

Ontology-Based Decision Support System for Enterprise Interoperability

Mahdi Zouch^{1,2}, Wided Guedria^{1(✉)}, and Riadh Ben Halima²

¹ ITIS, Luxembourg Institute of Science and Technology (LIST),
5 avenue des Hauts-Fourneaux, 4362 Esch-sur-Alzette, Luxembourg
{mahdi.zouch,wided.guedria}@list.lu

² National Engineering School of Sfax ENIS,
University of Sfax, 3038 Sfax, Tunisia
{mahdi.zouch,riadh.benhalima}@enis.tn

Abstract. With the increased globalization of the economy, the competitiveness has become ubiquitous and enterprises need to be reactive and to collaborate with different stakeholders to survive in its environment. Within this context, the choice of suitable collaborators and partners with whom collaboration may happen without problems is a key factor of success. In this paper, an ontology-based approach to support interoperability and help enterprises to solve interoperability problems before they occur is proposed.

Keywords: Ontology · Decision support system
Enterprise interoperability · Interoperability problem · Knowledge

1 Introduction

There are a lot of success stories about enterprise collaboration. However, it should not be denied that collaboration is sometimes synonymously with problems and that there are also stories of collaborations that have been a failure. By having a close look, one's may clearly see that solving collaboration problems or preventing them from appearing by choosing the suitable partners, can help enterprises avoid failed business stories. When such a choice is available, decision makers need to take into account different criteria including the impact of a given collaboration in terms of potential problems. Making a decision is a tough task that has enormous impacts on the enterprise future and more likely the success or the failure of a business. According to Forbes¹, nine startups out of ten fails in making “good” decisions. Making “bad” decisions could lead to serious consequences, especially when these decisions are related to collaboration. This explains the growth of collaboration tool software. In fact, seven CEOs out of ten planned to increase their spending on collaboration tools in 2003 while the rate of growth of such software reached 38% in 2014. This shows the need for this kind of tools.

Since there is no decision support tool that prevents potential interoperability problems before occurring and proposes solutions to the identified problems to help

¹ American business magazine.

organizations planning their interoperability from an AS-IS situation to a TO-BE one, this work focuses on implementing such system.

The remainder of this paper is structured as follows: Sect. 2 presents some related work, while Sect. 3 presents the proposed approach and the architecture of the system. Section 4 shows a real use case scenario using the proposed system and finally Sect. 5 concludes the paper and highlights some future work.

2 Related Work

In this section, we introduce the ontology of enterprise interoperability which constitutes the basis of this work, and give an overview of some existing decision support systems.

2.1 The Ontology of Enterprise Interoperability (OoEI)

Interoperability is ubiquitous but not easy to understand due to its numerous definitions and interpretations [1]. Ford et al. point out that according to their survey, thirty-four definitions of interoperability were proposed since 1977 [2]. Some definitions can be found in [3–7]. Within this context the Ontology of Interoperability (OoI) [8] was proposed to give a common understanding about interoperability. It considers the Interoperability as a problem to solve: “*An interoperability problem appears when two or more incompatible systems are put in relation. Interoperability per se is the paradigm where an interoperability problem occurs*” [9].

Based on that, the Ontology of Enterprise Interoperability (OoEI) [1] was proposed as an extension of the OoI in the enterprise field. The OoEI is composed of three main parts which are:

- The Systemic model where the enterprises are instantiated and defined in the ontology. Definitions are found in [1, 9, 10].
- The Decisional model where the analysis is performed to identify problems and related solutions. Definitions are found in [10].
- The Knowledge model where the expert’s knowledge is stored [10].

2.2 Existing Decision Support Systems

A decision support system (DSS) is a “*set of related computer programs and data required to assist with analysis and decision making within an organization*”. Many decision support systems have been proposed in the literature. Some of them adopt an ontology-based approach. The main relevant works that we have found are:

- In the health field, the PESCaDO project exploits environmental data, weather forecasts as well as user’s health profile and activities to determine if the desired activity could cause potential health issues due to the weather [11]. Another relevant research work in the same field proposes a diagnosis and treatment recommendation system for diabetes [12].

