An ASP-Based System for e-Tourism

Salvatore Maria Ielpa[1], Salvatore Iiritano[2], Nicola Leone[1], and Francesco Ricca[1]

[1] Dipartimento di Matematica, Università della Calabria, 87030 Rende, Italy
{s.ielpa,leone,ricca}@mat.unical.it
[2] Exeura Srl, Via Pedro Alvares Cabrai - C.da Lecco 87036 Rende (CS), Italy
salvatore.iiritano@exeura.com

Abstract. In this paper we present the IDUM system, a successful application of logic programming to e-tourism. IDUM exploits two technologies that are based on the state-of-the-art ASP system DLV: (i) a system for ontology representation and reasoning, called OntoDLV; and, (ii) HiLeX a semantic information extraction tool. The core of IDUM is an ontology which models the domain of the touristic offers. The ontology is automatically populated by extracting the information contained in the touristic leaflets produced by tour operators. A set of specifically devised ASP programs is used to reason on the information contained in the ontology for selecting the holiday packages that best fit the customer needs. An intuitive web-based user interface eases the task of interacting with the system for both the customers and the operators of a travel agency.

1 Introduction

In the last few years, the tourism industry has strongly modified marketing strategies with the diffusion of e-tourism portals in the Internet. Big tour operators are exploiting new technologies, such as web portals and e-mails, in order to both simplify their advertising strategies and reduce the selling costs. The efficacy of e-tourism solutions is also witnessed by the continuously growing community of e-buyers that prefer to surf the Internet for buying holiday packages. On the other hand, traditional travel agencies are undergoing a progressive loss of marketing competitiveness. This is partially due to the presence of web portals, which basically exploit a new market. Indeed, Internet surfers often like to be engaged in self-composing their holiday by manually searching for flights, accommodation etc. Instead, the traditional selling process, whose strength lies in both direct contact with customer and knowledge about customer habits, is experiencing a reduced efficiency. This can be explained by the increased complexity of matching demand and offer. Indeed, travel agencies receive thousand of e-mails per day from tour operators containing new pre-packaged offers. Consequently, the employees of the agency cannot know all the available holiday packages (since they cannot analyze all of them). Moreover, customers are more demanding than in the past (e.g. the classic statement "I like the sea" might be enriched by "I like snorkeling", or "please find an hotel in Cortina" might be followed by "featuring beauty and fitness center") and they often do not declare immediately all their preferences and/or constraints (like, e.g., budget limits, preferred transportation mean or accommodation etc.). This knowledge of customers preferences plays a central role in the traditional selling process; However,

E. Erdem, F. Lin, and T. Schaub (Eds.): LPNMR 2009, LNCS 5753, pp. 368–381, 2009.

the task of matching this information with the large unstructured e-mail database is both difficult to carry out in a precise way and is time consuming. Consequently, the seller is often unable to find the best possible solution to the customer needs in a short time.

The goal of the IDUM project is to devise a system that addresses the above-mentioned causes of inefficiency by offering:

(i) an automatic extraction and classification of the incoming touristic offers (so that they are immediately available for the seller), and
(ii) an "intelligent" search that combines knowledge about users preferences with geographical information, and matches user needs with the available offers.

We could achieve the goal by exploiting Answer Set Programming (ASP) [1]. ASP is a powerful logic programming language, which is very expressive in a precise mathematical sense; in its general form, allowing for disjunction in rule heads and nonmonotonic negation in rule bodies, in a fully declarative way ASP can represent *every* problem in the complexity class Σ_2^{P} and Π_2^{P} (under brave and cautious reasoning, respectively) [2]. In particular, the core functionalities of IDUM were based on two technologies[1] relying on the state-of-the-art ASP system DLV [3]:

- OntoDLV [4,5] a powerful ASP-based ontology representation and reasoning system; and,
- HiLeX [6,7,8], an advanced tool for semantic information-extraction from unstructured or semi-structured documents.

More in detail, in the IDUM system, behind the web-based user interface (that can be used by both employees of the agency and customers), there is an "intelligent" core that exploits an OntoDLV ontology for both modeling the domain of discourse (i.e., geographic information, user preferences, and touristic offers, etc.) and storing the available data. The ontology is automatically populated by extracting the information contained in the touristic leaflets produced by tour operators. It is worth noting that, offers are mostly received by the travel agency in a dedicated e-mail account. Moreover, the received e-mails are human-readable, and the details are often contained in email-attachments of different format (plain text, pdf, gif, or jpeg files) and structure that might contain a mix of text and images. The HiLeX system allows for automatically processing the received contents, and to populate the ontology with the data extracted from touristic leaflets. Once the information is loaded on the ontology, the user can perform an "intelligent" search for selecting the holiday packages that best fit the customer needs. IDUM tries to mimic the behavior of the typical employee of a travel agency by exploiting a set of specifically devised logic programs that "reason" on the information contained in the ontology.

