An Ontological Framework for Decision Support

Marco Rospocher and Luciano Serafini

FBK-irst, Via Sommarive 18 Povo, I-38123, Trento, Italy
{rospocher,serafini}@fbk.eu

Abstract. In the last few years, ontologies have been successfully exploited by Decision Support Systems (DSSs) to support some phases of the decision-making process. In this paper, we propose to employ an ontological representation for *all* the content both processed and produced by a DSS in answering requests. This semantic representation supports the DSS in the whole decision-making process, and it is capable of encoding (i) the request, (ii) the data relevant for it, and (iii) the conclusions/suggestions/decisions produced by the DSS. The advantages of using an ontology-based representation of the main data structure of a DSS are many: (i) it enables the integration of heterogeneous sources of data available in the web, and to be processed by the DSS, (ii) it allows to track, and to expose in a structured form to additional services (e.g., explanation or case reuse services), all the content processed and produced by the DSS for each request, and (iii) it enables to exploit logical reasoning for some of the inference steps of the DSS decision-making process. The proposed approach have been successfully implemented and exploited in a DSS for personalized environmental information, developed in the context of the PESCaDO EU project.

1 Introduction

Decision support systems (DSSs) are information systems that support users and organizations in *decision-making* activities. DSSs have been applied in several diverse application contexts, to help to take decisions in domains like the medical, legal, computer security, and power consumption management ones.

At an abstract level, we can identify three phases in a decision-making process [1]:

1. the formulation of the decision-making problem;
2. the gathering, storing, and fusion of the data relevant for the given problem;
3. the reasoning on the data to take a decision;

To support the implementation of such process, DSSs usually comprise three main modules [2]: the (i) *dialogue* or *user* module, which supports the interaction of the user with the system, to formulate the problem and receive in output the result of the the DSS computation, the (ii) *data* module, which allows to store the data collected and processed by the DSS, and the (iii) *model* module, which implements the decision support strategy.

Being studied, both theoretically and technically, since the late 1960s, research in DSSs field has taken advantage in the last decade of the achievements and results of Semantic Web technologies. In particular, ontologies have recently been adopted in

H. Takeda et al. (Eds.): JIST 2012, LNCS 7774, pp. 239–254, 2013.

DSSs in various application domains [3,4,5,6,7,8,9,10], exploited for several purposes: to support via reasoning some of the decision support phases [6,3,8], to characterize the data manipulated by the DSS [11,3], and to define the tasks and parameters of the various modules of the system [9]. That is, so far ontologies have been adopted by DSSs to support only parts of the decision-making process, mainly to represent the data and to support their processing for taking decisions.

In this paper we propose to exploit an ontology-based knowledge base as the main (enhanced) data structure of a DSS, where *all* the content and data for a specific decision support request, processed and produced by the system, are stored. In details, our approach consists in designing the ontology underlying the knowledge base, i.e. the *T-Box* in Description Logics (DL) terminology, so that it is capable of formally representing all the details of the three decision-maikng phases described above, i.e., (i) the decision support request submitted by the user to the system, (ii) the data that the system processes for the given request, and (iii) the new content and conclusions produced by the DSS from the available data and in view of the given request, possibly together with some details on how the DSS arrived to those conclusions.

Each single request submitted to the system triggers the instantiation of a new *A-Box* of this ontology. The instantiation incrementally occurs in subsequent steps, coherently with the decision-making process phases. Therefore, at the end of the processing of a request by the DSS, the A-Box associated with the request contains a structured and comprehensive description, a *semantic request story-plan*, of the output produced by the DSS, linked to the data and the request that triggered that output.

The ontological representation of the DSS data that we propose is used to support the DSS activities

- as the main, shared, data structure of the DSS;
- as content exchange format between the different modules of the DSS;
- to keep track of all the intermediate data and results produced by the DSS in the course of solving a problem.

The advantages of using a semantic (ontology based) representation of the main data structure of a DSS are many. First, differently from what happened in the past where DSS were closed systems, in the semantic web era most of the knowledge and data useful to support a decision is available (in heterogeneous formats) in the web. As one of the main objectives of ontologies is to define shared domain models, an ontology-based representation of the knowledge in a DSS facilitates the integration of structured knowledge and data available in the web. Second, in the semantic services era we are now, a DSS can be seen as any other web-service and therefore it can be combined with other semantic services. Keeping a semantically rich track of the entire decision process followed by a DSS in order to reach a conclusion/suggestion/decision, and exposing it by adopting for instance the Linking Open Data[1] principles, enables the combination of the DSS with other complex services, such as explanation services (for which, just information about input-output is not enough) or case reuse/adaptation services (which can adapt the entire reasoning chain done by the DSS to slightly different cases). Finally, the third advantages is the fact that some of the inference steps of the DSS can be

[1] http://linkeddata.org/

performed via state of the art logical reasoning services, as for instance rule reasoners or ontology reasoners.

The proposed approach has been successfully applied in a running personalized environmental DSS, in the context of the PESCaDO EU project, where its features and advantages have been empirically demonstrated.

