

Utilizing Ontologies in an Agent-based Airline Ticket Auctioning System

Mladenka Vukmirovic

Montenegro Airlines, Podgorica, Montenegro

mladenka.vukmirovic@mgx.cg.yu

Michal Szymczak

Adam Mickiewicz University, Poznan, Poland

Maria Ganzha,

Elblag University of Humanities and Economics, Elblag, Poland

Marcin Paprzycki

SWPS, Warsaw, Poland

Abstract. *Internal data representation is one of interesting issues in designing an agent based airline ticket auctioning system. Obviously, such system could utilize directly the airline mandated data format. However, if purchasing airline tickets is a part of a travel support system that utilizes ontologically demarcated content, then also air travel data should be represented in that way. Here, we outline our attempt at building a semantic bridge between actual airline data and ontology-based travel support system.*

Keywords. software agents, air-travel ontology, travel support system, e-commerce, e-auctions.

1. Introduction

Recently we have developed an agent based e-commerce system (see [2] and references to our earlier work). In its initial design, the proposed system was to mimic an e-marketplace, where multiple *Buyer Agents*, acting on behalf of their owner negotiate prices with *Seller Agents* representing merchants. After selecting the shop to make a purchase from, *Buyer Agents* autonomously complete the transaction. In the system we have used an airline ticket purchasing model, where successful completion of price negotiations results in a limited time reservation. During that time *Buyer Agents* have to confirm purchase or their reservation expires. More recently we explored this metaphor further and asked what would happen if our “theoretical” system had to interact with an actual airline reservation system [20]. In the augmented system, we have created a *FlightOffer Agent* that was to interface with a *Global Distribution System (GDS)*, e.g. AMADEUS or SABRE. Furthermore, when an actual purchase of a ticket was to take place, the *Reservation Agent* – was to interact with the *GDS* to complete transaction.

While conceptualizing the extended system, we have simply assumed that all interactions between system agents and the *GDS* were to take place and all necessary information was to be somehow exchanged. However, this is easier said than done. Here, we need to consider the way that the travel information is going to be represented in our system. If an airline ticket selling system is considered separately, then the internal data representation should match that used by the airlines (obviously, it is impossible to change the way information is stored and processed in a global system like Worldspan). However, situation changes when the system under consideration is to provide travelers with a complete support of their travel needs including, among others, restaurant information, ZOO schedule, historical points of interest, local weather etc. In such a case a unified way of representing and processing information within the system needs to be proposed and a way of interfacing with the *GDS* has to be specified.

In a separate project, we have developed and started implementing an agent based travel support system [8], in which travel-related data is represented as RDF demarcated instances of a travel ontology and used to deliver personalized information to users [5, 6, 7]. Let us now assume that airline ticket purchasing is to be integrated with such a travel support system. Obviously, in this case we need an air travel ontology that is to be fully integrated with the travel ontology used in the system. Such air travel ontology should also easily interface with the *GDSs* data.

The aim of this paper is to outline our initial attempt at designing air travel ontology. In the next section we briefly describe our airline ticket auctioning system. We follow with a summary of existing general and travel-related ontologies. In section 4 we present initial design of our proposed ontology of air travel.