In the remainder of the paper, we first introduce the employed ASP-based technologies; then, in Section 3, we describe how the crucial tasks have been implemented. We show the architecture of the IDUM system in Section 4, and we draw the conclusion in Section 6.

[1] Both systems are developed by Exeura srl, a technology company working on analytics, data mining, and knowledge management, that is working on their industrialization finalized to commercial distribution.

2 Underlying ASP-Based Technologies

The core functionalities of the e-tourism systems IDUM were based on two technologies relying on the DLV system [3]: OntoDLV [4,5] a powerful ASP-based ontology representation and reasoning system; and, HiLeX [6,7,8], an advanced tool for semantic information-extraction from unstructured or semi-structured documents.

In the following we briefly describe both OntoDLV and HiLeX, the reader interested in a more detailed description is referred to [4,5] and [6,7,8], respectively.

2.1 The OntoDLV System

Traditional ASP in not well-suited for ontology specifications, since it does not directly support features like classes, taxonomies, individuals, etc. Moreover, ASP systems are a long way from comfortably enabling the development of industry-level applications, mainly because they lack important tools for supporting programmers. All the abovementioned issues were addressed in OntoDLV [4,5] a system for ontologies specification and reasoning. Indeed, by using OntoDLV, domain experts can create, modify, store, navigate, and query ontologies; and, at the same time, application developers can easily develop their own knowledge-based applications on top of OntoDLV, by exploiting a complete Application Programming Interface [9]. OntoDLV implements a powerful logic-based ontology representation language, called OntoDLP, which is an extension of (disjunctive) ASP with all the main ontology constructs including classes, inheritance, relations, and axioms. In OntoDLP, a *class* can be thought of as a collection of individuals who belong together because they share some features. An individual, or *object*, is any identifiable entity in the universe of discourse. Objects, also called class instances, are unambiguously identified by their object-identifier (oid) and belong to a class. A class is defined by a name (which is unique) and an ordered list of attributes, identifying the properties of its instances. Each attribute has a name and a type, which is, in truth, a class. This allows for the specification of *complex objects* (objects made of other objects). Classes can be organized in a specialization hierarchy (or data-type taxonomy) using the built-in *is-a* relation (*multiple inheritance*). Relationships among objects are represented by means of *relations*, which, like classes, are defined by a (unique) name and an ordered list of attributes (with name and type). OntoDLP relations are strongly typed while in ASP relations are represented by means of simple flat predicates. Importantly, OntoDLP supports two kind of classes and relations: (base) classes and (base) relations, that correspond to basic facts (that can be stored in a database); and *collection* classes and *intensional* relations, that correspond to facts that can be inferred by logic programs; in particular, *collection classes* are mainly intended for object reclassification (i.e., for repeatedly classifying individuals of an ontology). For instance, the following statement declares a class modeling customers, which has six attributes, namely: *firstName, lastName*, and *status* of type string; *birthDate* of type Date; a positive integer *childNumber*, and *job* which contains an instance of another class called Job.

```
class Customer (firstName: string, lastName: string,
    birthDate: Date, status: string,
    childNumber: positive integer, job: Job).
```

As in ASP, logic programs are sets of logic rules and constraints. However, OntoDLP extends the definition of logic atom by introducing class and relation predicates, and complex terms (allowing for a direct access to object properties). This way, the OntoDLP rules merge, in a simple and natural way, the declarative style of logic programming with the navigational style of the object-oriented systems. In addition, logic programs are organized in *reasoning modules*, to take advantage of the benefits of modular programming. For example, with the following program we single out the pairs of customers having the same age

```
module (CustomersWithTheSameAge) {
  sameAge(C1,C2,D) :- C1: Customer(birthDate:D),
                      C2: Customer(birthDate:D).
}
```

The core of OntoDLV is a rewriting procedure [5] that translates ontologies, and reasoning modules to an equivalent standard ASP program which, in the general case, runs on state-of-the art ASP system DLV [3].

