Journal of Engineering and Applied Sciences 14 (Special Issue 6): 9332-9339, 2019
ISSN: 1816-949X

Combining Logical and Probabilistic Reasoning for Ontology-Based Application

[1, 2]Foni Agus Setiawan, [1]Eko K. Budiardjo, [1]Achmad N. Hidayanto and [3]Meyliana Meyliana
[1]Faculty of Computer Science, Universitas Indonesia, Depok, Indonesia
[2]Research Center for Limnology, Indonesian Institute of Sciences, Jakarta, Indonesia
[3]Bina Nusantara University, Central Jakarta, Indonesia

Abstract: Ontology is widely used in semantic web to model the knowledge in the form of classes or individuals and relationships among them. Ontology languages such as Web Ontology Language (OWL) and Resource Description Framework (RDF) are built based on discrete logic which cannot deal with probabilistic knowledge about a domain. Various approaches have been made to represent uncertainty in ontology, such as Bayes OWL, Multi-Entity Bayesian Networks (MEBN) and Probabilistic OWL (PR-OWL). These research emphasize on how to represent uncertainty and reasoning in ontology based on Bayesian network approach. On the other hand, sometimes we need to solve a problem with the logic and probabilistic approach simultaneously as well. This study discusses the importance of combining both approaches. The research also proposes an approach or framework to perform both logic and probabilistic reasoning that can be implemented into an ontology-based application. A prototype has been developed as an experimental in order to simulate the work of this framework by having two cases: investor problem and social CRM for higher education.

Key words: Ontology-based application, logical reasoning, probabilistic reasoning, description logic, Bayesian network, used

INTRODUCTION

Ontologies are the foundation of the semantic web and are widely used in machine knowledge representation. They facilitate the definition and conceptual characterization of classes, individuals and their relationships within a domain (Santos and Jurmain, 2011). Ontology is an explicit representation of concepts in a domain consisting of resources with their characteristics and their relationships. Ontologies are often used when there is an application that requires a knowledge representation that is able to maintain all relevant information about a domain.

However, ontology languages in the semantic web such as Web Ontology Language (OWL) and Resource Description Framework (RDF) are based on crisp logic and thus cannot handle incomplete, partial, fuzzy or probabilistic knowledge about a domain. For example in ontology reasoning, one might want to know not only whether there is a relationship between A and B but how close A-B or someone might want to know how similar A-B although they are unrelated.

Reasoning in ontology currently supported in OWL by using Description Logic (DL) described in OWL-DL. DL is used to describe and reason about the relevant concepts of an application domain. The fundamental modelling concept of a DL is the axiom, a logical statement relating roles and/or concepts (Grau *et al*., 2008). Many Description Logics (Dls) are decidable fragments of First-order Logic (FoL) (Baader *et al*., 2008) and are usually fragments of two-variable logic or guarded logic. Reasoning mechanisms using logic like this we refer to as logical reasoning.

On the other hand, the need of reasoning is not for logical only (true or false, 1 or 0) but also probabilistic (degree of certainty about the possibilities that will happen). The degree of certainty is usually expressed in a range of values between 0 and 1. For example, the statements) The probability of a patient is exposed to smallpox is 0.53). The patient is exposed to smallpox after a known occurrence of red spots on the skin has the probability of 0.78. The last statement has a prior knowledge which is “a known occurrence of red spots on the skin”.

Bayesian Network (BN) is one of the model to compute probabilities from connected nodes (as in ontology). BN depicted in the form of Directed Acyclic Graph (DAG) consisting of nodes and relationships between nodes (arcs). Nodes represent facts, classes or objects while directed arcs describe unidirectional relationship between the parent nodes and child nodes. The relationship can describe how much the parent

Corresponding Author: Foni Agus Setiawan, Faculty of Computer Science, Universitas Indonesia, Depok, Indonesia influences to the child (influential relations) or how strong the parent incident causes the child (causal relations). For the above example, a Bayesian network could represent the probabilistic relationships between diseases and symptoms. Given symptoms, the network can be used to compute the probabilities of the presence of various diseases. Reasoning in probabilistic environment like this we refer to as probabilistic reasoning.

