DESIGNING ONTOLOGY FOR THE OPEN TRAVEL ALLIANCE AIRLINE MESSAGING SPECIFICATION

Mladenka Vukmirovic
Industry Development Department, Montenegro Airlines
Beogradska 10, 81000 Podgorica, Montenegro
e-mail: mladenka.vukmirovic@mgx.cg.yu

Marcin Paprzycki
Computer Science Institute, SWPS
Chodakowska 19/31, 03-815 Warsaw, Poland
and
Systems Research Institute, Polish Academy of Science
Newelska 6, 01-447 Warsaw, Poland
e-mail: marcin.paprzycki@ibspan.waw.pl

Michał Szymczak
Department of Mathematics and Computer Science
Adam Mickiewicz University
Umultowska 87, 61-614 Poznań, Poland
e-mail: wildmike@interia.pl

ABSTRACT

Advancement of usage of ontologies requires not only creation of new ontologies (e.g. in a top-down manner), but also ontological support for existing domain-specific real-world standards. One of such standards that gains popularity is the Open Travel Alliance (OTA) messaging system that defines, among others, the way that entities should communicate about air travel related issues. The aim of this note is to outline our efforts leading toward creating an air-travel ontology that would match the OTA messaging specification as well as satisfy procedures described in IATA manuals.

1 INTRODUCTION

Recently we have proposed an airline ticket auctioning system that could be used as a part of an agent-based Travel Support System (TSS) [4, 14, 15]. In the TSS, travelers find complete support of their needs including, among others, items like restaurant information, historical points of interest, local weather etc. It is important to emphasize that the ticket auctioning system and the Travel Support System were developed separately. The ticket auctioning system was conceived in the context of a model agent-based e-commerce system (see [1, 2] and references to our earlier work contained there). During its creation we were aware that we should interface our agents with the Global Distribution System (GDS) used by airlines to sell their inventory, but we did not pursue this problem further [14, 15]. Meanwhile, in the TSS we made ontologically demarcated data the centerpiece of our system and have developed ontologies of a *hotel* and a *restaurant* [3]. When attempting at merging the two systems, we have realized that, on the one hand we have to deal with data representation used in airline distribution systems, which is regulated by IATA and which has nothing to do with ontologies, and with RDF demarcated instances of our travel ontology on the other. The natural way of solving this problem was to find or develop a suitable ontology to represent airline travel data. After an in-depth research of existing air-travel ontologies we have concluded that none of them is capable of addressing issues that arise when real-life IATA rules come to play [4, 6, 7, 8, 9, 10, 11]. Therefore we decided to develop a new ontology that would reuse as much as possible from the ontology used in the TSS, as well as existing air-travel ontologies (initial results of our work have been summarized in [14, 15]). Meanwhile we have found that changes take place in data representation used in airlines' proprietary systems. They are spearheaded by IATA's movement to utilize new technologies such as XML (and lately there is even talk about ontologies). Furthermore, as we have studied existing air-travel related ontologies, we became interested in the Open Travel Alliance (OTA) messaging specification. It needs to be stressed that this messaging standard was developed with intent to model the process (and not persist) data. Therefore this is not an ontology per se. However, it proved to be very suitable for data exchange; even to the extent that some GDSs already offer interfaces based on it. Therefore, we have decided to modify our air-travel ontology in such a way that it would be able to persist data defined by the OTA messaging system. Ontology with these features would enable us to exchange data between the TSS and the GDSs that offer an OTA-based interface.

2 OTA AND OTA AIR-MESSAGES

The Open Travel Alliance (OTA) is a non-profit organization working to establish a common electronic vocabulary for exchange of travel information. It defines messages using the eXtensible Markup Language (XML). Its 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 [5]. The OTA Air Messages standard specifies structure and elements of different scenarios involved in selling air travel tickets. Let us note that since this is a specification of messaging, it does not cover any other operations involved in selling air-tickets (e.g. airfare calculations). These operations have to be treated separately. OTA messages have been proposed as pairs of request and response messages (RQ / RS below). Let us summarize their main features (their complete description can be found in [12]).

OTA_AirAvailRQ/RS – establishes airline flight availability for a city pair, specific date, specific number and type of passengers. The request can also be narrowed to a specific airline, flight or booking class. Optional requested information can include: time / time window, connecting cities, client preferences (airlines, cabin, flight types etc.). The response message (*RS*) contains flight availability. Furthermore, a set of origin and destination options is returned, each of which contains one or more (connecting) flights that serve that city pair. For each flight information about: origin and destination airports, departure and arrival date/times, booking class availability, equipment, meal information and code-share information is returned.