OntoDLV features an advanced persistency manager allows one to store ontologies transparently both in text files and internal relational databases; while powerful type-checking routines are able to analyze ontology specifications and single out consistency problems. Importantly, if the rewritten program is stratified and non-disjunctive [1,10,11] (and the input ontology resides in relational databases) then the evaluation is carried out directly in mass memory by exploiting a specialized version of the same system, called DLV ^{DB} [12]. Note that, since class and relation specifications are rewritten into stratified and non-disjunctive programs, queries on ontologies can always be evaluated by exploiting a DBMS. This makes the evaluation process very efficient, and allows the knowledge engineer to formulate queries in a language more expressive than SQL. Clearly, more complex reasoning tasks (whose complexity is NP/co-NP, and up to Σ_2^P/Π_2^P) are dealt with by exploiting the standard DLV system instead.

2.2 The HiLeX System

HiLeX [6,7,8] is an advanced system for ontology-based information extraction from semi-structured and unstructured documents, that has been already exploited in many relevant real-world applications. The HiLeX system implements a semantic approach to the information extraction problem based on a new-generation semantic conceptual model by exploiting:

- ontologies as knowledge representation formalism;
- a general document representation model able to unify different document formats (html, pdf, doc, ...);
- the definition of a formal attribute grammar able to describe, by means of declarative rules, objects/classes w.r.t. a given ontology.

Most of the existing information extraction approaches do not work in a semantical way and they are not independent of the specific type of document they process. Contrariwise, the approach implemented in HiLeX confirms that it is possible recognize, extract and structure relevant information form heterogeneous sources.

HiLeX is based on OntoDLP for describing ontologies, since this language perfectly fits the definition of semantic extraction rules.

Regarding the unified document representation, the idea is that a document (unstructured or semi-structured) can be seen as a suitable arrangement of objects in a two-dimensional space. Each object has its own semantics, is characterized by some attributes and is located in a two-dimensional area of the document called *portion*. A portion is defined as a rectangular area univocally identified by four cartesian coordinates of two opposite vertices. Each portion "contains" one or more objects and an object can be recognized in different portions.

The language of HiLeX is founded on the concept of *ontology descriptor*. A "descriptor" looks like a production rule in a formal attribute grammar, where syntactic items are replaced by ontology elements, and where extensions for managing two-dimensional objects are added. Each descriptor allows us to describe: (i) an ontology object in order to recognize it in a document; or (ii) how to "generate" a new object that, in turn, may be added in the original ontology.

Note that an object may also have more than one descriptor, thus allowing one to recognize the same kind of information when it is presented in different ways.

3 The IDUM System

In this section we describe the core of the IDUM system and its innovative features based on ASP. IDUM is an e-tourism system, conceived for classifying and driving the search of touristic offers for both travel agencies operators and their customers.

IDUM, like other existing portals, has been equipped with a proper (web-based) user interface; but, behind the user interface there is an "intelligent" core that exploits knowledge representation and reasoning technologies based on ASP. In IDUM (see Figure 1, the information regarding the touristic offers provided by tour operators is received by the system as a set of e-mails. Each e-mail might contain plain text and/or a set of leaflets, usually distributed as pdf or image files which store the details of the offer (e.g., place, accommodation, price etc.). Leaflets are devised to be human-readable,

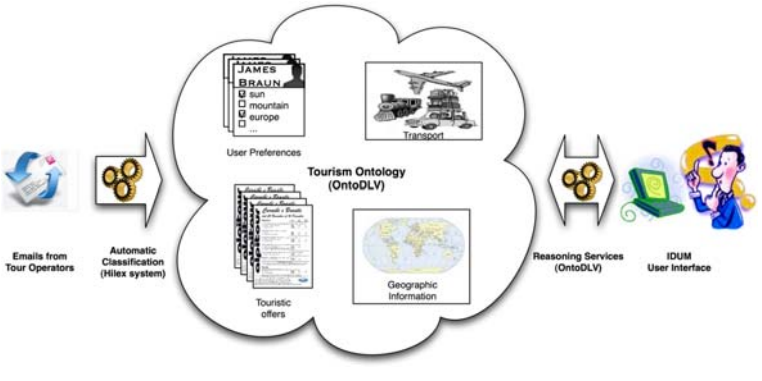

Fig. 1. The IDUM System

might mix text and images, and usually do not have the same layout. E-mails (and their content) are automatically processed by using the HiLeX system,[2] and the extracted data about touristic offers is used to populate an OntoDLP ontology that models the domain of discourse: the *"tourism ontology"*. The resulting ontology is then analyzed by exploiting a set of reasoning modules (ASP programs) combining the extracted data with the knowledge regarding places (geographical information) and users (user preferences) that we inserted in the tourism ontology. The system mimics the typical deductions made by a travel agency employee for selecting the most appropriate answers to the user needs.