- In [13], the authors propose an ontology based decision support system to help nontechnical consumers to select the “right” domestic solar hot water system (i.e. that is tailored to their needs), with updated information on installation costs, components and interrelationships [13].
- A research work has also been conducted in the field of self-driving and autonomous vehicles. It focuses on making fast driving decisions at cross-roads by representing the data collected from sensors in a machine understandable format to help vehicles understanding traffic situations and making decisions [14].

The research works cited above have a number of common components which are: The Ontology, the reasoner, the inference rules, the knowledge base and the query engine. The ontology is defined as an “explicit specification of a conceptualization” [15]. It allows to define standardized concept, notions and relationships that can be used by every individual involved in a specific domain. There are several types of ontologies, the most common types are upper-ontologies and domain ontologies [16]. But other works suggest other types such as task and application ontologies [17]. Ontology is composed of concepts, relations and instances [18]. The reasoner is an important element in any ontology-based system. It is a tell/ask interface [16] that uses inference rules to perform the reasoning. Inference rules used to derive new knowledge from existing knowledge in the ontology [16]. Knowledge bases are the combination of an ontology with associated instances [18]. The components cited above will be considered to establish a proof of concept of a DSS for enterprise interoperability.

3 The Ontology-Based Decision Support System for Enterprise Interoperability

In order to implement the decision support system, some additions and changes have been done to the OoEI. We start by redefining the knowledge model. In fact, in the literature, the knowledge base definition is often linked to TBOX and ABOX pair only [13]. So, it is a set of instances around the concepts. Since the knowledge model in the ontology does not interfere with the reasoning process, we can split the OoEI into two parts. The first part includes the Systemic model and the Decisional Model where the reasoning happens. The second part is the knowledge graph that contains the knowledge model and other concepts from the decisional model. This is done for performance reasons because the system loads the ontology for each request. Thus, it seems reasonable to put only the part where the reasoning will take place in order to minimize the loading time.

3.1 The Knowledge Graph

The knowledge graph (Fig. 1) is composed of two parts: (a) The knowledge base which contains the problems and solutions as depicted in the left part of the Fig. 1, (b) The enterprise and user’s database where all users and enterprises information are stored as shown in the right part of Fig. 1.

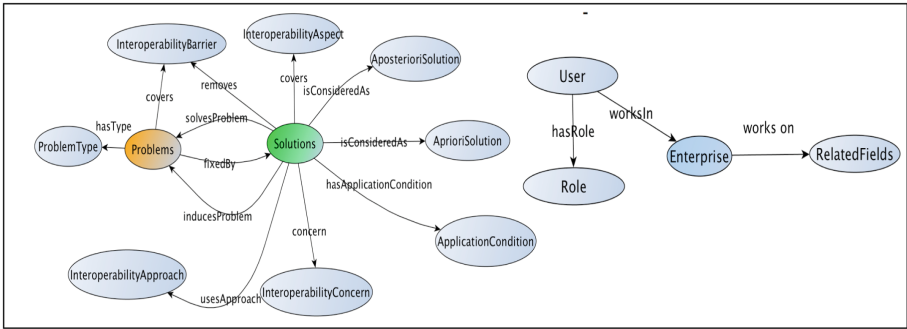


Fig. 1. The knowledge graph and enterprise nodes

3.2 The Systemic and Decisional Model

In this section, we give an overview of the systemic and decisional model, as defined in [1] as well as the added concepts and properties. For the sake of space, some concepts are omitted and some other concepts are removed because they can be represented as data properties. Figure 2 shows the main concepts of the ontology. Added concepts and properties are presented with green color while the orange ellipses represent concepts of the systemic model. The blue ellipses are concepts of the decisional model and the yellow ones are concepts linking the decisional model with the systemic model.