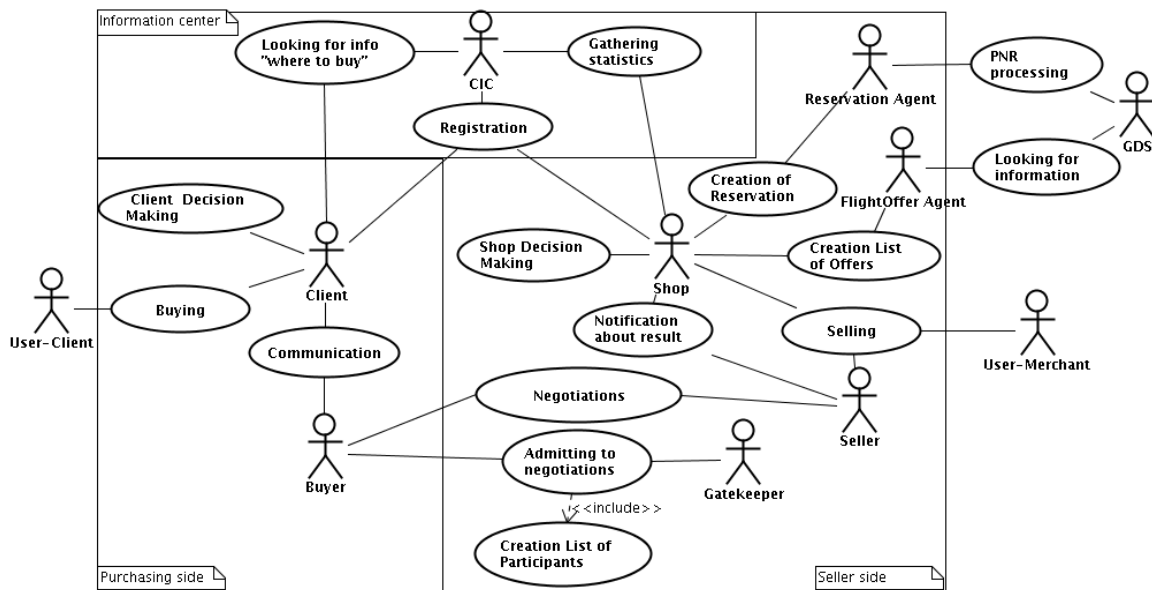


Figure 1. Airline ticket auctioning system – Use Case diagram.

2. Air-ticket auctioning system

Let us present design of the system through its UML Use Case diagram in Figure 1 (detailed descriptions of the system can be found in [20]). We can distinguish three major parts of the system. (1) The *Information center* area where white-page and yellow-page data is stored and serviced by the *CIC Agent*. Currently, *User-Merchants* sell tickets only for specific routes that they believe to be profitable. Each such route is advertised through the *CIC*. Every time the *Client Agent* is searching for an airline ticket for its *User-Client* it communicates with the *CIC* to find out which e-shops sell it. (2) The *Purchasing side* where agents and activities representing *User-Client* are represented. Here the *User-Client* informs the *Client Agent* which tickets it would like to purchase. The *Client Agent* should be viewed as an incarnation of a *Personal Agent* [11] that knows preferences of its *User-Client* and autonomously acts on her behalf. *Client Agent* obtains from the *CIC* information which e-shops sell requested tickets and sends a *Buyer Agent* to each one of them. *Buyer Agents* engage in price negotiations with *Seller Agents*. Successful price negotiations results in a reservation. *Client Agent* decides which store to make a purchase from and, if the reservation did not expire and the tickets are still available in the *GDS*, they are purchased. (3) The *Seller side* involves *Shop Agent* acting on behalf of its *User-Merchant* and attempting at selling air tickets for routes defined by her. It interacts

with the *FlightOffer Agent* in creating a list of specific offers that are registered with the *CIC*. Upon successful price negotiations the *Reservation Agent* creates and manages a reservation and is responsible for completing purchase. Observe that both the *FlightOffer Agent* and the *Reservation Agent* interact directly with the *GDS*. In this way they act as “wrapper agents” translating data between the outside world (the *GDS*) and the system.

3. General and travel ontologies

Let us first note that there exist at least four projects that attempt at creating an upper-level ontology defining the most general and most often used concepts. (1) The Cyc project is a proprietary system under development since 1985, which consists of foundation ontology and several domain-specific ontologies (called microtheories) [13]. (2) WordNet is a general dictionary designed for use in Natural Language Processing. It includes concepts, related to each other not only by subsumption relations, but also by other semantic relations such as part-of and cause. WordNet has never been axiomatized to make logical relations between concepts precise [21]. (3) Suggested Upper Merged Ontology (SUMO) is written in the SUO-KIF language and has been mapped to the WordNet lexicon [18]. (4) The SENSUS project is a thesaurus extension of the WordNet. It consists of more than 70,000 node taxonomy and a framework to add extra knowledge [17].