It is worth pointing out that, the core of the system is based on ASP technologies, since both the two most innovative and important features: automatic extraction of touristic offers and intelligent search, were implemented by exploiting OntoDLV and HiLeX.

In the following sections, we briefly describe the tourism ontology and the implementation of the above-mentioned ASP-based features.

3.1 The Tourism Ontology

The "tourism ontology" has been specified by analyzing the nature of the input (we studied the characteristics of several touristic leaflets) with the cooperation of the staff of a real touristic agency, who were repeatedly interviewed. In this way, we could model the key entities that describe the process of organizing and selling a complete holiday package: in particular, the "tourism ontology" models all the required information, such as user profile, geographic information, kind of holiday, transportation means, etc. In Figure 2, we report some of the most relevant classes and relations that constitute the tourism ontology. In detail, the class *Customer* allows us to model the personal information of each customer, while a number of relations is used to model user preferences, like *CustomerPrefersTrip* and *CustomerPrefersMeans*, which associate each customer to his preferred kind of trip, and his preferred transportation means, respectively. The kind of trip is represented by using a class *TripKind*. Examples of *TripKind* instances are: *safari*, *sea_holiday*, etc. In the same way, e.g. *airplane, train*, etc. are instances of the class *TransportationMean*. Geographical information is modeled by means of the class *place*, which has been populated with the information regarding more than a thousand touristic places. Moreover, with each place is associated a kind of trip by means of the relation *PlaceOffer* (e.g. Kenya offers safari, Sicily offers both sea and sightseeing). Importantly, the natural a *part-of* hierarchy of places is easily modeled by using the intensional relation *Contains*. This allowed us to assert manually only some basic facts and to obtain all the basic inclusions in a simple yet efficient way. Indeed, the full hierarchy is computed by evaluating a rule (which, basically, encodes the transitive closure).

The mere geographic information is, then, enriched by other information that is usually exploited by travel agency employees for selecting a travel destination. For instance, one might suggest avoiding sea holidays in winter, or going to India during the

[2] Note that, e-mails are the main source of touristic offers, but IDUM can deal with sources different from e-mails; indeed, e-mails are "unwrapped": attachments are analyzed, enclosed external links are followed and corresponding web pages analized.

```
class Customer (firstName: string, lastName: string,
    birthDate: Date, status: string,
    childNumber: positive integer, job: Job).
relation CustomerPrefersTrip ( cust:Customer, kind: TripKind ).
relation CustomerPrefersMeans ( cust:Customer,
    means: TransportationMean ).  ...

class Place ( description:string ).
intentional relation Contains ( pl1:place, pl2:place )
{ Contains(P1,P2) :- Contains(P1,P3), Contains(P3,P2).
 Contains('Europe', 'Italy'). Contains('Italy', 'Sicily').
 Contains('Sicily', 'Palermo'). ... }

relation PlaceOffer( place: place, kind: tripKind ).
relation SuggestedPeriod ( place:place, period: positive integer ).
relation BadPeriod ( place:place, period: positive integer ).

class TouristicOffer( start: Place, destination: Place,
    kind: TripKind, means: TransportationMean,
    cost: positive integer, fromDay: Date, toDay: Date,
    maxDuration: positive integer, deadline: Date,
    uri: string, tourOperator: TourOperator ).

class TransportationMean ( description: string ).
class TripKind ( description: string ).
...
```

Fig. 2. Main entities of the touristic ontology

wet monsoon period; whereas, one should be recommanded a visit to Sicily in summer. This was encoded by means of the two relations *SuggestedPeriod* and *BadPeriod*.

Finally, the *TouristicOffer* class contains an instance for each available holiday package. The instances of this class are added either automatically, by exploiting the HiLeX system (see next section), or manually by the personnel of the agency.

3.2 Automatic Extraction of Touristic Offers

Touristic offers are mainly available in digital format and they are received via e-mail. It is usual that more than a hundred emails per day crowd the mail folder of a travel agency, and often the personnel cannot even analyze the entire in-box. This causes a loss of efficiency in the selling process, because some interesting offers might be ignored. Note that, most of the information is contained in pdf, gif or jpeg files attached to the e-mail messages, and this strongly limits the efficacy of standard search tools like, e.g., the ones provided by e-mail clients.

