the *Hotel Room*). But before he gets there, the *Customer* can be provided with info on *Connection to the main City* if the *Airport* or the *Railway Station* is somewhere in the suburbs. This way he can learn which *Means of*

Local Transportation he may use. Once in his *Hotel Room* the *Customer* may wish to do some sightseeing so we may list *Entertainment Sites* in the *Place* called Krakow and have a new *Query for route* with a *Destination* being a *Museum* (of type *Site*) and *Profile for sightseeing* on the way which will use *Means of Local Transportation* such as a *Tram*, a *Bus* or a *Tube*. Before leaving his *Hotel Room* the *Customer* can already have his *Tickets* booked for *Event* of some *Name* as well as a reservation for a *Table* for *4 persons* in the *non-smoking* area for *time of the event* set to a few minutes more than the *Estimated timeofArrival* of *Route proposed* beginning in the *Museum* and a *Destination* set to a *Restaurant* with the *Name* and knowing in advance what are the *Menu Items* in there and whether he will find *Vegetarian* meals (and even what are those *Dishes Names* – the *Restaurant* as the *Service Provider* may expose whole menu as the list of *GastronomyService*s).

Later the *Customer* will have yet another *Query for route*, this time heading (*Destination*) his *Hotelalike Place Optimising* it *for speed* probably using a *Taxi*. Please note that this example shows only one of the possible uses of the ontology as it was developed in such a way that it does not impose any specific structure of the travel agent system. This example is pretty straightforward and used only a small number of notions available in our ontology. Nevertheless we hope that this presentation shows that our ontology is very flexible and universal and yet at this stage of development it creates easy ways of describing vast majority of situations from the world of travel.

5.3. ONTOLOGY IMPLEMENTATION

We have selected the RDF language to encode/implement our ontology and therefore each "statement" consists of a triple (subject, predicate, and object) [31]. Moreover, RDF allows easy usage of other existing schemas and ontologies. In our project we have used the RDF Schema in order to describe the vocabulary. Let us now proceed to briefly illustrate selected features of implementation of our travel ontology through snippets of the RDF code. Starting from the top, the "header" of our ontology has the following form:

```
<!DOCTYPE rdf:RDF [<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">]>
<rdf:RDF
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
 xml:base="http://travelSystemServer.com/schemas/travel_ontology">
```

We have then proceeded to define useful data types:

```
<rdfs:Datatype rdf:about="&xsd;integer"/>
<rdfs:Datatype rdf:about="&xsd;string"/>
```

```
<rdfs:Datatype rdf:about="&xsd;boolean"/>
<rdfs:Datatype rdf:about="&xsd;float"/>
```

Furthermore, the following lines of RDF define one of the node-classes and its properties; class *Place* is the root class for the Real-World part in the presented ontology:

```
<rdfs:Class rdf:ID="Place"/>
   <rdf:Property rdf:ID="geoName">
      <rdfs:domain rdf:resource="#Place"/>
      <rdfs:range rdf:resource="&xsd;string"/>
   </rdf:Property>
```

Domain provides information "what kind" or, rather, "where from" a document described by the current property originates, while the range is the expected value of the property. The next RDF snippet illustrates a class hierarchy and of inheriting properties:

```
<rdfs:Class rdf:ID="Site">
   <rdfs:subClassOf rdf:resource="#Place"/>
</rdfs:Class>
   <rdf:Property rdf:ID="isSeasonal">
      <rdfs:domain rdf:resource="#Site"/>
      <rdfs:range rdf:resource="&xsd;boolean"/>
   </rdf:Property>
```

Here, the class *Site* is defined as a subclass of *Place* and this enables sharing of properties. In this way, the class *Site* will have both: *geoName* and *isSeasonal* properties.

The most important class in our ontology is the *HotelAlike* (as presented in Figure 5). This class is an indirect subclass of the class *Place*. To describe hotel characteristics such as amenities, types of available rooms and "things" which guests can find in them, we used the idea of recursion as illustrated in the following code:

```
<rdf:Bag rdf:ID="additionalHotelFacilities">
   <rdfs:domain rdf:resource="#HotelAlike"/>
   <rdfs:range rdf:resource="#Site"/>
</rdf:Bag>
```

This RDF code tells us that the document describing the hotel may have some *additionalHotelFacilities* which will belong to the class *Site* and may have all properties of this class. Additionally *rdf:Bag* means that we can expect a collection of *additionalHotelFacilities* represented as the *bag* type available in RDF. In a similar way we have represented hotel room types:

```
<rdf:Bag rdf:ID="typesOfRoomsAvailable">
```

```
   <rdfs:domain rdf:resource="#HotelAlike"/>
   <rdfs:range rdf:resource="#HotelRoom"/>
</rdf:Bag>
```