DL is good for relational reasoning about classes or objects in an ontology domain while BN is good for reasoning with uncertainty. Combining both reasoning models into an ontology-based application will not only give the expressive power because the application will have a complete reasoner engine but also enrich the ontology itself. This is because both of the reasoning models will complement facts to each other. Recently, research in Bayesian networks has shown that logical relationships between variables in a network can be used to enhance the efficiency of computations done using the network by eliminating impossible combinations of the values of variables during processing Gelfond.

The challenge in making application that combine the two models of reasoning is the application must be able to read both data format (OWL and BN) should be able to complete the information on the OWL reasoner machine to BN and vice versa as well as to be able to do reasoning with complete facts from both of sources.

Literature review: Several approaches have been proposed on handling uncertainty in ontologies by creating framework or module that combines ontology with BN such as BayesOWL, MEBN, PR-OWL and Pronto.

Bayes OWL is a framework that was first introduced by Ding *et al*. (2005). Main function of this framework is to translate an ontology into BN complete with its Conditional Probability Table (CPT). The framework architecture contains three main components: OWL Parser, BN Structure Constructor and CPT Constructor.

At first, ontology and uncertainty knowledge entered into the system. OWL Parser consists of two sub-components, namely: Taxonomy Parser (T-Parser) and Probability Parser (P-Parser). T-Parser will translate ontology into taxonomy then BN structure constructor will construct DAG of BN from this taxonomy. P-Parser process the uncertainty of knowledge and collaborates with CPT Constructor construct CPT for the translated BN. Control nodes are created during the construction of BN to facilitate modelling relations among class nodes that are specified by OWL logical operators and there is a converging connection from each concept nodes involved in this logical relation to its specific control node. Multi-Entity Bayesian Network (MEBN) is a knowledge representation formalism that combines the power of first-order logic with uncertainty (Laskey, 2008). MEBN provides syntax, a set of model construction and inference processes and semantics that together provide a means of defining probability distributions over unbounded and possibly infinite numbers of interrelated hypothesis (Cesar and Kathryn, 2005). MEBN theories use directed graphs to specify joint probability distributions for a collection of related random variables.

Implementation of MEBN in ontology is constructed into a framework known as PR-OWL. PR-OWL was first introduced by Laskey (2008) as an upper ontology enhancement from MEBN. PR-OWL was developed as an extension enabling OWL ontologies to represent complex Bayesian probabilistic models based on different probabilistic technologies (e.g., PRMs, Bns, etc.) (Costa and Laskey, 2006).

Reification of relationships in PR-OWL is a necessary thing because of the fact that properties in OWL are binary relations (i.e., link two individuals or an individual and a value) while many of the relations in a probabilistic model include more than one individual (i.e., N-ary relations) (Costa and Laskey, 2006). These differences lead to differences in a way of representing relationships, its handling and the process of reasoning. The use of reification for representing N-ary relations on the Semantic Web is covered by a working draft from the W3C's Semantic Web Best Practices Working Group.

Another approach that tries to combine elements of probability into ontology is Pronto. Pronto is not a framework but only an additional extension tacked on top of Pellet (Sirin *et al*., 2007) to support the probabilistic reasoning based on Nilsson probabilistic logic (Nilsson, 1986). Pellet itself is OWL-DL Reasoner which works according to the rules in the Description Logic (DL) SHOIN.

Pronto added capability of non-monotonic probabilistic reasoning based on DL SHIQ standard formalisation referred to as P-SHIQ. The main reasoning task in P-SHIQ is entailing new conditional constraints, both terminological and assertional from probabilistic knowledge bases. Pronto is the first reasoner for a Nilsson-style probabilistic DL that can effectively deal with non-propositional classical knowledge base.

The technical implementation to combine ontology with BN has also been initiated. Fenz *et al*. (2009) provide clues to perform the translation of an ontology into BN Fenz *et al*. (2009) described an algorithm for transforming/morphing an ontology into Object-oriented Bayesian Network (OOBN). BN representation in XML format to standardize the data exchange has been trying pioneered by XMLBIF Working Group by formulating a Bayesian XML Interchange Format (XMLBIF). XMLBIF has been implemented on Java Bayes and Genie modelling tools.

MATERIALS AND METHODS

Ontology-based application: The use of ontologies in software applications is growing in line with the emergence of many new standardization with respect to ontology. The standardization was made both with respect to its representation as RDF, RDF(s) and OWL and reasoning like DAML+OIL, SPARQL and OWL-DL. Tools to help building applications using ontology knowledge base even more and more popping up, most of them are built using Java.