To deal with this problem, the IDUM system has been equipped with an automatic classification system based on HiLeX. Basically, after some pre-processing steps, in which e-mails are automatically read from the inbox, and their attachments are properly handled (e.g. image files are analyzed by using OCR software), the input is sent to HiLeX. In turn, HiLeX is able to both extract the information contained in the e-mails

Fig. 3. Extracting offer information

and populate the *TouristicOffer* class. This has been obtained by encoding several ontology descriptors (actually, we provided several descriptors for each kind of file received by the agency). For instance, the following descriptor:

```
<TouristicOffer (Destination, Period)> ->
    <X:place(XX)>{Destination:=X;}
    <X:date(XX)>{Period:=X;}
    SEPBY <X:separator()>.
```

extracts from the leaflet in Figure 3(a) that the proposed holiday package regards trips to both the Caribbean islands and Brazil. Moreover, it also extracts the period in which this trip is offered. The extracted portions are outlined in Figure 3(b). The result of the application of this descriptor are two new instances of the TouristicOffer class.

3.3 Personalized Trip Search

The second crucial task carried out in the IDUM system is the personalized trip search. This feature has been conceived to make simpler the task of selecting the holiday packages that best fit the customer needs. We tried to "simulate" the deductions made by an employee of the travel agency in the selling process by using a set of specifically devised logic programs. In a typical scenario, when a customer enters the travel agency, an employee tries to understand his current desires and preferences at first; then, the seller has to match the obtained information with a number of pre-packaged offers. Actually, a number of candidate offers are proposed to the customer and, then, the employee has to understand his needs by interpreting the preferences of the customer. Customer preferences depend on his personal information (age, gender, marital status, lifestyle, budget, etc.), but also on his holiday habits (e.g. he prefers going to the mountains, or he has already been to Italy). Actually, this information has to be elicited by the seller by interviewing the customer, but most of it might be already known by the employee if he is serving an old customer. In this process, what has to be clearly understood (for properly selecting a holiday package fitting the customer needs) is summerized in the following four words: *where*, *when*, *how*, and *budget*. Indeed, the seller has to understand *where*

the customer desires to go; *when* he can/wishes to leave; how much time he will dedicate to his holiday; which is the preferred transportation means (*how*); and, finally, the available budget. However, the customer does not directly specify all this information, for example, he can ask for a sea holiday in January but he does not specify a precise place, or he can ask for a kind of trip that is unfeasible in a given period (e.g. winter holiday in Italy in August). In this case the seller has to exploit his knowledge of the world for selecting the right destination and a good offer in a huge range of proposals. This is exactly what the IDUM system does when a user starts a new search. Current needs are specified by filling an appropriate search form (see Figure 4), where some of the key information has to be provided (i.e. where and/or when and/or available money and/or how). Note that, the tourism ontology models both the knowledge of the seller (geographic information and preferred places), and the profile of customers (clearly, in the case of new customers the seller has to fill the ontology with profile information, whereas the ontology already contains the information regarding old customers); moreover, the extraction process continuously populates the ontology with new touristic offers. Thus, the system, by running a specifically devised reasoning module, combines the specified information with the one available in the ontology, and shows the holiday packages that best fit the customer needs. For example, suppose that a customer specifies the kind of holiday and the period, then the following (simplified) module creates a selection of holiday packages:

```
module(kindAndPeriod) {
  %detect possible and suggested places
  possiblePlace(P) :- askFor(tripKind:K),
                      PlaceOffer(place:P, kind:K).

  suggestPlace(P) :- possiblePlace(P), askFor(period:D),
                     SuggestedPeriod(place:P1, period:D),
                     not BadPeriod(place:P1, period:D).

  %select possible, alternative and suggestible packages
  possibleOffer(O) :- O:TouristicOffer(destination:P),
                      possiblePlace(P).

  alternativeOffer(O) :- O:TouristicOffer(destination:P),
                         suggestPlace(P).

  suggestOffer(O) :- O:TouristicOffer(destination:P, mean:M),
                     suggestPlace(P), askFor(cust:C),
                     CustomerPrefersMean(cust:C, mean:M).
}
```

The first two rules select: possible places (i.e., the ones that offer the kind of holiday in input); and places to be suggested (because they offer the required kind of holiday in the specified period). Finally, the remaining three rules search in the available holiday packages the ones that: offer an holiday that matches the original input (possible offer); are good alternatives in suggested places (alternativeOffer); or match the customer's preferred means and can be suggested.