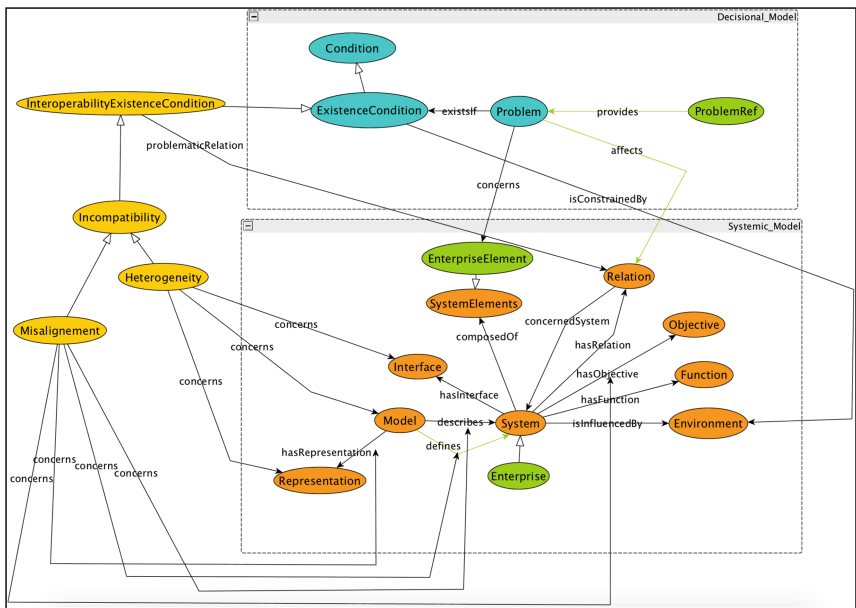


Fig. 2. Decisional and the systemic model (Color figure online)

The *Enterprise* concept as well as the *EnterpriseElement* concept are represented as subclasses of the concept *System*. We have added the concept *ProblemRef* that contains all the references of the problems in the knowledge graph. It is related to the *Problem* concept with a new relation “*provides*”. The object property “*affects*” between the *Problem* and the *Relation* is also added to help us find the enterprises involved in a particular problem. We have also added new Data properties. For example, all instances of the *Representation* concept have the following data properties:

- *Speaks*: xsd:string, it is used to add the official language of an enterprise.
- *isClient*: xsd:Boolean, it is used to indicate if the enterprise is the client or is a part of the networked enterprise.
- *useOfficeTool*: xsd:string, it is used to indicate what office suite the enterprise is using.

3.3 Inference Rules

Inference rules are crucial in a decision support system. They are interpreted and used by the reasoner to achieve the problem detection process. In order to detect interoperability problems, three inference rules have been defined for almost each problem as depicted in Fig. 3. These rules are incremental and dependent:

- First rule is used to detect the existence of an incompatibility between enterprises.
- Second rule is used to locate the source of incompatibility by finding the concerned enterprises.
- Third rule is mainly used to find the problem; it summarizes all the previous rules.

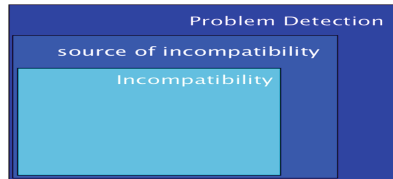


Fig. 3. Incremental rules in the problem detection process

It is worth noting that these rules are dynamic. This means that they change every time we add new constraints. One example to mention is the internal language barrier problem: Suppose that we have an interoperability problem due to the language barrier between two enterprises *s1* and *s2* that speak different languages *a* and *b*, respectively. The corresponding rules are written as follows:

- Incompatibility detection rule:
 $Enterprise(?s1) \wedge Enterprise(?s2) \wedge Model(?m1) \wedge Model(?m2) \wedge Representation(?sr1) \wedge Representation(?sr2) \wedge describes(?m1, ?s1) \wedge describes(?m2, ?s2) \wedge hasRepresentation(?m1, ?sr1) \wedge hasRepresentation(?m2, ?sr2) \wedge speaks(?sr1, ?a) \wedge speaks(?sr2, ?b) \wedge swrlb:notEqual(?a, ?b) \rightarrow Incompatibility(heterogeneity) \wedge concerns(heterogeneity, ?sr1) \wedge concerns(heterogeneity, ?sr2)$