In order for an application software can use the knowledge base ontology then the application must be able to read the OWL data formats and capable of reasoning to the knowledge base using SPARQL or DL query. Illustration of a basic ontology-based application can be seen in Fig. 1.

The knowledge base consists of a collection of RDF/OWL documents regarding a particular domain which are spread across both intranet and internet. Reasoner is assigned to conduct reasoning against the knowledge base for every query sent by the application in the format of SPARQL or DL query.

Standardized knowledge base has been published a lot on the internet to fulfil the needs for interoperable data to be used by ontology-based applications known as Linked Data. Among them are DBPedia that contains the "info box" data automatically collected from Wikipedia, FOAF (Friend of A Friend) ontology for specifying information about people and their social and business relationships and FreeBase is a community driven web portal that allows people to enter facts as structured data. It is possible to query freebase and get results as RDF.

Similar with the knowledge base many institutions have developed reasoners that support standardized SPARQL-DL query format. Among them are Hermit and Pellets that has been integrated into Stardog as discussed above.

Proposed approach: Current main approaches (BayesOWL and MEBN/PR-OWL) focused on morphing/transforming Ontology to BN then perform the reasoning in resulted BN. Only one approach (Pronto) tried to adopt probabilistic reasoning directly in ontology using Nilsson's probabilistic logic instead of Bayesian network, therefore still cannot support/adopt prior knowledge in reasoning.

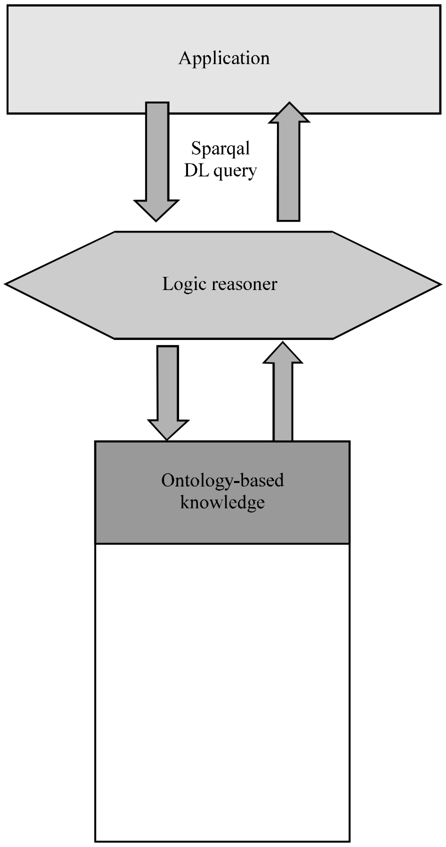

Fig. 1: Basic architecture of ontology-based application

We tried to propose a framework for application software that use ontologies as its knowledge base. We call it ByNowLife, an acronym for Bayesian network and OWL integration framework as shown in Fig. 2.

This framework consists of three main parts, application, reasoner and the knowledge base. The application can be either an independent software or a software agent that is written in common programming languages such as C/C++, Java, PHP or Python.

The reasoner consists of two components, logic reasoner is assigned to conduct logic-based reasoning based on SROIQ specifications in OWL 2 DL and probabilistic reasoner: is assigned to conduct probabilistic reasoning against the knowledge base in the form of BN, OOBN or Dynamic Bayesian Network (DBN).

The knowledge base contains knowledge in two forms: Ontologies in OWL/RDF format and Bayesian networks in XML of Decision Systems Laboratory (XDSL) format. In the knowledge base, there is a component named Morpher which is assigned to conduct data transformation from standardized ontologies (OWL/RDF) to BN XDSL and vice versa. The Morpher also serves to enrich the ontology with probability values of the nodes