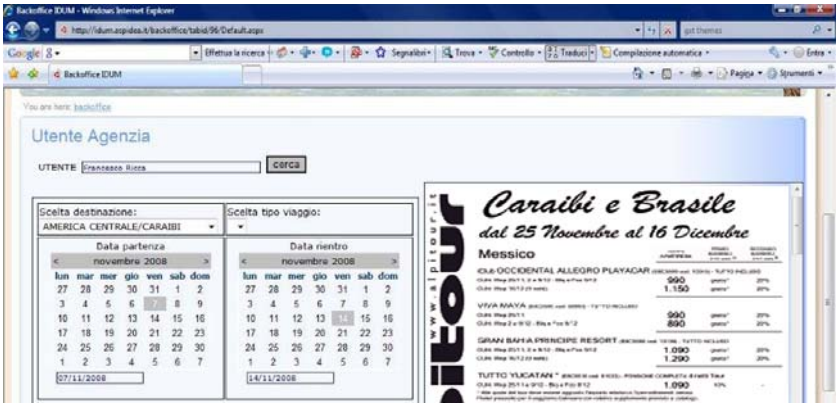

Fig. 4. The IDUM User Interface

The example above reports one of the several reasoning modules that have been devised for implementing the intelligent search. In the development process, we exploited many of the advanced features of the language like negation as failure and aggregates. The declarative nature of ASP allowed us to design effective solutions and to tune rapidly our modules by following the suggestions of the domain experts.

4 System Architecture

The architecture of the IDUM system, which is depicted in Figure 5, is made up of four layers: *Data Layer, Information Layer, Knowledge Layer*, and *Service Layer*. In the Data Layer the input sources are dealt with. In particular, the system is able to store and handle the most common kind of sources: e-mails, plain text, pdf, gif, jpeg, and HTML files. The Information Layer provides ETL (Extraction, Transformation and Loading) functionalities, in particular: in the loading step the documents to be processed are stored in an auxiliary database (that also manages the information about the state of the extraction activities); whereas, in the Transformation step, semi-structured or non-structured documents are manipulated. First the document format is normalized; then, the "bi-dimensional logical representation" is generated (basically, the HiLeX portions are identified); finally, HiLeX descriptors are applied in the Semantic Extraction step and ontology instances are recognized within processed documents. The outcome of this process is a set of concept instances, that are recognized by matching semantic patterns, and stored in the core knowledge base of the system where the tourism ontology resides (Knowledge Layer). Domain ontology and extracted information are handled by exploiting the Persistency manager of the OntoDLV system (see Section 2.1). The Services Layer features the profiling service and the intelligent search (see Section 3.3) which implements the reasoning on the core ontology by evaluating in the OntoDLV system a set of logic programs. The Graphical User Interface (GUI) can access the system features by interacting with a set of web-services.

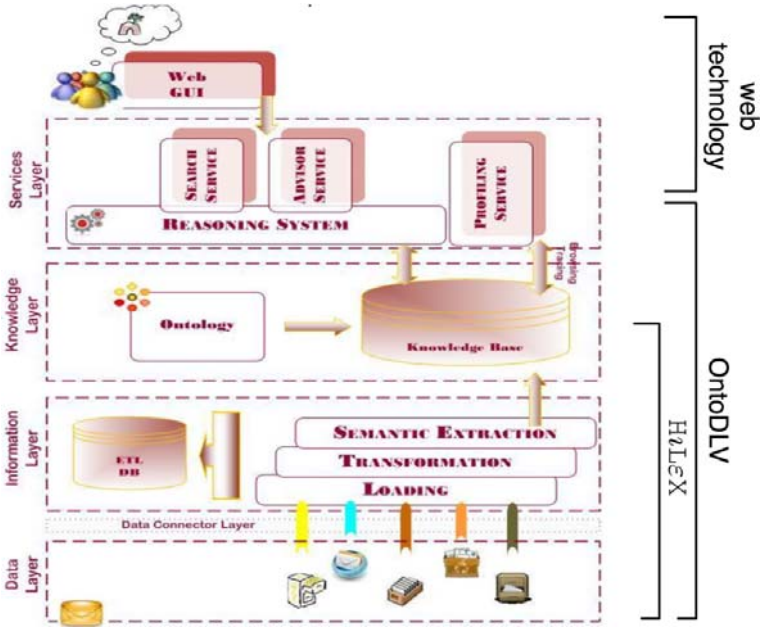

Fig. 5. System Architecture

5 Related Work

The usage of ontologies for developing e-tourism applications was already studied in the literature [13,14,15,16,17], and the potential of the application of semantic technology recognized [18,19].