- Locating the source of incompatibility:
 $Entreprise(?s1) \wedge Entreprise(?s2) \wedge Model(?m1) \wedge Model(?m2) \wedge Relation(?r) \wedge Representation(?sr1) \wedge Representation(?sr2) \wedge composedOf(?s1, ?r) \wedge composedOf(?s2, ?r) \wedge concerns(heterogeneity, ?sr1) \wedge concerns(heterogeneity, ?sr2) \wedge describes(?m1, ?s1) \wedge describes(?m2, ?s2) \wedge hasRepresentation(?m1, ?sr1) \wedge hasRepresentation(?m2, ?sr2) \wedge speaks(?sr1, ?a) \wedge speaks(?sr2, ?b) \wedge swrlb:notEqual(?a, ?b) \wedge Incompatibility(heterogeneity) \wedge concerns(heterogeneity, ?sr1) \wedge concerns(heterogeneity, ?sr2) \rightarrow problematicRelation(existenceCondition, ?r)$
- Problem detection rule:
 $Entreprise(?s1) \wedge Entreprise(?s2) \wedge Model(?m1) \wedge Model(?m2) \wedge Relation(?r) \wedge Representation(?sr1) \wedge Representation(?sr2) \wedge composedOf(?s1, ?r) \wedge composedOf(?s2, ?r) \wedge concerns(heterogeneity, ?sr1) \wedge concerns(heterogeneity, ?sr2) \wedge describes(?m1, ?s1) \wedge describes(?m2, ?s2) \wedge hasRepresentation(?m1, ?sr1) \wedge hasRepresentation(?m2, ?sr2) \wedge problematicRelation(existenceCondition, ?r) \wedge speaks(?sr1, ?a) \wedge speaks(?sr2, ?b) \wedge swrlb:notEqual(?a, ?b) \wedge isClient(?sr1, false) \wedge isClient(?sr2, false) \rightarrow Problem(LanguageBarrierInternal) \wedge affects(LanguageBarrierInternal, ?r)$

3.4 System Architecture

The proposed system is designed for enterprises. It is mainly dedicated to decision makers who need to have a clear view about interoperability problems and related possible solutions regarding a potential collaboration or to improve a current one. The needed information is found in the knowledge base that has to be updated regularly by the knowledge expert. The system has also an administrator to configure it and to be contacted in case of technical problems. It was developed as a web application using Spring boot framework, apache tomcat as the web server as depicted in Fig. 4. Screenshots can be found at this link: <https://goo.gl/QCpnDX>.

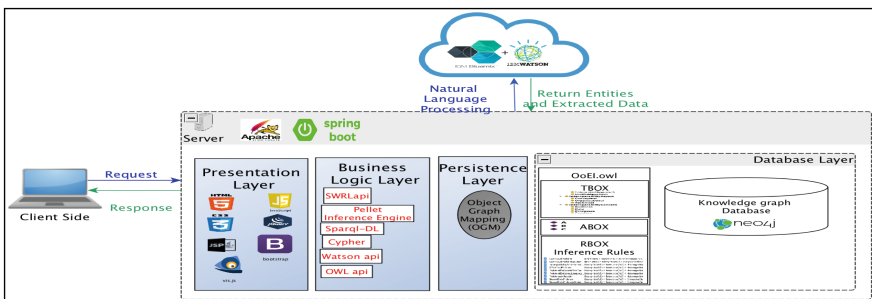


Fig. 4. System architecture

Three principal parts can be distinguished: (1) **The client side:** It represents the operations performed by the client using a web browser. In this case, the client can be the administrator, the knowledge expert or the user. (2) **The server side:** It receives all the requests from the client to process them and send the result back to the client. In the