The paper is organized as follow. First, in Section 2 we review the state of the art on the usages of ontologies in DSSs, while in Section 3 we briefly describe the context of the PESCaDO project. In Section 4, we present the general Decision Support Knowledge Base (DSKB), providing some directions on how to organize its components, together with details on how we actually implemented them for PESCaDO. In Section 5, we describe the steps followed by a DSS in building a semantic request story-plan, reporting the PESCaDO case as an example of such process. In Section 6, we present some examples of how the semantic representation of request story-plans can be exploited for further purposes (natural language reports generation, query-answering services over a repository of archived story-plans). We also remarks (Section 7) some checks to perform when building a DSKB. Finally, in Section 8 we conclude with some final remarks.

2 Ontologies for Decision Support: State of the Art

In the last decade, ontologies have been extensively used in decision support systems, to support tasks in several application domains: clinical management [3], system audit management [4], network security management [5], justice and legal advice [6,7], waste-water management [8], power consumption management [9], electronic issue management [10].

In [10], an ontology driven case-based reasoning system for electronic issue management is described. The ontology is used to formally represent issue management concepts. [9] presents the development of a system that supports decision-making for tasks aimed to reduce power consumption of oil-and-gas production enterprises. The system relies on two ontologies to support communication between the different modules of the system: one describing the objects of the domain, and one describing the tasks supported by the DSS.

[11] presents a framework for ontology-based DSSs in pervasive computing environment. The ontology is exploited to obtain a shared and common understanding of the knowledge domain of the pervasive computing environment, and an explicit conceptualization that describes the semantics of the data managed by the DSS. [3] presents an ontology-based fuzzy DSS to support neuroradiologists in the diagnosis and monitoring of multiple sclerosis. The ontology is used by the system to encode expert knowledge (i.e., qualitative linguistic labels) and the key data processed by the DSS (e.g., possible lesions). [8] shows how to use a static domain ontology for environmental decision support, to capture, understand and describe the knowledge about the physical, chemical and microbiological environment of a waste-water treatment plant, in order to provide reasoning-support for the decision-making phase. [6] presents an OWL ontology to support the inference process in a legal case-based reasoning systems.

As it emerges from all these works, so far ontologies have been successfully exploited to support *some* of the phases of the decision-making process that we described in Section 1, in particular the second phase (e.g., [11,3]), and the third phase (e.g., [6,3,8]). In this paper, we push further the usage of ontologies in DSSs to support all the phases of the decision-making process. In particular, we propose to use the ontology as the main data structure of the DSS, capable of representing *all* the information processed and produced by the DSS, i.e., the decision support request submitted by the user, the data relevant for that request, and the conclusions triggered by the decision support mechanism implemented in the DSS. Furthermore, our approach is not tailored to a specific domain, but it can be exploited in different application contexts.

3 The PESCaDO Use Case

Citizens are increasingly aware of the influence of environmental and meteorological conditions on the quality of their life. One of the consequences of this awareness is the demand for high quality environmental information and decision support that is tailored (i.e., personalized) to one's specific context and background (e.g., health conditions, travel preferences). Personalized environmental information may need to cover a variety of aspects (e.g., meteorology, air quality, pollen) and take into account a number of specific personal attributes of the user (e.g., health, age, allergies), as well as the intended use of the information. For instance, a pollen allergic person, planning to do some outdoor activities, may be interested in being notified whether the pollen situation in the area may trigger some symptoms, or if the temperature is too hot for doing physical exercise, while a city administrator has to be informed whether the current air quality situation requires some actions to be urgently taken.

The goal of the PESCaDO EU project[2] is to develop a multilingual web-service platform providing personalized environmental information and decision support. This platform takes advantage of the fact that nowadays, the Web already hosts a great range of *environmental nodes* (i.e. web-sites, web-services, open data repositories) that offer data on each of the above aspects, such that, in principle, the required basic data are available. The challenge is thus threefold: first, to discover and orchestrate these environmental nodes; second, to process the obtained data in accordance with the decision support needs of the user; and, third, to communicate the gained information in the user's preferred mode.

For a general overview of the running PESCaDO system[3], and the type of information produced, check the demonstration video[4], or directly play with the on-line demonstrator[5]. Shortly, users submit a decision support request to the system (e.g. "*I want to do some hiking in Nuuksio Park tomorrow: is there any health issues for me?*"), specifying in full details the type of request, the type of activity (if any) they want to perform, their profile, the geographic area and the time period to be covered. Then, the system (i) determines the data relevant for the request, (ii) retrieves the data from

[2] `http://www.pescado-project.eu`
[3] A more comprehensive description of the system workflow can be found in [12].
[4] `http://www.youtube.com/watch?v=c1Ym7ys3HCg`
[5] Accessible from the project web-site, or directly here `http://193.145.50.130/`

environmental nodes providing them[6], (iii) processes these data providing conclusions (e.g., warnings, recommendations) according to the needs of the users, and, finally, (iv) generates reports (e.g., text, tables, graphics) to be communicated to the user.

