Generation of Bayesian Networks using the Antipatten Ontology

Dimitrios Settas
United Nations University
International Institute for Software Technology
Macau, SAR, China
settdimi@iist.unu.edu

Antonio Cerone
United Nations University
International Institute for Software Technology
Macau, SAR, China
antonio@iist.unu.edu

Stefan Fenz
Institute of Software Technology and Interactive Systems
Vienna University of Technology and SBA Research
Vienna, Austria
stefan.fenz@tuwien.ac.at

Abstract

Apart from the plethora of antipatterns that are inherently informal and imprecise, the information used in the antipattern ontology itself is many times imprecise or vaguely defined. For example, the certainty in which a cause, symptom or consequence of an antipattern exists in a software project. However, ontologies are not capable of representing uncertainty and the effective detection of antipatterns taking into account the uncertainty that exists in software projects, stills remain an open issue. Bayesian Networks (BNs) have been previously used in order to measure, illustrate and handle antipattern uncertainty in mathematical terms. In this paper, we explore the ways in which the antipattern ontology can be used to generate Bayesian Networks. This approach allows software developers to quantify the existence or occurrence of an antipattern attribute using Bayesian Networks, based on probabilistic knowledge contained in the antipattern ontology regarding antipatterns attributes. The approach is exemplified with an ontology-based model generated using BNTab.

1. Introduction

Antipatterns are the next generation of design pattern research and are related with patterns in the sense that design patterns can evolve into antipatterns. An antipattern is a new form of pattern that has two solutions. The first is a problematic solution with negative consequences and the other is a refactored solution, which describes how to change the antipattern into a healthy solution [1]. There exist different categories of antipatterns. For the purposes of this paper, software project management antipatterns are used in order to exemplify the proposed approach.

In previous work, Bayesian Networks (BNs) [6] and the formalism of ontology [7] have been used separately in order to produce statistical and extensible ontological models of antipatterns. Bayesian Networks provided a framework for project managers, who would like to model the cause-effect relationships that underlie an antipattern, taking into account the inherent uncertainty of a software project. By applying the formalism of ontology, [7], a common lexicon of term that can be communicated across people and software tools was defined. This provided the basis for the application of further methodology in order to address similar antipattern ontologies and implement SPARSE [7], an ontology-based intelligent system that can detect antipatterns based on the symptoms that appear during a software project.

In this paper, we intertwine the formalisms of Bayesian Networks (BNs) and Ontology. The goal of this exploration is to further develop the existing antipattern ontology knowledge base that has been implemented with the Web Ontology Language (OWL), using Bayesian Network analysis. The task of capturing incomplete or uncertain antipattern knowledge is essential but is often not explored based on assumptions on the certainty and accuracy of software project data. Support for uncertainty is essential for the antipattern ontology because the creation of new antipatterns using the antipattern OWL ontology often relies on information from past project which comes from experience, memory and intuition. Antipattern ontology contributors might often rely on their own expert judgement to define how antipatterns might be linked, which has a clear effect on the effectiveness of the antipattern detection process. By incorporating BNs in the antipattern OWL ontology, we can represent probabilistic information regarding antipatterns and their attributes. This approach incorporates probabilistic information in the ontology [5] and allows the semi-automatic generation of BNs using BNtab [4]. In that way, BNtab can illustrate the uncertainty that exists in antipatterns and antipattern attributes.

This paper is divided in 6 sections, which are organized as follows: section 2 describes the background, the related work and the literature review used in our research. Section 3 presents the antipattern ontology. Section 4 describes the probabilistic extensions that were made in the antipattern ontology. In Section 5, the process of using BNTab to semi-automatically generate antipatter BNs is exemplified. Finally, in section 6, the findings are summarized, future work is proposed and conclusions are drawn.

2. Background and Related Work

A knowledge representation language has a syntactic and an inferential aspect. The syntactic aspect is notational and refers to the way in which one stores information in an explicit format. In Bayesian Networks this refers to the notational aspect of cause and effect relationships and the probability values of the NPTs (Node Probability Table). In the case of ontology this refers to the concepts and properties of the Ontology Web Language (OWL).

Both BBNs and ontology have the ability to reason within these formalisms. In the case of BBNs, reasoning is carried out under uncertainty and by illustrating the uncertainty that surrounds antipatterns. The Bayesian capacity to draw strong inferences from sparse data can not be ignored. However, it is the formalism of ontology that provided a sound basis for the further development of methods and software tools using ontology. By incorporating a BBN model of an antipattern inside the ontology, managers can still take advantage of the mathematical analysis of BBNs while using the formalism of ontology.

At the moment there exist a variety of ontology editors offering different functionality. Protege is one of the most widely used open source ontology editors and was chosen for the implementation of SPARSE and the antipattern OWL ontology [7]. Further development of the ontology in previous work tackled the challenge of providing a web-based collaborative environment by using WebProtege [9], which is a web-client for Collaborative Protege.