While, neither of these general projects provides us with “ontology of travel” there exist a number of attempts at defining one. (1) The Open Travel Alliance (OTA) specifications have been designed to serve: (a) as a common language for travel-related terminology and (b) as a mechanism for exchange of information between travel industry members [14]. It has to be stressed that OTA specifications focus on information exchange and can be viewed only as an implicit ontology. Therefore, to be usable, the “OTA ontology of travel” would have to be reverse engineered and formalized first. From our current perspective it is particularly interesting how the OTA tries to address the problem of tariff application and pricing description. Here, terms like *AirFareDisplay*, *AirPrice* and *AirRules* are defined and used to describe tariffs. Unfortunately, from offered examples it is not quite clear how OTA-based rules can be applied in an itinerary that consists of flights operated by several carriers. In the OTA we were able to locate *AirRules* message that is applicable to a specific fare basis code for a given airline, a specific date and a pair of cities. However, when several carriers are involved in transportation, fare basis as an only reference to tariff rules is often not sufficient. In the industry there exist rules that instruct the *GDS* how to compute the price for the itinerary that consist of flights operated by different carries (or even in the case when there is only one operating carrier, but the reservation is booked on classes that make necessary that two or more tariffs have to be combined to obtain final price for the itinerary). In this case, the reservation contains more than one fare basis and usually each will have different restrictions, while for the final ticket only one set of restrictions is applicable. Obviously, there exist rules how to chose which rules to apply, but it is unclear whether and how this is modeled in the OTA. (2) Mondeca’s [12] tourism ontology defines tourism concepts based on the WTO thesaurus. 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 to manage its travel ontology. Unfortunately, review of Mondeca’s site did not returned any results when it comes to IATA (International Air Transport Association) – a steering committee for all air transportation business. IATA is the regulatory body that defines way of coding

different air-entities (airplanes, airports, cities) and establishes procedures for information interchange among air transport companies. Fact that IATA definitions do not seem to be included in Mondeca makes it useless for our purpose. (3) 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 and Semantic Web technologies [19]. In addition to FIPA content language ontology, TAGA defines (a) basic travel concepts such as itineraries, customers, travel services, and service reservations, and (2) different types of auctions, roles participants play in them, and protocols used. Unfortunately, TAGA defines only very broad concepts (in not much detail) that are fairly unrealistic, due to the nature of TAGA simulations. Beside that, TAGA puts travel agency in the middle of the game, well representing what are the constraints on that side. On the other hand, AWSA – Airline Web Service Agent can only participate in a priceline auction (buyer initiated auction with goal to minimize buyers’ expenditure). This assumption is very unrealistic. In the real world, airlines’ goal is to maximize revenue. Therefore, if they are to offer tickets through an auction system, their typical goal would be to obtain price that would be better than that attained through the standard distribution system. In our opinion, a hotwire auction would be more suitable from airlines’ point of view (unless it is a special bargain pricing to get rid of excess inventory of empty seats, e.g. during off-season). Beside available auction type, here we find, once again, the fare applicability problem. As long as airlines impersonated in the *Shop agent* sell only point-to-point routes this problem can be remedied, since the revenue accounting remains within one organization, and it looks like TAGA is not ready to offer itineraries with more than one carrier involved in an air ticket. Though getting several tickets for itinerary that includes more than one carrier may be purchased for better price than one ticket, this introduces additional costs to the passenger in case he does not arrive on next flight as he was planning; due to various reasons (for example, flight delay or cancellation). In first case passenger does not have the right on compensation from the airline that caused delay and will incur the additional costs and in later case airline will participate in his costs. (4) Harmonize is an attempt at ontology-mediated integration of tourism systems following different standards [9]. 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. It is claimed that the 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). Goes without saying that this is a very interesting project, however, airline ticket sales are not included in the current version of Harmonize ontology. (5) Finally, a number of “minimalist” travel ontologies can be found within the DAML language portal [4]. For instance, the *Itinerary-ont* is an ontology for representing travel itineraries. It reuses the airport codes ontology and involves definitions of terms like *Aircraft*, *Class*, *Flight* etc. Another example is the *Trip Report Ontology* that defines *Airfare*, *Amount*, *Date*, etc., and models on-line sale. Here, steps of the on-line sale process are precisely described, and the fact that this ontology is used on Yahoo! for their on-line tickets sales, contributes to this opinion. At the same time, to be used in communication with the *GDS* it lacks certain details – as on-line sales do not introduce same restrictions as a direct sale from the *GDS*. There are several scenarios for airlines when it comes to on-line sales: (i) to publish special fares for on-line sale and give available seats in advance – this approach protects party that sells tickets (web portal) from direct communication with the *GDS*; (ii) to publish fares for on-line sales and bond third party to obtain software to communicate with the *GDS* (most *GDS*s offer interfacing software modules); (iii) offer direct sale through its own inventory. DAML ontology can be utilized only in first two scenarios as neither of the deals with direct communication to the *GDS*. Finally, even though DAML introduces an *Airfare* concept, it does not introduce tariff rules of application. This is only possible in the case when reserved tariffs are offered.