OTA_AirBookRQ/RS – requests to book a specific itinerary for one or more identified passengers. The message contains optional pricing information, allowing the booking class availability and pricing to be rechecked as part of the booking process. Optional requested information can include: seat and meal requests, Special Service Requests (SSR), Other Service Information (OSI), remarks, fulfillment information – payment, delivery details, type of ticket desired. If booking is successful, the RS message contains the itinerary (including the directional indicator, status of booking, and number of passengers), passenger and pricing information sent in the request, along with a booking reference number (PNR Locator) and the ticketing information. The RS echoes back received information with additional information – booking reference from the GDS through which reservation was created.

OTA_AirFareDisplayRQ/RS – allows a client to request information on fares, which exist between a city pair for a particular date or date range. No inventory check for available seats on flights is performed by the server before the RS is send back. The request can optionally contain information indicating that a more specific response (e.g. passenger information, specific flight information and information on the types of fares that the client is interested in) is required. The RS message repeats *FareDisplayInfo* elements, each of which contains information on a specific fare contract including airline, travel dates, restrictions and pricing. It can also return information on other types of fares that exist, but have not been included in the response.

OTA_AirFlifoRQ/RS – requests updated information on the operation of a specific flight (it requires the airline, flight number and departure date; the departure and arrival airport locations can be also be included). The RS includes real-time flight departure and arrival information. It also includes: departure airport, arrival airport, marketing and operating airline names; when applicable, flight number, type of equipment, status of current operation, reason for delay or cancellation, airport location for diversion of flight, current departure and arrival date and time, scheduled departure and arrival date and time, duration of flight, flight mileage, baggage claim location.

OTA_AirLowFareSearchRQ/RS – requests priced itinerary options for flights between specific city pairs on certain dates for a specific number and types of passengers. Optional requested information can include: time / time window, connection points, client preferences (airlines, cabin, flight types etc.), flight type (nonstop or direct), number of itinerary options desired. The RS contains a number of *Priced Itinerary* options. Each includes: a set of available flights matching the client's request, pricing information including taxes and full fare breakdown for each passenger type, ticketing information – ticket advisory information and ticketing time limits, fare basis codes and the information necessary to make a rules entry.

OTA_AirPriceRQ/RS – requests pricing information for specific flights on certain dates for a specific number and type of passengers. The message allows for optional information such as fare restriction preferences and negotiated fare contract codes to be included. The pricing request contains information necessary to perform an availability / sell from availability / price series of entries for an airline CRS or GDS. The RS contains a *Priced Itinerary* that includes: set of flights, pricing information including taxes and full fare breakdown for each passenger type, ticketing information, fare basis codes and the information necessary to make a fare rules entry.

OTA_AirRulesRQ/RS – requests text rules for a specific fare basis code for an airline and a city pair for a specific date. Negotiated fare contract codes can be included in the request. The RS contains a set of, human readable, rules, identified by their codes.

OTA_AirSchedulesRQ/RS – provides customer, or a third party, with ability to view flight schedules. It requires specification of the departure and arrival cities and a specific date. It offers flight information on airlines that provide service between requested cities and could be used when customer: (1) wants to determine what airlines offer service to/from specific destinations, (2) is looking for a specific flight number – by entering the arrival and departure cities, and the approximate arrival or departure time, specific flight number can be found, (3) needs to determine the days of the week that service is scheduled to and from requested destinations, (4) wants to determine aircraft type used to fly that route. Message may request other information that customers are interested in: meal service, duration of flight, on-time statistics and if smoking is allowed. In addition, these messages provide foundation for electronic timetables.

OTA_AirSeatMapRQ/RS – displays seats available on a given flight, as well as their location within the aircraft. It is used o make seat assignments as it identifies all information necessary to request and return an available seat map for a particular flight. Types of information for the seat map request include: airline, flight number, date of travel, class of service and frequent flier status. The RS includes: flight, aircraft and seat description information.