4 The Decision Support Knowledge Base

We propose a reference architecture for designing ontology-based knowledge bases for decision support systems, called Decision Support Knowledge Base (DSKB). It aims at guiding the development of an OWL [13] ontology capable of representing in a connected and comprehensive way all the content relevant for a given decision support request. In details, in our approach each decision support request is associated with an A-Box (i.e., a set of individuals and assertions on them) instantiating the T-Box part of the DSKB (see Figure 1[7]). The DSKB T-Box, to which we refer to as *Decision Support Ontology* (DSO), comprises three main components, namely Problem, Data, and Conclusions. These three components are connected by relations between the corresponding elements. As shown by Figure 1, these relations are[8]: *hasData* and *hasConclusion*, which relates a problem description with the data relevant for it and

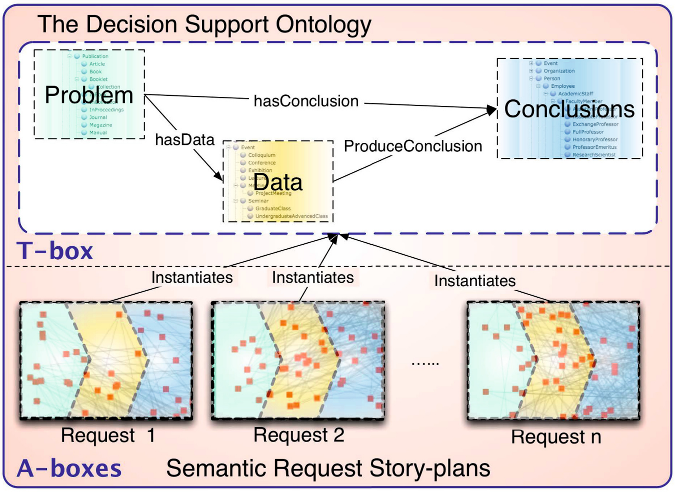

Fig. 1. The Decision Support Knowledge Base

[6] More precisely, environmental data are hourly distilled from environmental web-sites and stored in a dedicated repository, so when a decision request is submitted to the system, for efficiency reasons, this database is actually queried for data (instead of directly querying web-sites in real-rime).

[7] The three parts of each request A-Box correspond to the components of the T-Box.

[8] The set of object properties here presented is not exhaustive, and can be further extended depending also on the application context.

the conclusions provided for it by the DSS, and *ProduceConclusion* which connects the data with the conclusions they trigger. As we will remark in Section 4.4, these object properties allow to relate the instances of the different components of the DSO, to assembly a connected semantic request story-plan. Furthermore, these properties are particularly useful for explanation purposes: e.g., the *ProduceConclusion* allows to keep track of what data triggered a certain conclusion of the DSS.

Next, we describe each part of the DSKB, reporting the details of its implementation in the context of the PESCaDO DSS[9].

4.1 Problem Component

The purpose of this component of the DSKB is to formally describe all the aspects of decision support problems that the user can submit to the system. In its simplest form, this component could consist in a taxonomy of the request types supported by the system, enriched by the additional input parameters that are needed by the DSS to provide adequate decision support to the users (e.g., date/time and location of the request). In more advanced situations, a problem description may include also aspects of the users profile (e.g., age, preferences, diseases), or other additional problem features that may affect the decision support provided by the DSS. This component may also be used by the DSS dialogue module to guide (and constrain) users in composing a request.

PESCaDO Problem component The Problem component defined for the PESCaDO system comprises three interrelated parts (see Figure 2): *Request*, *Activity*, and *User*. *Request* describes a taxonomy of request types supported by the system (e.g., “Is there any health issue for me?”, “Do environmental conditions require to take some administrative actions?”). *User* enables to describe the profile of the user involved in the request. Examples of the aspects modelled in this component are the user typology (e.g., end-user or administrative user), the age of the user, the gender, diseases or allergies the user may suffer from. Finally, *Activity* describes the activities the user may want to undertake, and that may affect the decision support provided by the system. For instance, different factors are considered by the DSS if the user decides to do some physical outdoor activity rather than travelling with public transportation.

These three sub-components are interrelated by object properties (e.g., a request has to have a user profile associated with it, and may involve an activity the user wants to undertake) and subclass axioms which constrain the possible combination allowed. The PESCaDO user interface module has been developed to dynamically read this constraints from the ontology, to guide the users in formulating their decision support problems. For instance, the subclass restriction “hasRequestUser only AdministrativeUser” on class “CheckAirQualityLimits” states that a request for checking the air quality limits can be submitted only by administrative users. Similarly, the restriction “hasRequestActivity some (AttendingOpenAirEvent or PhysicalOutdoorActivity or Travelling)” on

[9] The DSO of the PESCaDO DSKB consists of 210 classes, 99 object properties, 42 datatype properties, and 641 individuals, and it is available at: `http://www.pescado-project.eu/ontology.php`

Fig. 2. Excerpt of the PESCaDO Problem component

class "AnyHealthIssue" enables to propose to the users only some activities, those of type "AttendingOpenAirEvent" or "PhysicalOutdoorActivity" or "Travelling", forcing them to select one of those. Further parameters are also defined to allow the specification of all the necessary details to compose a complete problem description: for instance, the time period and geographical region considered in the request.

4.2 Data Component

The purpose of this component of the DSKB is to formally describe the data accessed and manipulated by the DSS. For instance, in the case of an environmental DSS, the Data component could describe physical phenomena observations like temperature, humidity, or wind speed, while in the case of a financial DSS, it may represent stock market rates or currency exchanges.

To some extent, this component play the role of the *domain ontology* of the DSS application. Differently from the other two components of the DSO which are more application-oriented, an ontology to be used as Data component may be already available in the web, and thus reused in the DSKB. By adopting an ontological representation of the data processed by the DSS, we favour the integration of (structured) data provided by heterogeneous sources, like web-sites or nodes of the Linking Open Data cloud. For instance, this approach enables to easily exploit for decision making purposes the data exposed by smart city initiatives, like the SmartSantander project[10].