server, we distinguish four layers: (a) Presentation layer: it is the layer responsible for the user interface. It translates tasks and results to a user-understandable form using jsp, html, css, javascript, jquery, bootstrap and visjs. (b) Business logic layer: it is the layer containing all the application functionalities. This layer uses a number of APIs and technologies namely OWLapi, SWRLapi, Pellet inference engine, watson api, Cypher and sparql-DL, etc. (c) Persistence layer: also known as Data Access Layer, it makes the access to the database easier for the Business Logic Layer by providing an API that exposes methods for managing the database. This layer uses OGM (Object graph mapping) which is a fast object-graph mapping library for Neo4j database that uses Cypher query language, it is like JPA and uses annotations on simple POJO domain objects. (d) Database layer: it contains the Knowledge base stored in Neo4j and the Ontology file. This is where information is stored and retrieved. (3) **The cloud services:** They are used for natural language processing to extract relevant entities from the enterprise description text. To achieve this, two cloud services are used: (a) IBM Watson Knowledge Studio: This service allows to build a dedicated and personalized machine learning model for the enterprise domain. (b) IBM Bluemix: This platform allows to run, build, deploy and manage applications on the cloud.

3.5 The Decision Support System Workflow

The proposed decision support system works as follows: (1) Enterprise data are instantiated as ABOX in the ontology file, the data comes either from database or a text description. (2) The reasoner uses the SWRL rules to infer new information and detect the possible problem by finding the problem reference. (3) Once the problem references are detected, the system launches the query module which uses sparql and Cypher to retrieve the result from the Knowledge graph. Figure 5 gives an overview of the system workflow.

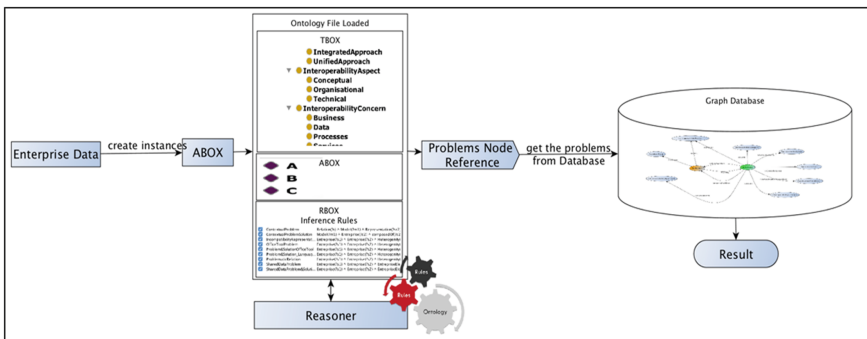


Fig. 5. Overview of the decision support workflow

4 Use Case Description

In order to establish the proof of concept of our decision support system, we have defined our system based on interviews with real enterprises located in Luxembourg. For privacy reasons, we will omit the names of the enterprises and use fake names instead of the real ones.

CL is the client in this scenario. (It offers to print companies a variety of innovative software services for managing and orchestrating printers.) It is willing to settle up a new business in Luxembourg. **CL** contacted **Innovation Company (Innov)** (an innovation and strategy consulting agency and a member of **Innov-hub** and the **Marketing Group (MG)**) for its consulting services. After checking its profile and needs in terms of innovation, **Innov** oriented **CL** to **Innov-hub** network (a soft-landing platform and accelerator for national and international start-ups.). For the purpose of this study, no details about **Innov-hub** network are given and it is considered as a black box entity that has as input a startup that needs help and output a growing startup. After doing the project with **Innov-hub**, our client wanted to promote the project, **Innov** oriented **CL** to **Offline Marketing Company (OMC)** (a full-service communication consulting company and member of the **MG**).

The first problem encountered by **OMC** and **CL** is the language Barrier. In fact, **CL** has German as official language while **OMC** has French as official language.

It is worth saying that this kind of problem is frequent in Luxembourg where there exist three official languages: Luxembourgish, French and German.

To overcome this problem, **OMC** which is a part of **MG**, needed a mediator, in our case **Digital Marketing (DG)** (a digital marketing agency) which is also a part of the **MG**. **DG** plays the role of the mediator between the two enterprises. A meeting was organized between **OMC** (i.e. the service provider), **DG** (i.e. the mediator) and **CL** (i.e. the client) to gather the client's needs.