The architecture of an e-tourism system able to create a touristic package in a dynamic way is presented in [14]. This system permits the customer to specify a set of preferences for a vacation and dynamically access and query a set of information sources to find component such as accommodation, car rental, and leisure activities in real time. It is based on an ontology written in OWL-DL [20]. The ontology exploited in [15,14] encodes the same key concepts of ours, but does not include information about user preferences. Another advantage of the our approach is the possibility of developing ASP programs that reason on the data contained in the ontology for developing complex searches, while there is no accepted solution for combining logic rules and OWL-DL ontologies.

The SPETA system [21], which is based on the ontology of [14], acts as an advisor for tourists. Fundamentally, SPETA follows people who need advising when visiting a new place, and who consequently do not know what is interesting to visit. Here the ontology is enriched with user profile information for determining the common characteristics of the previously visited places and the user behavior. In this way the system recommends attractions which are likely to fit the user expectations. It exploits GPS technology to know user position and it gets user data from previous users history and

also from social networks. Both SPETA and IDUM exploit an ontology for building personalized solutions for the users, but the goal of SPETA is different from that of IDUM. Indeed, the former was conceived for offering assistance and information to the user when they already are on a place, while the goal of IDUM is to assist the users in the selection of an holiday.

The E-Tourism Working Group at DERI [22] is developing e-tourism solutions based on the Semantic Web technology. Their goal is to develop an advanced eTourism Semantic Web portal which will connect the customers and virtual travel agents from anywhere at anytime with any needs and requests. In [23], they present OnTour an information retrieval system that exploits an RDF engine for storing the data regarding accommodation facilities of different types. DERI also developed a content management system: OnTourism [24]. The solution is very similar to IDUM, but it is based on the use of Lixto Software [25] which makes information extraction from web pages. Information about current events is crawled from several web sources and it is rendered in a machine-accessible semantic.

6 Conclusion and Market Perspective

In this paper we have described a successful example of commercial and practical use of logic programming: the e-tourism system IDUM.

The core of IDUM is an ontology modeling the domain of the touristic offers, which is automatically populated by extracting the information contained in the e-mails sent by tour operators; and an intelligent search tool based on answer set programming is able to search the holiday packages that best fits the customer needs.

The system has been developed under project "IDUM: Internet Diventa Umana" (project n. 70 POR Calabria 2000/2006 Mis. 3.16 Azione D Ricerca e Sviluppo nella Imprese Regionali - Modulo B Voucher Tecnologici) funded by the Calabrian Region. The project team involved five organizations: the Department of Mathematics of the University of Calabria (that has ASP as one of the principal research area), the consortium Spin, Exeura srl (a company working on knowledge management), Top Class srl (a travel agency), and ASPIdea (a software farm specialized in the development of web applications). The members exploited their specific knowledge for developing the innovative features of the system. The strong synergy among partners made it possible to push the domain knowledge of the travel agency TopClass in both the ontology and in the reasoning modules. The result is a system that mimics the behavior of a seller of the agency and it is able to search in a huge database of automatically classified offers. IDUM combines the speed of computers with the knowledge of a travel agent for improving the efficiency of the selling process.

The IDUM system was initially conceived for solving the specific problems of a travel agency, and it is currently employed by one of the project partners: Top Class srl. We are working on an enterprise version of the system conceived for offering its advanced services to several travel agencies. The enhancements of IDUM will be developed under another technology-transfer PIA (Pacchetti Integrati di Agevolazione industria, artigianato e servizi) project funded by the Calabrian region. The value of the system was confirmed by the good position obtained by the proposal in the project

evaluation ranking (IDUM occupies the 2nd position over more than 400 competing proposals). Moreover, we received very positive feedbacks from the market, indeed many travel agents are willing to use the system, and the potential of IDUM has been recognized also by the chair of the Italian touring club, which is the most important Italian association of tour operators.