All the aspects of the data that may affect the conclusions taken by the DSS should be described in this component: e.g., validity, provenance, trust, uncertainty. For instance, the Data component may incorporate standardization efforts like the Open Provenance Model [14].

[10] `http://www.smartsantander.eu/`

```
EnvironmentalData
    EnvironmentalData SubClassOf hasFromDateTime some dateTime
    EnvironmentalData SubClassOf hasEnvironmentalDataNature exactly 1 EnvironmentalDataNature
    EnvironmentalData SubClassOf hasEnvironmentalDataEnvironmentalDataType exactly 1 EnvironmentalDataType
    EnvironmentalData SubClassOf hasToDateTime some dateTime
EnvironmentalNode
    EnvironmentalNode SubClassOf hasEnvironmentalNodeLocation max 1 Location
    EnvironmentalNode SubClassOf hasEnvironmentalNodeEnvironmentalNodeAreaType max 1 EnvironmentalNodeAreaType
    EnvironmentalNode SubClassOf hasEnvironmentalNodeName exactly 1 string
    EnvironmentalNode SubClassOf hasEnvironmentalNodeForm exactly 1 EnvironmentalNodeForm
    EnvironmentalNode SubClassOf hasEnvironmentalNodeEnvironmentalNodeType max 1 EnvironmentalNodeType
    EnvironmentalNode SubClassOf hasEnvironmentalNodeConfidenceValue max 1 double
    EnvironmentalNode SubClassOf hasEnvironmentalNodeEnvironmentalData only EnvironmentalData
    EnvironmentalNode SubClassOf
        hasEnvironmentalNodeEnvironmentalNodeSourceOfEmissionType max 1 EnvironmentalNodeSourceOfEmissionType
    EnvironmentalNode SubClassOf hasEnvironmentalNodeURL max 1 anyURI
    EnvironmentalNode SubClassOf hasEnvironmentalNodeEnvironmentalNodeLandUseType max 1 EnvironmentalNodeLandUseType
```

Fig. 3. Excerpt of the PESCaDO Data component

PESCaDO Data component The Data component in PESCaDO describes the environmental data used by the system to provide decision support: e.g., meteorological data (e.g., temperature, wind speed), pollen data, and air quality data (e.g., NO2, PM10, air quality index). Environmental related data like traffic and road conditions are also represented. All the necessary details to comprehensively describe observed, forecast, or historical data are described, including values (both quantitative and qualitative values are supported), the time period covered by the data, and the type of the data (e.g., instantaneous, average, minimum, maximum). Concerning the values, the mapping between qualitative values and quantitative ones is also encoded in the ontology: for instance, a *moderate* quantity of birch pollen in the air correspond to a concentration between 10 and 100 grains per meter cube of air. Detailed information on the environmental node providing the data is also representable, like its type (e.g., measurement station, web-site, web-service), geographical location, and confidence value.

The development of this component of the PESCaDO DSKB has involved (i) the reuse of (part of) some already available environmental domain ontologies (e.g., SWEET[11]), (ii) the application of techniques for automatic ontology extension [15] (e.g., to define the pollen sub-domain), and (iii) the contribution of environmental domain experts. An excerpt of the Data component of the PESCaDO DSKB, specifically the characterization of environmental data and nodes, is illustrated in Figure 3.

The ontological representation of the data processed by the PESCaDO DSS is used to integrate for decision-making purposes the input data coming from heterogeneous sources, and obtained with different techniques, like by querying environmental web-services or by distilling data from text and images offered by environmental web-sites.

4.3 Conclusions Component

The purpose of this component of the DSKB is to formally describe the output produced by the DSS by processing the problem description and the data available. Examples of the content to be produced are warnings/suggestions/instructions, as well as the results of further processing of the data (e.g., data aggregations, data analysis results). Details on the confidence of the system about this content may also be represented, for instance

[11] `http://sweet.jpl.nasa.gov/ontology/`

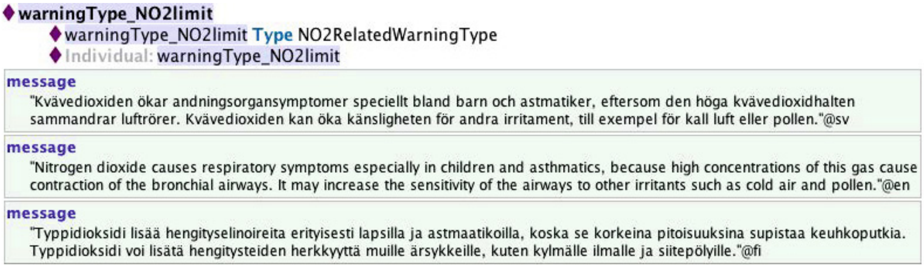

Fig. 4. Excerpt of the PESCaDO Conclusions component

supporting the possibility to assign a weight. Furthermore, if needed, this component may also allow to represent the feedback left by the users about their degree of satisfaction of the content produced by the DSS for the submitted request. We also recall that the "ProduceConclusion" object property links the conclusions produced by the DSS to the data that triggered those conclusions, an information that may be exploited for explanation purposes.