As it can be seen, we were not able to find travel ontology that would contain a fully developed air travel part and that could also interface with an actual *GDS*.

4. Proposed air-travel ontology

The above described attempts at developing ontology of travel and their lack of important features necessary for being able to deal with actual air-travel data seem to be related to their “academic” nature. According to our best knowledge, the only project that involved airline industry was the OTA specification (which, on the other hand, is not an ontology, but rather a specification of a “communication language” involved in travel reservations). To remedy this situation we have decided to follow the path undertaken in our earlier work [7] and use a very pragmatic approach. The main decision was that the actual IATA mandated data-set will guide our ontology design. Therefore, we applied a bottom-up approach in designing ontology suitable for communication between our ticket auction system [20] and the *GDS*. Our initial goal was to model reservation as presented in the AMADEUS system while utilizing definitions and rules set by IATA.

As a result the most basic classes included in our ontology are: *Airline*, *Airplane*, *Airport*, *AirportCode*, *Classes*, *Currency*, *FareBasis*, *DiscountAvailable**, *Flight*, *IATAdiscounts*, *Itinerary*, *LocationProperties**, *Passenger*, *MeanOfPayment**, *PaxIATATitle*, *Tariff*, *PhoneContacts**, and *TicketingIndicator*. Since this ontology will be joined with ontology designed in [5, 6, 7], classes marked with * were inherited from ontology used in that project. Let us now describe in more detail each of them.

Airline class contains basic info to describe an airline. Its properties include information necessary for coding and decoding the airline in the *GDS* [1]. We have also introduced a website URL, so passengers could find more information. Here is an instance of this class, in N3 notation:

```
#instance of Airline class
:YM a :Airline;
    :alName "Montenegro Airlines";
    :nIATACode "409";
    :icao "MGX";
    :homePage
        "http://www.montenegroairlines.com/".
```

Airplane class was defined in line with description given in [10]. Properties of this class contain information about manufacturer, name of the plane, category of the plane: engine type and number of engines on the plane, and number of available seats. For instance the Fokker 100 airplane would be described as:

```

#instances of Airplane
:100 a :Airplane;
      :manfact "Fokker";
      :aplname "Fokker 100";
      :cat      "2J";
      :noSeats  "95-109".
      :flightNo "100";
      :origin   apt:AptPodgorica;
      :destination apt:CharlesDeGaulle;
      :flightDate "2006-02-18";
      :departureTime "08:00";
      :arrivalTime  "10:45";
      :aircraft apln:100.