OTA_AirBookModifyRQ/OTA_AirBookRS – requests to modify an existing booking file. It contains all elements of the *OTA_AirBookRQ* plus a general type of modification, i.e. name change, split, cancel or other; as indicated with the attribute *ModificationType*. The modification operation on different elements is either indicated with the existing attribute *Status* (for air segments, SSR's and seat requests) or with attribute *Operation* of type *ActionType* for other elements (i.e. other service information, remarks or *AirTraveler* elements). In the *AirBookModifyRQ*, all data to be changed is submitted and in the *AirReservation* element all existing data may be submitted. This allows the receiving system to perform a consistency check before updating the booking file (but to keep the message small, this part can be omitted). Changes to a booking (1) may result in required updates of the ticket (e.g. revalidation), (2) may imply charges for the change, (3) the pricing may change, and/or (4) some fees may need to be collected. Pricing and fulfillment details required to achieve results of *AirBookModify* ticketing, are out of scope and are omitted. The RS confirms changes in the itinerary.

3 PROPOSED ONTOLOGY

As indicated above, in our research [14, 15] we have established that existing air-travel ontologies have been designed as "academic" demonstrator systems – rather than with the goal of actually working within the context of real-life airline reservation systems – and this explains lack of important features when it comes to dealing with genuine air travel data. According to our best knowledge, the only project that actually involves airline industry is the OTA specification (which, as stated above, is only a messaging specification). Therefore, we decided to create new ontology that would: (1) utilize IATA mandated data descriptions and recommended practices; (2) utilize as much as possible from existing travel ontologies – as long as they follow IATA practices, (3) match features included in the OTA specification, and (4) be synchronized with our existing travel ontology. To achieve this goal we have applied a bottom-up approach and our initial goal was to model reservations occurring in the AMADEUS global distribution system.

IN the proposed ontology we have divided main classes into following groups: *AirInfrastructureCodes, AirTravel, AirInfrastructure* and *AirtravelCodes*. *AirInfrastructure* group encloses most basic terms related to air travel industry such as *Airline*, *Airplane* and *Airport*. While all three are defined in line with specifications presented in [6, 7, 11], the latest (*Airport*) is a subclass of our *OutdoorLocation* class that was designed for the TSS [4]. In this way it is possible for the traveler to obtain more data regarding the airport then the city name, which usually is the only information that can be obtained from other airline travel related ontologies. *AirInfrastructureCodes* group contains, used in other classes, codes for airports and countries. Included classes are *ISOCountryCode* and *AirportCode*. *AirTravelCodes* group comprises industry codes used in GDSs and CRSs for itinerary reservation and ticket issuance: *IATATicketIndicator*, *IATAStatusCode*, *CabinClass*, *BookingClass*, *IATAFareBasis*, *MealCode*, *SSRCode*, *SSRMealCode*, *TicketDestignator* (details can be found in [6, 7 , 8, 9, 10, 11]). Finally, the *AirTravel* group takes care of upper-level terms that define more complex objects used in air travel systems. Following classes are included in this group: *OfficeID*, *TerminalID*, *AgentCredentials* – that define credentials of the GDS/CRS user, *AvailabilityDisplay* – that defines available flight options for a certain route, *Flight* - with usual properties together with status statistics, *IATAItinerary* – that defines itinerary for the passenger, *PNR* – Passenger Name Record or, simply described, a reservation with all details of the passenger, the itinerary, special requests and the GDS/CRS locator code, *Pricing* – that describes available prices for a certain route with or without taxes included, *SeatMapPlan* – for a certain flight, *Tariff* - with *Category* properties that are coded as in the *ATPCO*'s (Airline Tariff Publishing Company) recommendation, and *TimetableDisplay* – with timetable of different airlines for a certain route.

As stated above some classes were inherited or used as upper level classes from the TSS. These classes are: *OutdoorLocation*, *IATADiscountCodes**, *MeanOfPayment*, *FareTax*, *Discounts*, *DiscountCodes*, *IATATaxCodes**, *NameRecord*, and *PersonTitle*. Marked with * are classes that were sub classed from classes inherited from the TSS.

One additional, very important, concept in traveling is currency. At first we designed a very simple class that contained only the currency code. Promptly this showed to be insufficient as air travel currency application involved some complicated restrictions. As in the case of air travel ontology, we made an effort to find an already existing ontology of currency, and inject it into our project. We studied several currency ontologies (more details can be found in [13]) and found out that ontology used in Cambia webservice [3] was the most appropriate one. Unfortunately, it was rather broad, and furthermore we had to modify it so that it could be used for currency conversion guided by the IATA conversion rules [9].