PESCaDO Conclusions component The Conclusions component in PESCaDO allows to describe conclusions like warnings, recommendations, and suggestions that may be triggered by environmental conditions, or exceedances of air pollutants limit values that may be detected from the data. An example of warning type encoded in the PESCaDO DSKB is described in Figure 4, together with the associated warning message to be reported to the users (available in all the three languages supported in PESCaDO, English, Finnish, and Swedish). A warning issued by the PESCaDO system for a given decision support request, is represented as a new instance in the A-Box associated with the request, having an object property asserting the type of warning, and a datatype property asserting the relevance (a decimal value in [0..1]) of the warning for the current problem request.

4.4 Semantic Request Story-Plans

The three components of the DSO provide a schema to represent the main aspects of a decision support request. Therefore, for any given decision support request submitted to the system, the actual content about these three aspects of the given request can be formalized as a set of individuals, and assertions on them, instantiating the T-Box part of the DSKB. This results in a connected set of triples, what we called a semantic request story-plan. A semantic request story-plan is an RDF graph covering all the aspects of any decision support request: its formulation, the data relevant for it, and the conclusions generated by the DSS from the data.

An A-Box of the PESCaDO DSKB Figure 5 shows an excerpt of an A-Box instantiating the proposed schema. The three blocks in Figure 5 represents the three components of the T-Box of which the individuals, subjects of the assertions, instantiate some

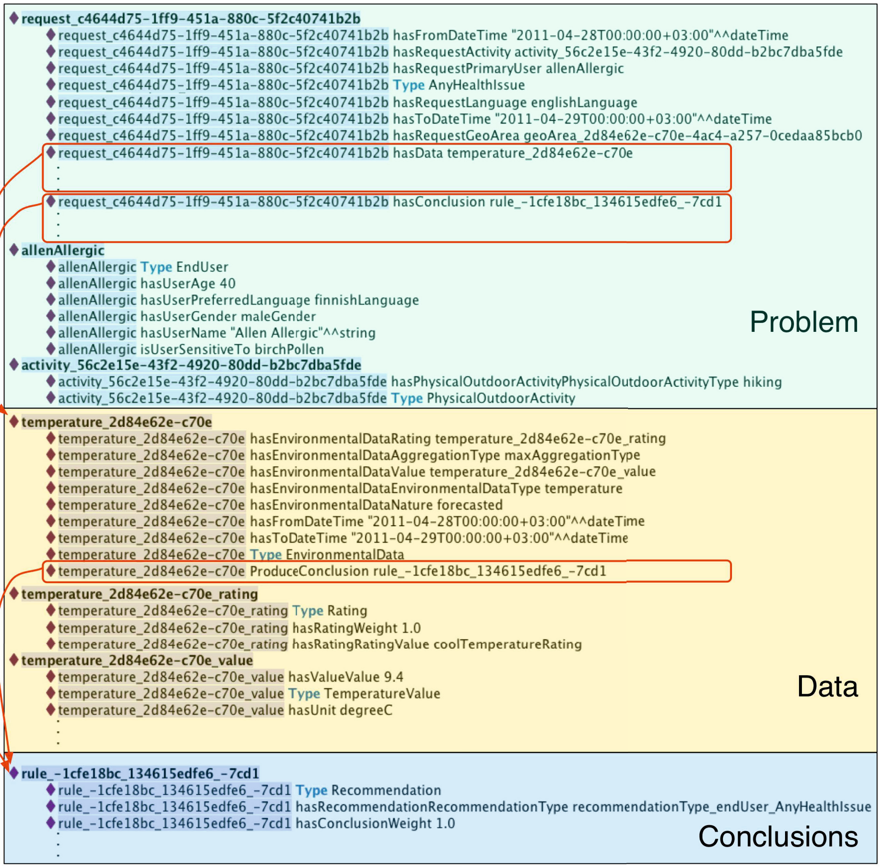

Fig. 5. An A-Box of the PESCaDO DSKB

classes. Note the connections between the individuals of different components, highlighted by the red boxes and corresponding arrows.

Next, we show how a semantic request story-plan is composed by the DSS while processing a request.

5 Incrementally Building Semantic Request Story-Plans

Coherently with the main phases of a decision-making process (see Section 1), semantic request story-plans are incrementally built by DSSs in three consequent phases.

Phase1: Instantiation of the problem. In the first phase, the problem part of the DSO is instantiated. This occurs when the user submits the request to the DSS via the dialogue module. That is, a module of the system processes the input selections and parameters provided by the user, and generates the instances and assertions characterizing the user decision support request. A consistency check of the input instances with the schema defined by the DSO can be performed via reasoning, to verify that the user request is compliant with the problem supported by the DSS.

Phase2: Instantiation of the data. In the second phase, the actual data which will be used by the DSS to provide decision support, and generate the final conclusions, are instantiated in the A-Box, and connected to the instances describing the problem being processed. First, the DSS determines which data are relevant for the user decision support request submitted. For this purpose, different strategies and techniques can be exploited. In PESCaDO, we encoded in the DSO some mappings between the three parts (request, user, activity) of the Problem component, and the types of environmental data formalized in the Data component, to represent that certain environmental data are relevant for some problem aspects. These mappings, defined together with the domain experts involved in the project, are formalized as OWL hasValue restrictions on the classes of the Problem component. For instance, a restriction of the form "hasRelevantAspect **value** Rain" on the class characterizing the users sensitive to some pollen, states that data about precipitation should be retrieved and taken into consideration when providing decision support for this typology of user. By modelling the mapping this way, the environmental data types for which to retrieve data about, can be automatically determined via DL-reasoning, simply checking the new assertions inferred by the reasoner to the request, user, and activity individuals forming the current user decision request.