7. CONCLUDING REMARKS

We are in the process of developing and implementing the proposed ontology. Currently, the ontology of the hotel is almost complete. Completing it and combining with the *restaurant* ontology provided by the ChefMOZ [32] is our next order of business. We expect that this step may result in further tuning of the *hotel* ontology. Other sources of possible changes will be, among others: (a) the data collection subsystem, (b) the personalized content management subsystem, (c) the GIS subsystem that are currently being co-developed and (d) other existing ontologies. In the latter case, when defining the *hotel* we have used *HotelRoom* concept, derived from a different ontology. Overall, we expect that the subsystems under development as well as already existing travel ontologies will cross-influence each-other and the travel ontology throughout an iterative refinement process. In the near future we plan to devote our attention to completion of the *Entertainment, EntertainmentService* and appropriate *Ticket* and *Reservation* ontologies. We will report on our progress in subsequent papers.

REFERENCES

[1] GILBERT A., GORDON M. , PAPRZYCKI M., WRIGHT J., *The World of Travel: a Comparative Analysis of Classification Methods*", Annales UMCS Informatica, A1, 2003, 259-270

[2] GILBERT A., NAULI A., PAPRZYCKI M., GORDON M., WILLIAMS S., J. Wright, *Indexing Agent for Data Gathering in an e-Travel System*, Informatica, Vol. 28, No. 1, 2004, 69-78

[3] GENESERETH, M.R., NILSSON N., *Logical Foundation of Artificial Intelligence*. Morgan Kaufmann, Palo Alto, CA, 1986

[4] http://www-ksl.stanford.edu/kst/what-is-an-ontology.html

[5] ENDERTON, H.B., *A mathematical introduction to logic*, Academic Press, N. Y., NY, 1972

[6] LEVESQUE, H. J., *A logic of implicit and explicit belief. In Proceedings of the Fourth National Conference on Artificial Intelligence*, Austin, TX, 198-202

[7] LEVESQUE, H. J., COHEN, P. R., NUNES, J. H. T. (1990). *On acting together*. In Proceedings of the Eighth National Conference on Artificial Intelligence, Boston, MA, 94-99

[8] FENSEL D., *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*, Springer, Berlin, Germany, 2001

[9] http://www.opencyc.org

[10] http://www.daml.org/ontologies/269
[11] http://www.daml.org/ontologies/196
[12] http://www.daml.org/ontologies/172
[13] http://www.ontologyportal.org
[14] http://www.isi.edu/natural-language/projects/ONTOLOGIES.html
[15] ANGRYK R., GALANT V., GORDON M., PAPRZYCKI M., *Travel Support System – an Agent Based Framework* in: H. R. Arabnia, Y. Mun (ed.), Proceedings of the International Conference on Internet Computing (*IC'02*), CSREA Press, Las Vegas, NV, 2002, 719-725
[16] http://www.opentravel.org
[17] http://www.mondeca.com
[18] http://www.world-tourism.org
[19] http://www.agentcities.org
[20] http://www.fipa.org
[21] http://deri.at/research/projects/e-tourism
[22] http://www.deri.at/research/projects/e-tourism/2004/d10/v0.1/20040719
[23] http://www.daml.org
[24] http://www.daml.org/ontologies/178
[25] http://www.daml.org/ontologies/365
[26] http://oas.otago.ac.nz/OAS2003/
[27] http://oas.otago.ac.nz/OAS2003/papers/oas03-dickinson.pdf
[28] http://arti.vub.ac.be/kads/CH/subsubsection3.2.1.2.html
[29] http://arti.vub.ac.be/kads/CH/subsubsection3.4.1.2.html
[30] ESHELMAN, L.. *MOLE: A knowledge-acquisition tool for cover-and-differentiate systems*. In Marcus, S. (Ed.)., Automating Knowledge Acquisition for Expert Systems. Kluwer Academic Publishers, The Netherlands, 37-80
[31] http://www.w3.org/RDF
[32] http://www.chefdmoz.com/

SUMMARY

Spełnienie oczekiwań dotyczących Internetu jako wartościowego źródła informacji stwarza potrzebę znalezienia narzędzi umożliwiających semantycznie-zorientowane przetwarzanie danych. Celem artykułu jest, po pierwsze, wprowadzanie pojęcia ontologii w rozumieniu informatycznym, oraz przedstawienie najważniejszych projektów badawczych mających na celu stworzenie globalnych jak i specyficznych ontologii (w naszym przypadku ontologii podróży). Po drugie, zaprezentowany zostanie zarys proponowanej przez nas ontologii podroży. Propozycja zilustrowana zostanie przykładem zastosowania, oraz wybranymi detalami implementacji w języku RDF.