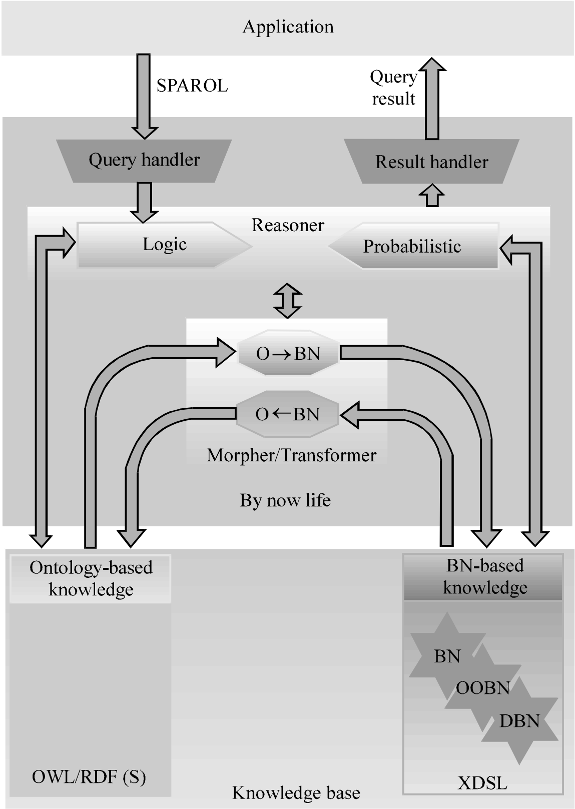

Fig. 2: The proposed framework for ontology-based application

and links in the BN. The probability values of nodes in the BN will be translated into axioms in the ontology and links into relations. This framework allows an application to query the knowledge base in SPARQL Query format for logical reasoning and special property "hasProbValue" for probabilistic reasoning.

RESULTS AND DISCUSSION

Experiment and validation cases: We have implemented the framework into a prototype of application software. The application is developed using C# language. It utilizes OWLAPI for .Net Libraries as ontology reader and writer, Pellet as logical reasoner and SMILE as its probabilistic reasoner. We build ByNowLife library package that contains three main modules module that handle query and its execution result, module that control/coordinate logical and probabilistic reasoning and Morpher/Transformer module that translate OWL/RDF into XDSL and vice ersa. Sample of cases resolved using this technique are as.

Case 1; Investor problem: An investor has the shares in the following stocks: Mandiri, KPC and Telkomsel.

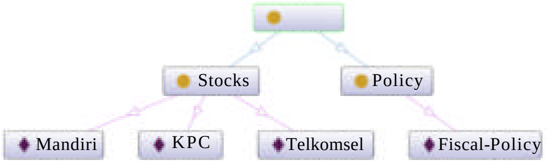

Fig. 3: Investor the structure of classes and individuals in OWL document

Mandiri is a bank, KPC is a mining company and Telkomsel is a telecommunication company. Each of these stocks has a price property. On the other hand, there are government policies that affect the fluctuation of the stock price. For example, the latest condition of the case represented in the two documents OWL/RDF and XDSL as follows (Fig. 3).

In BN document, we have rules that the Fiscal-Policy influence stock prices and the fluctuation of the stock price will affect the investor income as can be seen in the structure.

Suppose that we want to answer the query: display a stock name which has price • 5000 and has the highest impact to profit among others. In this case, we cannot answer the query by utilizing one of the knowledge document only and nor one of the reasoning type. We need them both. Let's take a look.

To reason for stock names which have price • 5000 we need data from OWL/RDF document where the information about stock price resides. To reason for which one of the stocks that has the highest impact to profit among others, we need data from BN document where rules about impact of stocks to income resides. We need to combine the data so it can complete each other. This is the area where the Morpher works.

The Morpher will create individuals that did not exist before in OWL/RDF document in this case Income as a generic type. Generic type is a subclass of thing. The Morpher will then create special property has ProbValue to every individual exist in the system. The reasoner will set probabilistic values in BN document based on the data exist in OWL/RDF document do probabilistic reasoning and give the result back to OWL/RDF document by updating its respective property values. The result of running program shown in Fig. 4-6 as follows.