```

To create a reservation, three letter IATA codes for cities and airports are required. We decided to define *Airport* class as a subclass of *LocationProperties* class enabling in that way more information regarding airport to be presented to the passenger. Other proposed travel ontologies, discussed in Section 3, restrained themselves to the airport code and airport name only. This class has just one additional property – *airportCode* with range of *AirportCode* class. In this way we were able to keep additional info from the *LocationProperties* class and to have the required three letter code:

```

@prefix locp:<LocationProperties#>.
@prefix apc:<AirportCode#>.
@prefix :<#>.
:Airport rdfs:subClassOf
:LocationProperties.
:airportCode a rdf:Property;
              rdfs:domain :Airport;
              rdfs:range apc:AirportCode.
# instances of the Airport class
:aptPodgorica a :Airport;
              :airportCode apc:TGD.
:CharlesDeGaulle a :Airport;
                 :airportCode apc:CDG.

```

As for the *AirportCode* class, having in mind that we have direct access to AMADEUS where all airports codes are stored, it is not hard to gather instances of this class directly from there. This would enable us to have up-to-date list available in our system. Class *Classes* contains definitions of *CabinClass* and *BookingClass*. The first is used to define cabin class in the plane and the second is used to define specific class in which reservation is to be made. *BookingClass* is a subclass of *CabinClass*. Classes *Currency* and *IATAdiscounts* are defined in line with IATA Resolutions [15, 16]. We based our work on AMADEUS and our class follows flight display of this *GDS* having following properties: *carrier* (range *Airline*), *flightNo*, *origin* (range *Airport*), *destination* (range *Airport*), *flightDate*, *departureTime*, *arrivalTime*, *dateInd* (indication if the flight's duration is more then one day) and *aircraft* (range *Airplane*). An instance of this class:

```

:YM100_20060218 a :Flight;
                 :carrier airln:YM;

```

With a detailed Flight definition we can define *Itinerary* as bag of doubles of *Flight* for certain date and booking class. Simple example follows:

```

# instance of Itinerary class
:Rtng01 a :Itinerary;
        :flight flt:YM100_20060218;
        :class  cls:C.

```

An interesting task, that was not discussed in most ontologies listed in Section 3, was to define class for tariff. Airline tariffs are defined by a carrier or IATA for a specific route. Each tariff has its own set of rules. For consistency, we introduced tariffs as coded in AMADEUS. Since all tariffs have fare basis code we introduced *FareBasis* class that contains codes for all tariffs. Following are sample instances of this class:

```

#instances of FareBasis
:CRTYM a :FareBasis.
:COWYM a :FareBasis.
:YRTYM a :FareBasis.

```

Besides the code, *Tariff* class has following properties: *origin*, *destination* (these first two have *AirportCode* class as range), *carrier* with *Airline* class as range, *farebasis* with *FareBasis* class as range, *bk* with *BookingClass* as range, *currency* for currency of publication of the tariff (which is in the most of cases same like currency of payment with some exceptions as defined in [15]), *ow* – amount for one-way trip, *rt* – amount for return trip followed by all thirty categories that may be defined for a tariff. Categories define different restrictions on applicability of the tariff. Category properties are coded as per *ATPCO's* (Airline Tariff Publishing Company) recommendation. User friendly descriptions for some of them are eligibility, seasonability, flight application, advance reservation/ticketing, minimum and maximum stay, stopover, transfer, sales restrictions, penalties and ticket endorsement. These categories also include fare by rule and negotiated fares. This enables us to model all types of published tariffs currently available at the *GDSs*. Besides that, we are able to obtain a tariff together with rules based on an itinerary in any reservation involving any combination of flights and carriers provided that

a published tariff and an interline agreement between carriers exists. This is because we base our search on more data than the fare basis which represents just another peace of information that uniquely differentiates one tariff from another.

All these terms were defined to allow us to model reservation or, as it is referred among airlines a *PNR*, that consists of several obligatory elements – name of passenger, itinerary, contact detail and ticketing arrangement. When it comes to buying ticket, a *PNR* involves pricing data that is connected with a tariff. As follows from the above, we can now fully represent a *PNR*.