Let us stress that since the OTA was defined as a messaging system used for information exchange, while the proposed ontology was created with intention to describe persistent data in our system, therefore quite often more then one class from our ontology has to be used in association with a single OTA message. As request (RQ) messages contains only data used to make a query, let us illustrate how the RS message matches with the proposed ontology in the case of the *OTA_AirAvailRS*. In our ontology an equivalent class is *AvailabilityDisplay*. Both are used with regard to available flight information and contain flight numbers, arrival/departure details as well as available seats etc. In Table 1 we depict how elements of the message match elements in our ontology.

OTA message	*OTA message element*	*Related classes from our ontology*
`OTA_AirAvailRS`	OriginDestinationOption	
	FlightSegment atributes *DepartureDateTime* *ArrivalDateTime* *FlightNumber* *JourneyDuration* *SmokingAllowed* *OnTimeRate* *Ticket*	Flight class properties *departureTime* *arrivalTime* *flightNo* *journeyduration* *smokingAlowed* *onTime* *etkt*
	DepartureAirport LocationCode	Flight class property *origin*
	ArrivalAirport LocationCode	Flight class property *destination*
	Equipment *AirEquipType, ChangeofGauge*	Flight class property *aircraft* with range airplane with further details of equipment
	MarketingAirline	Relates to *CodeShare* class that contains data regarding marketing carrier
	BookingClassAvail *ResBookDesigCode,* *ResBookDesigQuantity*	*AvailableClassElement, AvailableClasses, AvailableFlightElement* and *AvailableFlights* all used to show hierarchy for connecting flights and classes available in plane. First relates to *class* of *AvailableClassElement* and later relates to *noAvailableSeats* property of the same class
	Meal *MealCode*	*MealCode* class

Table 1. *Matching the OTA message with the air-travel ontology*

6 CONCLUSION

In this note we have discussed crucial aspects of design of complete air-travel ontology. Proposed ontology goes beyond what is proposed thus far in existing air-travel ontologies, as it is based on IATA manuals and OTA messaging system. Therefore, when completed (currently the proposed merged travel ontology it is available for comments at: http://agentlab.swps.edu.pl) can be used to interface our travel support system with an actual GDS, which is one of goals of our project. Specifically, we will develop a subsystem that will parse OTA RQ messages originating from the GDS into SPARQL queries to be executed in the TSS. Obtained responses (in form of RDF triples) will be parsed back to OTA RS messages.

References

[1] Bădică, C., Bădită, A., Ganzha, M., Paprzycki, M., Developing a Model Agent-based E-commerce System, in: Jie Lu et. al. (eds.) E-Service Intelligence - Methodologies, Technologies and Applications (to appear)

[2] Bădică C., Ganzha M., Paprzycki M., UML Models of Agents in a Multi-Agent Ecommerce System. In: Proceedings of the ICEBE 2005 Conference, IEEE Press, Los Alamitos, CA, 56-61

[3] Cambia Service, http://zurich.agentcities.whitestein.ch/Services/Cambia.html

[4] 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

[5] http://www.opentravel.org/about.cfm

[6] IATA Airline Coding Directory – Airline Designators, 70th Edition

[7] IATA City Code Directory, 43rd Edition, Effective 9 December 2005 – 31 December 2006

[8] IATA Passenger Services Conference Resolutions Manual, 24th Edition, Effective 1 June 2005 – 31 May 2006

[9] IATA Passenger Tariff Coordination Conferences Manual, Composite, Dec 9, 2005 until Dec 31, 2006

[10] IATA Reservation Service Manual, 23rd Edition

[11] IATA Standard Schedules Information Manual, Mar 1, 2006 until Sep 30, 2006

[12] OpenTravelTM Alliance, Message Users Guide. 2005B Version 1.0, 2 December 2005

[13] Szymczak M., Gawinecki M., Vukmirovic M., Paprzycki M., Ontological reusability in state-of-the-art semantic languages, Proceedings of the XVIII Summer School of PIPS (to appear)

[14] Vukmirovic M., Ganzha M., Paprzycki M.: Developing a Model Agent-based Airline Ticket Auctioning System. In: Proceedings for the IIPWM Conference, LNAI

[15] Vukmirovic M., Szymczak M., Ganzha M., Paprzycki M.: Utilizing Ontologies in an Agent-based Airline Ticket Auctioning System. In: Proceedings of the 28th ITI Conference, IEEE Computer Society Press, Cavtat, Croatia, 385-390