Once the data to be used are determined, the module of the system in charge of retrieving these data from the data sources can instantiate them in the DSKB according to the schema defined in the Data component. The connection of the ontology individuals formalizing the data with the individuals formalizing the request under processing is also instantiated (see the "hasData" assertions - first red box - Figure 5).

Phase3: Instantiation of the conclusions. In the third phase, the conclusions triggered by the data according to the user decision support request are instantiated in the A-Box. The way conclusions are computed depends on the techniques for decision support implemented in the DSS. For instance, in PESCaDO we implemented a module responsible for computing conclusions which combines some complementary techniques, based on DL-reasoning and rule-based reasoning. More in details, a two-layers reasoning infrastructure is currently in place: the first layer exploits the HermiT reasoner [16] for the OWL DL reasoning services. The second layer is stacked on top of the previous layer, and implements the Jena [17] RETE rule engine, which performs the rule-based reasoning computation. Next we report an example of rule for triggering the introduction of a recommendation to pollen-sensitive users in case of abundant pollen levels:

```
[ruleAbundantPollen:
     (?request rdf:type pescadoProblem:AnyHealthIssue)
     (?request pescadoProblem:hasUser ?user)
     (?user pescadoProblem:isSensitiveTo ?pollen)
     (?pollen rdf:type pescadoData:PollenDataType)
     (?request pescado:hasGeoArea ?geoArea)
     (?request pescado:hasData ?data)
     (?data pescadoData:hasEnvironmentalDataType ?pollen)
```

```
    (?data pescadoData:hasAggregationType pescadoData:max)
    (?data pescadoData:hasRating ?rating)
    (?rating pescadoData:hasRatingValue pescadoData:abundantPollen)
    makeTemp(?rec)
  ->
    (?rec rdf:type pescadoConclusions:Recommendation)
    (?rec pescadoConclusions:hasRecommendationType
          pescadoConclusions:recommendation_abundantPollen)
    (?rec pescadoConclusions:hasWeight '1.0'^^xsd:double)
    (?request pescado:hasConclusion ?rec)
    (?data pescado:ProduceConclusion ?rec)]
```

Once the conclusions are instantiated in the DSKB according to the schema defined in the Conclusions component, they are also connected to the individuals formalizing the request under processing (see the "hasConclusion" assertions - second red box - Figure 5), and to the data triggering them (see the "ProduceConclusion" assertions - third red box - Figure 5).

Each semantic request story-plan is maintained by a DSS at least for the lifetime of the processing of the request by the system. Then, the DSS can dispose of the story-plan, or it can archive it in a dedicated cases repository for other purposes (see Section 6). Note that, especially for web-based DSS like PESCaDO, where simultaneous requests may be submitted by users to the system, several semantic request story-plans may be there at the same time in the DSKB. To manage them in parallel, the DSS can adopt an *ontology pooling* mechanism [18]: multiple in-memory ontologies (aka *pools*) are available in the system, and each decision support request submitted to the DSS is assigned exclusively to one of these pools. This solution, adopted in the PESCaDO DSS, allows to keep the size of the ontology used in each pool relatively small (on average the PESCaDO system is working with A-Boxes containing approximatively 20,000 triples), allowing to efficiently exploit the ontology also for some DL-reasoning tasks, like the ones we previously described in this section.

6 Exploitation of Semantic Request Story-Plans

In this section, we present a couple of examples where the semantic representation of a request story-plan can be further exploited to offer additional enhanced services.

6.1 Natural Language Generation of DSS Reports

At the end of a DSS computation, the A-box associated with a decision support request contains a complete "semantic" snapshot of all the information processed and produced by the DSS for the given request: it contains a complete description of the user request, the data relevant for the request and that were used for the decision support computation, and the conclusions and inferred content produced by the DSS together with the information on what triggered those conclusion. All this information can be used to automatically generate a text, summarizing and explaining in natural language the most relevant information to be reported to the user. In the context of an environmental DSS, like in PESCaDO, this automatically generated text, which may complement information provided by the system in graphical or tabular form, is especially appreciated by laymen, or even media corporations which may directly spread it through their communication channels.

In PESCaDO, an approach for multilingual personalized information generation from dynamically instantiated ontologies is adopted [19]. Two modules are involved in the information generation: the text planning module and the linguistic generation module. In particular, the text planning module consists of a content selection and a discourse structuring phase, both performed on a dynamically instantiated ontology (e.g., the T-Box + an A-Box of our DSKB) extended by an additional ontology module capable of representing content selection schemas and elementary discourse units. The output of the text planning module is thus an instantiated ontology enriched with information on the content selected, and the way the text should be organized. This constitutes the input of the linguistic generation module, which produces the text in the three languages supported by the system. Next we report an excerpt of the kind of output produced by the PESCaDO system by exploiting the semantic request story-plan.