Case 2; Social CRM in higher education: Social CRM is the Customer Relationship Management (CRM) concept combined with the power of social media to generate new approaches in maintaining and enhancing relationships with customers. Social CRM Model contains semantic relationships between stakeholders and entities that must be understood thoroughly to identify the role of each

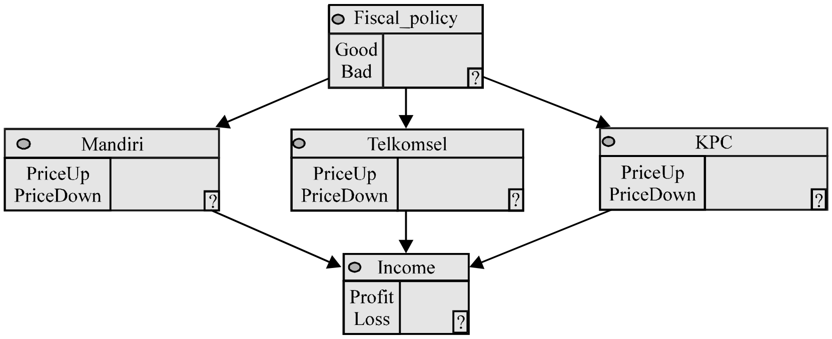

Fig. 4: Investor the relationship between concepts: fiscal policy, stocks and income

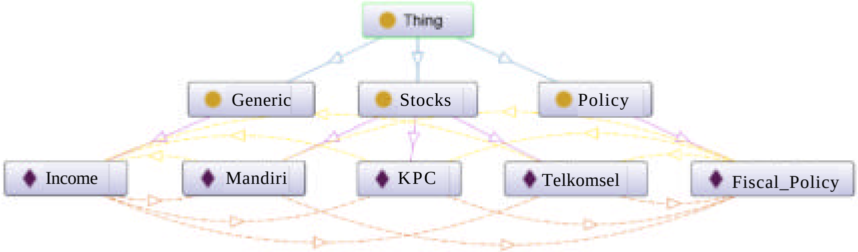

Fig. 5: Investor the result of running program for OWL document

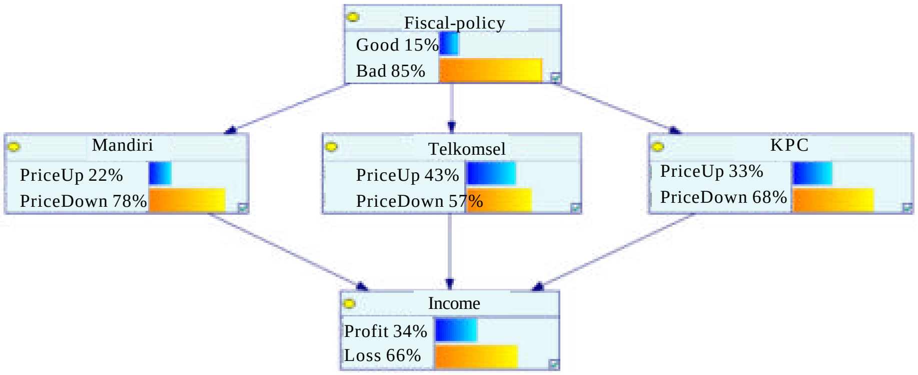

Fig. 6: Investor the result of running program for BN document

stakeholder and entities in the business process. Stakeholders include the university (in which there is the foundation, rector, lecturers and staffs), prospective students, students and alumni. These semantic relationships bring consequences to logical reasoning.

Entities included items contained in the Critical Success Factors (CSFs) such as budget and project management, knowledge management, technology selection and adaptation, vendor relationships, IT infrastructure, training, culture and leadership. Social CRM Model contains relationships of cause or effect of CSFs to the successful implementation of social CRM in the university. These causal relationships bring consequences to probabilistic reasoning. There is a simplified model that describes the semantic relationship between entities in a university (Fig. 7). Other information regarding CSFs that determine the successful implementation of social CRM in the university resides in a BN document (Fig. 8). Suppose that we want to answer the query: display units that handle CSFs and its budget allocations for the successful implementation of social CRM for this case, we can only query from OWL document to get the result. List 3 factors of CSFs that most influence on the successful implementation of social CRM for this case, we can only query from BN document to get the result. List 3 units that handle CSFs and its