5. Concluding remarks

In this paper we have discussed how to interface a travel support system that utilizes ontologically demarcated data and that includes air ticket sales, with an actual ticket reservation and sales system used by real-world airlines. In such a situation, air-travel data should be represented in the form of air-travel ontology and integrated with the ontology utilized by the travel support system. However, such an ontology should also represent the way that air travel is conceptualized within actual travel reservation systems. Our analysis of existing travel ontologies indicates that they either are only implicitly represented in a system focused on other functionalities (e.g. the OTA), or have been created as a theoretical exercise and are not closely enough related to data that is actually required and utilized by the airline industry.

As a response we have presented an initial version of a complete air-travel ontology. Its roots are in the way that the AMADEUS *GDS* is dealing with airline ticket reservations as well as in IATA rules of air ticket processing. The proposed ontology is currently being developed in such a way that it can be fully integrated with ontologies that have been created for the travel support system.

References

- [1] Airline Coding Directory – Airline Designators, IATA
- [2] Bădică C., Ganzha M., Paprzycki M., UML Models of Agents in a Multi-Agent E-Commerce System. In: Proceedings of the ICEBE 2005 Conference, IEEE Press, Los Alamitos, CA, 56-61
- [3] City Code Directory, 43th Edition, Effective 9 December 2005 – 31 December 2006
- [4] DAML Ontologies, <http://www.daml.org>
- [5] Ganzha M., Gawinecki M., Paprzycki M., Gąsiorowski R., Pisarek S., Hyska W., Utilizing Semantic Web and Software Agents in a Travel Support System. In: A. F. Salam and Jason Stevens (eds.) Semantic Web Technologies and eBusiness: Virtual Organization and Business Process Automation, Idea Group (in press)
- [6] Gawinecki M., Gordon M., Paprzycki M., Szymczak M., Vetulani Z., Wright J., Enabling Semantic Referencing of Selected Travel Related Resources. In: W. Abramowicz (ed.) Proceedings of the 8 th International Conference on Business Information Systems (BIS 2005), Poznań University of Economics Press, Poznań, Poland, 2005, 271-288
- [7] Gordon M., Kowalski A., Paprzycki M., Pelech T., Szymczak M., Wąsowicz T., Ontologies in a Travel Support System. In: D. J. Bem et. al. (eds.) Internet 2005, Technical University of Wrocław Press, 2005, 285-300
- [8] Gordon M., Paprzycki M., Designing Agent Based Travel Support System. In: Proceedings of the ISPDC 2005 Conference, IEEE Computer Society Press, Los Alamitos, CA, 2005, 207-214
- [9] Harmonize, <http://deri.at/research/projects/e-tourism>
- [10] IATA Standard Schedules Information Manual, Mar 1, 2006 until Sep 30, 2006
- [11] Maes P., Agents that Reduce Work and Information Overload. Communications of the ACM, 37, 7, 1994, 31-40
- [12] Mondeca, <http://www.mondeca.com>
- [13] OpenCyc, <http://www.opencyc.org>
- [14] OpenTravel™ Alliance, Message Users Guide. 2005B Version 1.0, 2 December 2005
- [15] Passenger Tariff Coordination Conferences Manual, Composite, Dec 9, 2005 until Dec 31, 2006
- [16] Passenger Services Conference Resolutions Manual, Effective 1 June 2005 – 31 May 2006
- [17] SENSUS, <http://www.isi.edu/natural-language/projects/ONTOLOGIES.html>
- [18] SUMO, <http://www.ontologyportal.org>
- [19] TAGA, <http://www.agentcities.org>
- [20] Vukmirovic M., Ganzha M., Paprzycki M., Developing a Model Agent-based Airline Ticket Auctioning System, LNAI (in press)
- [21] WordNet, <http://www.daml.org/ontologies/196>