> Situation in the selected area between 08h00 and 20h00 of 07/05/2012. The ozone warning threshold value ($240g/m^3$) was exceeded between 13h00 and 14h00 ($247g/m^3$), the ozone information threshold value ($180g/m^3$) between 12h00 and 13h00 ($208g/m^3$) and between 14h00 and 15h00 ($202g/m^3$). The minimum temperature was 2°C and the maximum temperature 17°C. The wind was weak (S). There is no data available for carbon monoxide, rain and humidity.
> **Ozone warning**: ozone irritates eyes and the mucous membranes of nose and throat. It may also exacerbate allergy symptoms caused by pollen. Persons with respiratory diseases may experience increased coughing and shortness of breath and their functional capacity may weaken. Sensitive groups, like children, asthmatics of all ages and elderly persons suffering from coronary heart disease or chronic obstructive pulmonary disease, may experience symptoms. [...]

6.2 Semantic Archive of Request Story-Plans

The semantic request story-plans produced by the DSS could be archived in a semantic repository (e.g., a triple store like Sesame [20] or Virtuoso [21]), whose schema is defined by the DSO of the DSKB. This allows to build an incrementally growing archive of all the decision support requests handled by the DSS, together with the data used to process each request and the conclusions generated, as well as some feedback of the user on the decision support provided by the system.

This semantic archive of request story-plans can be exploited for several purposes, enabled by the possibility to semantically access/query its content. For instance, the archive could be used to fine-tune the decision support strategies implemented in the DSS by querying and inspecting the requests not positively rated by the users. Similarly, in the case of DSSs implementing case-based reasoning strategies, the positively rated requests could be used to strengthen the selection of cases used for taking decisions.

Furthermore, thanks to the precise semantic provided by the DSO, this archive could be exposed to the world, adopting the Linked Open Data philosophy. Therefore, the content of the archive could be further exploited by other applications or web-services, like for instance a case reuse/adaptation service which can adapt to slightly different cases the decision-making process done by the DSS, the main phases of which are tracked in each semantic request story-plan.

Relevant statistics could also be produced by semantically querying the archive. For instance, in the context of PESCaDO, one may be interested in checking how frequent is the occurrence of warnings, triggered by environmental conditions, reported to sensitive users, or which geographic areas are more frequently part of the decision support requests submitted by users, or which type of requests are more frequently submitted by each of the various typologies of users supported by the system.

7 On Engineering a DSKB

As any other engineering artefact, the DSKB has to undergo some checks to verify its adequacy for the DSS it supports. Next, we remark some of the important aspects to be taken into consideration.

In addition to checking the formal consistency of the DSO part of the DSKB with state of the art OWL reasoners (e.g., Hermit), and to verifying its correct instantiation with the usage in the DSS, it is crucial to assess the completeness and appropriateness of the content represented in the DSO with respect to the requirements of the application context in which the ontology is used. This requires to check that:

- the types of decision support requests to be supported by the system are formally representable in the Problem component;
- the data relevant for the decision support to be provided are characterized in the Data component;
- the types of decision support conclusions and explanations to be generated by the DSS are formalized in the Conclusions component;

In PESCaDO, the adherence to these guidelines of the three components of the PESCaDO DSO has been evaluated separately, and with different strategies. The Problem component has been evaluated by checking that it is able to represent all and only the possible types of decision support requests that the PESCaDO DSS is supposed to handle. This task was eased by the availability of a detailed description of the typology of these requests in the PESCaDO Use Cases Specification document (Deliverable D8.5[12]). The check showed that all the types of requests defined in the use cases can be represented by the Problem component of the PESCaDO DSO. For the other two components (Data and Conclusions) of the PESCaDO DSO, we asked some environmental expert users (4 people) to judge the appropriateness and completeness of their content, by filling some closed questionnaires. Instead of asking them to directly inspect the content of the ontology, we presented a set of representative decision support requests, the type of environmental data that the system determined to be relevant (and, thus, that the ontology is able to represent) for each of these requests, as well as the type of conclusions that the system is able to provide for them. For the Data component, the results showed an appropriateness of 94% and a completeness of 92% of the content, while for the Conclusions component we obtained an appropriateness of 90% and a completeness of 87%. Therefore, despite some minor adjustments to be made, the PESCaDO DSO proved to be adequate with respect to the proposed guidelines.

[12] http://www.pescado-project.eu/Pages/Pdfs-pages/D8.5.pdf

8 Conclusions

In this paper we proposed to employ an ontology-based knowledge base as the main data structure in DSSs. In our approach, to each decision support request submitted to the DSS corresponds a semantic request story-plan in the knowledge base, which describes in a structured way (i) the request itself, (ii) the data relevant for the request, and (iii) the conclusions/suggestions/decisions generated by the system by processing the data. We described the possible usages and advantages offered by the proposed approach, demonstrating them in a concrete implementation for an environmental DSS, developed in the context of the PESCaDO EU project. In details, we showed that a semantic representation of the content processed and produced by the PESCaDO DSS enables (i) to integrate heterogeneous sources of data available in the web (e.g., web sites, web services), (ii) to track, and to expose in a structured form to additional services (e.g., a natural language report generation service), all the content processed and produced by the DSS for each request, and (iii) to exploit logical reasoning for several of the inference steps of the DSS decision-making process.

Acknowledgements. The work described in this paper has been partially funded by the European Commission under the contract number FP7-248594, PESCaDO (Personalized Environmental Service Configuration and Delivery Orchestration) project. The authors would like to thank all the partners involved in the project, and Chiara Di Francescomarino for her support in designing the evaluation of the PESCaDO DSO.