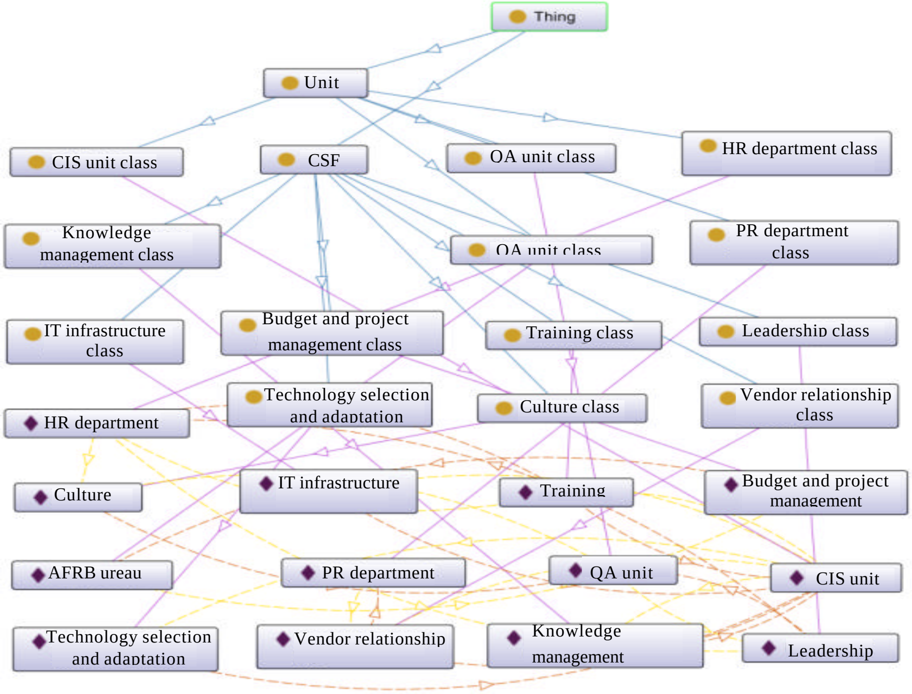

Fig. 7: Social CRM the structure of classes and individuals in OWL document

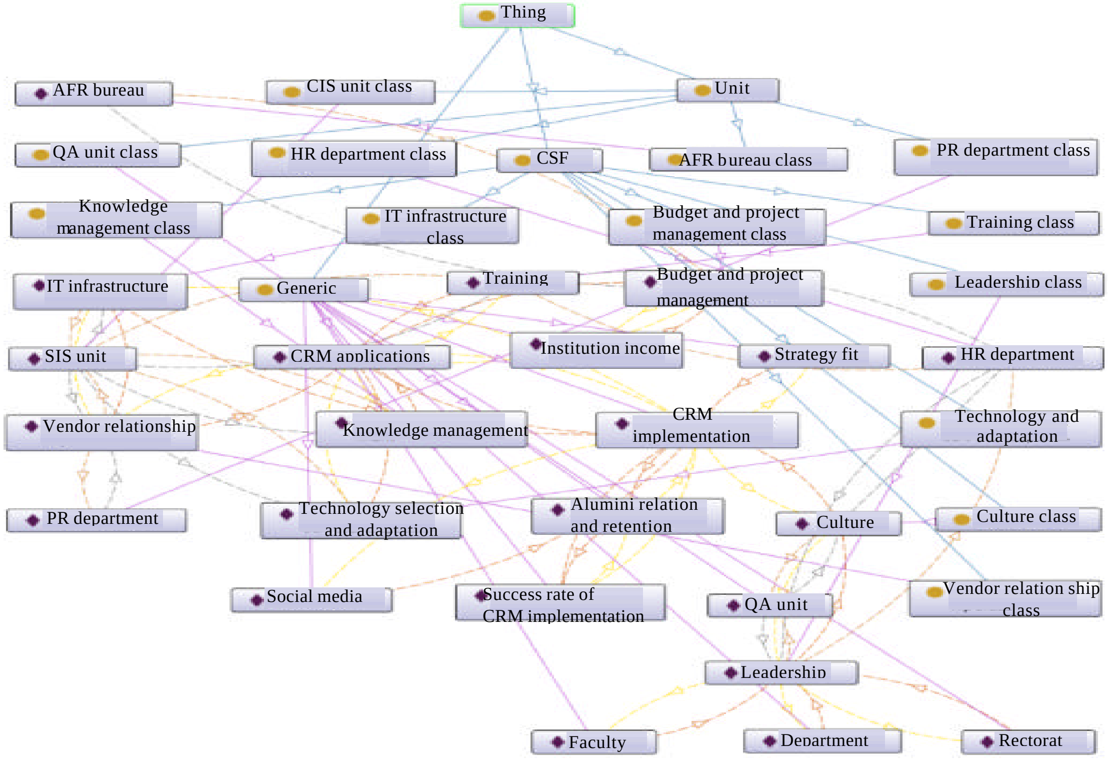

Fig. 8: Social CRM the result of running program for OWL document

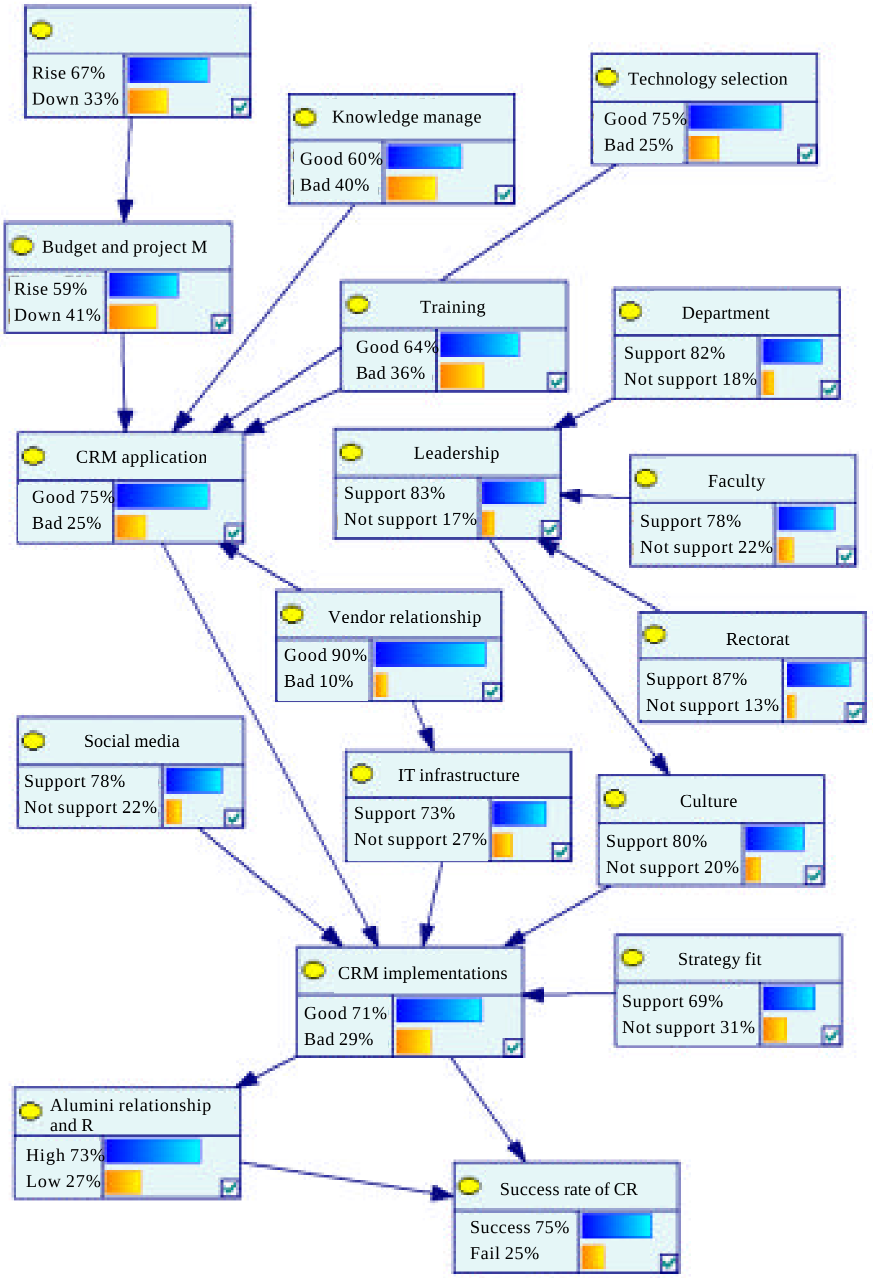

Fig. 9: Social CRM the result of running program for BN document

budget allocations which are prioritized to realize the successful implementation of social CRM for this case, we need to query from OWL and BN documents simultaneously to get the result. The example of SPARQL query for query No. 3 is as follows (Fig. 9):