After removing the language barriers, **OMC** is still having problems in understanding the client's needs due to its lack of expertise regarding innovation projects. To overcome this problem, **OMC** contacted **Innov** which is also member of the **MG**. Experienced in innovation, **Innov** played the role of the mediator and helped **OMC** understanding the client's needs. **OMC** then proposed an offer. After some negotiations, **CL** accepted the service offer and signed the contract. As soon as the contract signed, **OMC** defined the main tasks and related deadlines and assign tasks to the concerned actors.

Taking the fact that in the contract, the client has only three propositions, if none of them was satisfying the client should pay for the next propositions. Due to the misunderstanding between the **CL** and **OMC** and to the tight budget of the client, **CL** was not satisfied and ended the contract.

For the sake of space and clarity, we represent an extract of the instantiated systemic model for the language barrier problem as depicted in Fig. 6. The orange and green ellipses represent the concepts in the ontology while the pink ellipses are the instances. The blue color represents the data properties with the corresponding data.

The data properties can be assimilated to attributes in a java class. They carry all the needed information. These information as well as inference rules, are processed via the

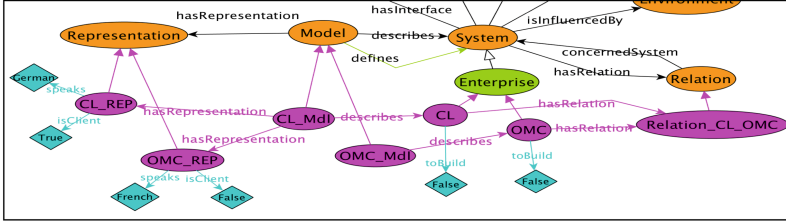


Fig. 6. Extract of the instantiated systemic model (Color figure online)

reasoner to discover the existence of a language barrier. The DSS workflow is described in Sect. 3. Figure 6 shows the knowledge that is needed to detect the external language barrier. CL speaks German and is the client, that’s why we see the *speaks* property and *isClient* data property in the *CL_REP* instance. The same thing for OMC that speaks French. The applied rules are the ones presented in Sect. 3. For our case, we use the following rule:

$$\begin{aligned}
 & \text{Entreprise}(?s1) \wedge \text{Entreprise}(?s2) \wedge \text{Model}(?m1) \wedge \text{Model}(?m2) \wedge \text{Relation}(?r) \wedge \\
 & \text{Representation}(?sr1) \wedge \text{Representation}(?sr2) \wedge \text{composedOf}(?s1, ?r) \wedge \text{composedOf}(?s2, \\
 & ?r) \wedge \text{concerns}(\text{heterogeneity}, ?sr1) \wedge \text{concerns}(\text{heterogeneity}, ?sr2) \wedge \text{describes}(?m1, \\
 & ?s1) \wedge \text{describes}(?m2, ?s2) \wedge \text{hasRepresentation}(?m1, ?sr1) \wedge \text{hasRepresentation}(?m2, \\
 & ?sr2) \wedge \text{problematicRelation}(\text{existenceCondition}, ?r) \wedge \text{speaks}(?sr1, ?a) \wedge \text{speaks}(?sr2, \\
 & ?b) \wedge \text{swrlb:notEqual}(?a, ?b) \wedge \text{isClient}(?sr1, \text{true}) \wedge \text{isClient}(?sr2, \text{false}) \rightarrow \text{Problem} \\
 & (\text{LanguageBarrierExternall}) \wedge \text{affects}(\text{LanguageBarrierExternal}, ?r).
 \end{aligned}$$

5 Conclusion

In this paper, we have proposed a decision support system for enterprise interoperability. In order to build the system, we have used a real use case scenario. We have also made changes and adaptations to the existing Ontology of Enterprise Interoperability which was developed in a previous research work. We then designed and built the prototype. The developed system will constitute the basis for further improvements which can be categorized into short-term, mid-term and long-term goals. The mid-term goals are more oriented to the quality of the use case. In order to improve the prototype, there is a need to define a much more complex use case with a large amount of constraints. Long-term consist on developing new machine learning based mechanisms to find the relevant knowledge and feed the knowledge graph without the intervention of knowledge experts in the feeding process of the knowledge base.