References

1. Laskey, K.B.: Decision Making and Decision Support (2006), `http://ite.gmu.edu/~klaskey/SYST542/DSS_Unit1.pdf`
2. Marakas, G.M.: Decision support systems in the twenty-first century. Prentice-Hall, Inc., Upper Saddle River (1999)
3. Esposito, M., De Pietro, G.: An ontology-based fuzzy decision support system for multiple sclerosis. Eng. Appl. Artif. Intell. 24(8), 1340–1354 (2011)
4. Ishizu, S., Gehrmann, A., Minegishi, J., Nagai, Y.: Ontology-driven decision support systems for management system audit. In: Proc. of the 52nd Annual Meeting of the ISSS (2008)
5. Choraś, M., Kozik, R., Flizikowski, A., Renk, R., Hołubowicz, W.: Ontology-based decision support for security management in heterogeneous networks. In: Huang, D.-S., Jo, K.-H., Lee, H.-H., Kang, H.-J., Bevilacqua, V. (eds.) ICIC 2009. LNCS (LNAI), vol. 5755, pp. 920–927. Springer, Heidelberg (2009)
6. Wyner, A.: An ontology in owl for legal case-based reasoning. Artif. Intell. Law 16(4), 361–387 (2008)
7. Casanovas, P., Casellas, N., Vallbé, J.-J.: An ontology-based decision support system for judges. In: Proceedings of the 2009 Conference on Law, Ontologies and the Semantic Web: Channelling the Legal Information Flood, pp. 165–175. IOS Press (2009)
8. Ceccaroni, L., Cortés, U., Sànchez-Marrè, M.: OntoWEDSS: augmenting environmental decision-support systems with ontologies. Environmental Modelling & Software 19(9), 785–797 (2004)
9. Zagorulko, Y., Zagorulko, G.: Ontology-based approach to development of the decision support system for oil-and-gas production enterprise. In: Proceedings of the 2010 Conference on New Trends in Software Methodologies, Tools and Techniques: Proceedings of the 9th SoMeT 2010, pp. 457–466. IOS Press, Amsterdam (2010)

10. Pangjitt, T., Sunetnanta, T.: A model of ontology driven case-based reasoning for electronic issue management systems. In: International Joint Conference on Computer Science and Software Engineering (2011)
11. Saremi, A., Esmaeili, M., Habibi, J., Ghaffari, A.: O2dss: A framework for ontology-based decision support systems in pervasive computing environment. In: Proceedings of the 2008 Second Asia International Conference on Modelling & Simulation (AMS 2008), pp. 41–45. IEEE Computer Society, Washington, DC (2008)
12. Wanner, L., Vrochidis, S., Tonelli, S., Moßgraber, J., Bosch, H., Karppinen, A., Myllynen, M., Rospocher, M., Bouayad-Agha, N., Bügel, U., Casamayor, G., Ertl, T., Kompatsiaris, I., Koskentalo, T., Mille, S., Moumtzidou, A., Pianta, E., Saggion, H., Serafini, L., Tarvainen, V.: Building an environmental information system for personalized content delivery. In: Hřebíček, J., Schimak, G., Denzer, R. (eds.) ISESS 2011. IFIP AICT, vol. 359, pp. 169–176. Springer, Heidelberg (2011)
13. Smith, M.K., Welty, C., McGuinness, D.L.: Owl web ontology language guide. W3C Recommendation (2004)
14. Moreau, L., Clifford, B., Freire, J., Futrelle, J., Gil, Y., Groth, P.T., Kwasnikowska, N., Miles, S., Missier, P., Myers, J., Plale, B., Simmhan, Y., Stephan, E.G., Van den Bussche, J.: The open provenance model core specification (v1.1). Future Generation Comp. Syst. 27(6), 743–756 (2011)
15. Tonelli, S., Rospocher, M., Pianta, E., Serafini, L.: Boosting collaborative ontology building with key-concept extraction. In: Proceedings of the Fifth IEEE International Conference on Semantic Computing, Stanford, CA, USA (2011)
16. Shearer, R., Motik, B., Horrocks, I.: HermiT: A Highly-Efficient OWL Reasoner. In: Ruttenberg, A., Sattler, U., Dolbear, C. (eds.) Proc. of the 5th Int. Workshop on OWL: Experiences and Directions (OWLED 2008 EU), Karlsruhe, October 26-27 (2008)
17. hp: Jena - A Semantic Web Framework for Java (2002), `http://jena.sourceforge.net/index.html`
18. Moßgraber, J., Rospocher, M.: Ontology management in a service-oriented architecture. In: 11th International Workshop on Web Semantics and Information Processing, WebS 2012 (2012)
19. Bouayad-Agha, N., Casamayor, G., Mille, S., Rospocher, M., Saggion, H., Serafini, L., Wanner, L.: From ontology to NL: Generation of multilingual user-oriented environmental reports. In: Bouma, G., Ittoo, A., Métais, E., Wortmann, H. (eds.) NLDB 2012. LNCS, vol. 7337, pp. 216–221. Springer, Heidelberg (2012)
20. Broekstra, J., Kampman, A., van Harmelen, F.: Sesame: A generic architecture for storing and querying rdf and rdf schema. In: Horrocks, I., Hendler, J. (eds.) ISWC 2002. LNCS, vol. 2342, pp. 54–68. Springer, Heidelberg (2002)
21. Erling, O., Mikhailov, I.: Virtuoso: Rdf support in a native rdbms. In: Semantic Web Information Management, pp. 501–519. Springer (2009)