There is a variety of frameworks and tools under development in order to support the study of uncertainty in ontologies. BayesOWL framework [3], can translate an OWL ontology into a BBN structure, but is still under development. Other frameworks currently under development such as PR-OWL [2] can extend OWL vocabulary for representing uncertainty in different expressivities. These frameworks are being studied by the Uncertainty Reasoning for the World Wide Web Incubator Group (URW3-XG) but at the moment they lack compatibility with OWL. In this paper, the issue of quantifying uncertainty in the antipattern ontology is addressed by using the BNTab Protege plugin. BNTab is the only plug-in that is currently available for Protege that can generate Bayesian Networks. The BNTab plug-in enables users to efficiently generate Bayesian networks based on existing ontologies.

3 The Antipattern OWL Ontology

A complete description of the complete antipattern ontology [7] is outside the scope of this paper. However it is important to understand the existing ontology before describing the required extensions in order to allow ontology-based BN generation. The ontology consists of 7 concepts, 21 roles (19 object and 2 datatype roles), 192 individuals and 7 SWRL rules.

The antipattern ontology consists of seven concepts. In addition to the intuitive `Antipattern` concept, The other three antipattern-related concepts have been defined directly as subclasses of the `Antipattern` concept, that is,

$$\texttt{SoftwareDevelopment} \sqsubseteq \texttt{Antipattern}$$
$$\texttt{SoftwareArchitecture} \sqsubseteq \texttt{Antipattern}$$
$$\texttt{SoftwareProjectManagement} \sqsubseteq \texttt{Antipattern}$$

The `Cause` concept is used in order to define the causes of the ontology. It has been defined as the subclass of the intersection of three universal role restrictions.

$$\begin{aligned}\texttt{Cause} \sqsubseteq\ & \forall \texttt{causeToCause.Cause} \sqcap \\ & \forall \texttt{causeToSymptom.Symptom} \sqcap \\ & \forall \texttt{causeToConsequence.Consequence}\end{aligned}$$

In that way, for a `Cause` instance, all of its values in the `causeToCause` role belong to the `Cause` concept, all of its values in the `causeToSymptom` role belong to the `Symptom` concept and all of its values in the `causeToConsequence` role belong to the `Consequence` concept.

The `Symptom` concept is used in order to define the symptoms of the ontology. It has been defined as the subclass of the intersection of four universal role restrictions. The definition is similar to the `Cause` concept, apart from an additional restriction on the `symptomToConsequence` role that defines the existence of at least one value in the role.

The `Consequence` concept is used in order to define the consequences of the ontology and it has been defined as the subclass of a single universal restriction.

The `Antipattern` concept is the root concept of the antipattern hierarchy and is defined in terms of at least one cause, symptom and consequence instance values in the corresponding roles:

$$\begin{aligned}\texttt{Consequence} \sqsubseteq\ & \forall \texttt{hasCause.Cause} \sqcap \\ & \forall \texttt{hasSymptom.Symptom} \sqcap \\ & \forall \texttt{hasConsequence.Consequence} \sqcap \\ & \geq 1\ \texttt{hasCause}.\top \sqcap \\ & \geq 1\ \texttt{hasSymptom}.\top \sqcap \\ & \geq 1\ \texttt{hasConsequence}.\top\end{aligned}$$

The ontology roles allow the definition of basic knowledge related to antipattern causes, symptoms and consequences, as well as to their correlations. The ontology defines two datatype roles for providing human-readable textual descriptions for ontology instances. The `title` role can be used in order to define a short title for an instance and the `description` role can be used in order to provide a detailed documentation. Intuitively, the `title` role can be used as the human-readable description of an ontology instance `rdf:ID`.

The antipattern ontology allows the definition of correlations among causes, symptoms and consequences. In this section, for simplicity, we describe only the roles that correlate a cause with a cause.

The `causeToCause` object role allows the correlation of a cause with another cause. In that way, there is no need to state explicitly all the causes of a specific antipattern. The ontology reasoning procedure through SWRL rules is able to infer all the relevant (implicit) causes for a specific antipattern following the `causeToCause` relations.

4. Probabilistic extensions to the Antipattern OWL Ontology

The required extensions to allow ontology-based BN model generation using BNTab included 6 new ontology concepts and 12 roles (object type). The AntipatternAprioriProbability concept is used in order to define the different instances of antipatterns that have an associated probability value. The object property assertions of this concepts is the specific antipattern itself and its occurence, which can be of type ThreepointLikertScale (low,medium or high). Similarly the concepts CauseAprioriProbability, SymptomAprioriProbability and ConsequenceAprioriProbability were added in order to define the instances of antipattern attributes which have an associated probability value. Simarly to the AntipatternAprioriProbability, the concepts of the antipattern attributes also have two new object property assertions which define a specific attribute and its occurence measured with a probabilistic value scale. The other two concepts required for BN model generation were the ontology concepts Scale and ThreepointLikertscale added to the ontology in order to determine potential states and weights of the Bayesian network nodes. The boolean scale can be used to declare the occurence of an antipattern or antipattern attribute as True or False values, while the three point likert scale provides a Low, Medium and High scaling of occurence values. The ontology can be extended to handle different scales according to different user needs.