References

1. Gelfond, M., Lifschitz, V.: Classical Negation in Logic Programs and Disjunctive Databases. NGC 9, 365–385 (1991)
2. Eiter, T., Gottlob, G., Mannila, H.: Disjunctive Datalog. ACM TODS 22(3), 364–418 (1997)
3. Leone, N., Pfeifer, G., Faber, W., Eiter, T., Gottlob, G., Perri, S., Scarcello, F.: The DLV System for Knowledge Representation and Reasoning. ACM TOCL 7(3), 499–562 (2006)
4. Ricca, F., Gallucci, L., Schindlauer, R., Dell'Armi, T., Grasso, G., Leone, N.: OntoDLV: an ASP-based system for enterprise ontologies. Journal of Logic and Computation (2009)
5. Ricca, F., Leone, N.: Disjunctive Logic Programming with types and objects: The DLV$^+$ System. Journal of Applied Logics 5(3), 545–573 (2007)
6. Manna, M.: Semantic Information Extraction: Theory and Practice. PhD thesis, Dipartimento di Matematica, Universitá della Calabria, Rende, Cosenza Italia (2008)
7. Ruffolo, M., Manna, M.: HiLeX: A System for Semantic Information Extraction from Web Documents. In: ICEIS (Selected Papers). LNBIP, vol. 3, pp. 194–209. Springer, Heidelberg (2008)
8. Ruffolo, M., Leone, N., Manna, M., Saccà, D., Zavatto, A.: Exploiting ASP for Semantic Information Extraction. In: Proceedings ASP 2005 - Answer Set Programming: Advances in Theory and Implementation, Bath, UK, pp. 248–262 (2005)
9. Gallucci, L., Ricca, F.: Visual Querying and Application Programming Interface for an ASP-based Ontology Language. In: Proc. of (SEA 2007), pp. 56–70 (2007)
10. Gelfond, M., Leone, N.: Logic Programming and Knowledge Representation – the A-Prolog perspective. AI 138(1-2), 3–38 (2002)
11. Minker, J.: Overview of Disjunctive Logic Programming. AMAI 12, 1–24 (1994)
12. Terracina, G., Leone, N., Lio, V., Panetta, C.: Experimenting with recursive queries in database and logic programming systems. TPLP 8, 129–165 (2008)
13. Maedche, A., Staab, S.: Applying semantic web technologies for tourism information systems. In: Proc. of ENTER 2002 (2002)
14. Jorge, C.: Combining the semantic web with dynamic packaging systems. In: Proc. of AIKED 2006, pp. 133–138. World Scientific and Engineering Academy and Society, Singapore (2006)
15. Cardoso, J.: Developing an owl ontology for e-tourism. In: Semantic Web Services, Processes and Applications, pp. 247–282 (2006)
16. Martin, H., Katharina, S., Daniel, B.: Towards the semantic web in e-tourism: can annotation do the trick? In: Proc. of the 14th ECIS 2006 (2006)
17. Martin, H., Katharina, S., Daniel, B.: Towards the semantic web in e-tourism: Lack of semantics or lack of content? In: Proc. of European Semantic Web Conference (ESWC 2006)
18. Dogac, A., Kabak, Y., Laleci, G., Sinir, S., Yildiz, A., Kirbas, S., Gurcan, Y.: Semantically enriched web services for the travel industry. SIGMOD Rec. 33(3), 21–27 (2004)
19. Joe, B., Carole, G.: Tourist: the application of a description logic based semantic hypermedia system for tourism. In: Proc. of HYPERTEXT 1998, pp. 132–141. ACM, New York (1998)
20. Smith, M.K., Welty, C., McGuinness, D.L.: OWL web ontology language guide. W3C Candidate Recommendation (2003), `http://www.w3.org/TR/owl-guide/`

21. Angel, G.C., Javier, C., Ismael, R., Myriam, M., Ricardo, C.P., Miguel, G.B.J.: SPETA: Social pervasive e-tourism advisor. Telemat. Inf. 26(3), 306–315 (2009)
22. DERI: Digital Enterprise Research Institute. Technikerstrae 21a. Innsbruck, A., http://e-tourism.deri.at//
23. Siorpaes, K., Bachlechner, D.: OnTour: Tourism information retrieval based on YARS. In: Proceedings of ESWC 2006 (2006)
24. Ding, Y., Prantner, K., Luger, M., Herzog, C.: OnTourism: Semantic etourism portal. In: Proc. of Scientific Conference of the e-Business Forum Business in Travel, Tourism and Hospitality, Athens, Greece (2008)
25. Lixto, http://www.lixto.at//