Algorithm 1; SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX socialcrm:
        <http://localhost/latihan/ontologies/socialcrm#>
SELECT ?UnitName ?BudgetAlloc ?CSFVar ?SuccessRate
WHERE {?UnitName rdf:type socialcrm:Unit .
        ?UnitName socialcrm:Budget ?BudgetAllocation .
        ?CSFVar rdf:type socialcrm:CSF .
        ?CSFVar socialcrm:hasProbValueOfSuccessRate
        ?SuccessRate}
ORDER BY DESC(?SuccessRate)
LIMIT 3
```

The system will generate a knowledge base that combine data from both of sources: OWL and BN documents. The query will then be executed against the generated knowledge base to get the result. An example of the result can be seen in Fig. 8 and 9.

CONCLUSION

The need for the use of ontologies in software applications is currently growing and it is in line with the development of application architectures that share data and information on a massive scale. Data and information are connected to each other to form a knowledge base about a particular domain. An intelligent software application must be able to answer user queries about needed data and information from its knowledge base through inference mechanism. Currently, the ontology-based application can only make inferences through query against a knowledge base to perform logical reasoning. For probabilistic reasoning, the application must do so with separate knowledge base, separate processing, or with the third-party applications.

Sometimes, the query information is not only based on a logical proposition but also a statement containing the probability or degree of certainty. To achieve this condition, the application must have the ability to do both logical and probabilistic reasoning to the knowledge base simultaneously. The framework and its implementation in a form of prototype is demonstrated in investor problem case and social CRM for higher education case. The experiment with those two cases shows that this framework works well in performing both logical and probabilistic reasoning simultaneously and get the desired results. The existence of the software with reasoning capabilities of this kind perform an intelligent and powerful application.

REFERENCES

Baader, F., I. Horrocks and U. Sattler, 2008. Chapter 3: Description Logics. In: Handbook of Knowledge Representation, Harmelen, F.V., V. Lifschitz and B. Porter (Eds.). Elsevier, Amsterdam, Netherlands, pp: 135-182.

Cesar, G.D.P.C. and B.L. Kathryn, 2005. Multi-Entity bayesian networks without multi-tears. George Mason University, Fairfax, Virginia.

Costa, P.C. and K.B. Laskey, 2006. PR-OWL: A Framework for Probabilistic Ontologies. In: Frontiers in Artificial Intelligence and Applications, Bennett, B. and F. Christiane (Eds.). George Mason University, Fairfax, Virginia, pp: 237-249.

Ding, Z., P. Yun, P. Rong and Y. Yang, 2005. A bayesian methodology towards automatic ontology mapping. Proceedings of the AAAI-05 C&O Workshop on Contexts and Ontologies: Theory, Practice and Applications, July 9, 2005, University of Maryland, Baltimore, Maryland, pp: 1-8.

Fenz, S., A.M. Tjoa and M. Hudec, 2009. Ontology-based generation of bayesian networks. Proceedings of the International Conference on Complex, Intelligent and Software Intensive Systems, March 16-19, 2009, IEEE, Vienna, Austria, ISBN:978-1-4244-3569-2, pp: 712-717.

Grau, B.C., I. Horrocks, B. Motik, B. Parsia and S.P. Patel *et al.*, 2008. OWL 2: The next step for OWL. Web Semant. Sci. Serv. Agents World Wide Web, 6: 309-322.

Laskey, K.B., 2008. MEBN: A language for first-order Bayesian knowledge bases. Artif. Intell., 172: 140-178.

Nilsson, N.J., 1986. Probabilistic logic. Artif. Intell., 28: 71-87.

Santos, E. and J.C. Jurmain, 2011. Bayesian knowledge-driven ontologies: Intuitive uncertainty reasoning for semantic networks. Proceedings of the 2011 IEEE International Conference on Systems, Man and Cybernetics (SMC), October 9-12, 2011, IEEE, Hanover, USA., ISBN:978-1-4577-0653-0, pp: 856-863.

Sirin, E., B. Parsia, B.C. Grau, A. Kalyanpur and Y. Katz, 2007. Pellet: A practical OWL-DL reasoner. Web Semant. Sci. Serv. Agents World Wide Web, 5: 51-53.