Acknowledgements. This work has been conducted in the context of the PLATINE project (PLanning Transformation Interoperability in Networked Enterprises), financed by the national fund of research of the Grand Duchy of Luxembourg (FNR), under the grant C14/IS/8329172.

References

1. Guedria, W.: A contribution to enterprise interoperability maturity assessment (Doctoral dissertation, Bordeaux 1) (2012)
2. Ford, T., et al.: Measuring system interoperability: an i-score improvement. In: Proceedings of the 6th Annual Conference on Systems Engineering Research, Los Angeles, CA, 4–5 April (2008)
3. Institute of Electrical and Electronics Engineers. IEEE standard computer dictionary: a compilation of IEEE standard computer glossaries (1990)
4. Department of Defense, Washington. Joint Pub 1-02, DoD Dictionary of Military and Associated Terms, 2001 (INTEROP, 2003) Interoperability Research for Networked Enterprises Applications and Software, Network of Excellence, Annex 1 – Description of Work (2003)
5. IDEAS project deliverables (wp1–wp7). Technical report (2003)
6. Chen, D., Dassisti, M., Elvestor, B.: Di.3: enterprise interoperability framework and knowledge corpus - final report. Technical report, INTEROP Network of Excellence, IST-Contract no. IST-508 011 (2007)
7. Morris, E., Levine, L., Meyers, C., Place, P., Plakosh, D.: System of systems interoperability (SOSI). Final report. Software Engineering Institute, Carnegie Mellon University, Pittsburgh (2004)
8. Rosener, V., Naudet, Y., Latour, T.: A model proposal of the interoperability problem. In: CAISE05 workshops, EMOI 2005, vol. 2, pp. 395–400 (2005)
9. Naudet, Y., Latour, T., Hausmann, K., Abels, S., Hahn, A., Johannesson, P.: Describing interoperability: the OoI ontology. In: EMOI-INTEROP (2006)
10. Naudet, Y., Latour, T., Guedria, W., Chen, D.: Towards a systemic formalisation of interoperability. *Comput. Ind.* **61**(2), 176–185 (2010)
11. Rospocher, M., Serafini, L.: Ontology-centric decision support. In: Proceedings of the 2012 International Conference on Semantic Technologies Meet Recommender Systems & Big Data, vol. 919, pp. 61–72. CEUR-WS.org (2012)
12. Alharbi, R.F., Berri, J., El-Masri, S.: Ontology based clinical decision support system for diabetes diagnostic. In: Science and Information Conference (SAI), pp. 597–602. IEEE (2015)
13. Kontopoulos, E., Martinopoulos, G., Lazarou, D., Bassiliades, N.: An ontology-based decision support tool for optimizing domestic solar hot water system selection. *J. Cleaner Production* **112**, 4636–4646 (2016)
14. Zhao, L., Ichise, R., Sasaki, Y., Liu, Z., Yoshikawa, T.: Fast decision making using ontology-based knowledge base. In: 2016 IEEE Intelligent Vehicles Symposium (IV), pp. 173–178. IEEE (2016)
15. Gruber, T.R.: A translation approach to portable ontology specifications. *Knowl. Acquis.* **5**(2), 199–220 (1993)
16. Gomez-Perez, A., Fernández-López, M., Corcho, O.: *Ontological Engineering: with Examples from the Areas of Knowledge Management, e-Commerce and the Semantic Web*. Springer Science & Business Media (2006)
17. Guarino, N.: Formal ontology and information systems. In: Proceedings of FOIS, vol. 98, no. 1998, pp. 81–97 (1998)
18. Stevens, R., Goble, C.A., Bechhofer, S.: Ontology-based knowledge representation for bioinformatics. *Brief. Bioinform.* **1**, 398–414 (2000)