The "AntipatternAprioriProbabilityhasAntipattern" object role allows the correlation of an antipattern with an AntipatternAprioriProbability instance which aims to declare that a specific antipattern has associated probabilistic information on its occurrence. Similarly, the "AntipatternhasAntipatternAprioriProbability" is a role of the antipattern concept and correlates an antipattern with specific probabilistic information of AntipatternAprioriProbability members. The "AntipatternAprioriProbabilityhasProbability" object role correlated an AntipatternAprioriProbability instance with probabilistic information. This information combined with the previous roles can be used to correlate an antipattern with specific probabilities of occurence according to the chosen scale concept. The same roles have been defined for the cause, symptoms and consequences antipattern attributes. The resluting roles can associate any antipattern attribute of the ontology with probabilistic information that can be used by BNTab in order to generated a BN model in a semi-automated manner.

5. Example Ontology-based antipattern BN model generation

This section exemplifies the semi-automatic creation of an antipattern BN model based on knowledge described in the antipattern ontology. After installing the BNTab plugin in the Protege ontology editor and installing the required Netica BN editing software, the first step required in the BN model creation process is to choose the Boolean scale value which will allow the BN model to display the possibility that an antipattern attribute exists with an associated "True" and "False" percentage. Then the property "hasvalue" is chosen as this property correlates any chosen ontology construct with a specific value, in this example a value from "0" to "1" indicates if possibility of occurence or existence of an antipattern attribute with "1" being "True" and "0" being "False". In the next step, ontology individuals with probabilistic information will be added to the BN model. These are refered to as "classes" in BNTab. In the example of Figure 1, the classes "Cause", "Symptom" and "Consequence" are chosen in order to create the BN model. The next step is to select the specific individuals of those classes that the user is interested in modelling as a BN. The user has to select the corresponding properties to correlate the chosen attributes to the probabilistic knowledge contained

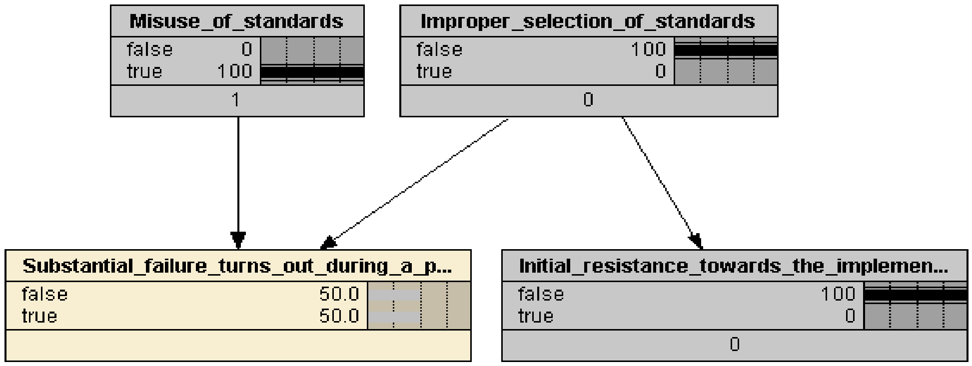

Figure 1. Creating antipattern BN models using BNTab

in the ontology regarding those antipatterns. This is carried out by selecting from ten types of properties. These properties are include the object roles additions that were described in the previous sub-section. For the example of Figure 1 the properties ”CauseToCause”, ”CauseToConsequence” and ”CauseToSymptom” were used in order to correlate the nodes of the BN model with relationships that define how the causes of a specific set of antipatterns are connected through their cause, symptom and consequence attributes. By selecting ”Generate Bayesian Network”, the plug-in launches Netica BN editor and displays the resulting antipattern attribute BN model. The user can then save the file as a .DNE file and process it later on with any BN editing software. The resulting BN model (Fig. 1) contains 4 nodes and 3 cause and effect relationships.

6 Conclusions

The main contribution of this paper is the application of Bayesian Networks on the antipattern OWL ontology. Bayesian modeling proved to be particularly useful and well suited to the antipattern ontology by capturing and by illustrating the cause and effect relationships between antipattern variables and by providing a solid graphical representation of the probabilistic relationships among the set of variables. BNTab can be used to generate ontology-based antipattern BN models. Future work will attempt to incoorporate the BNtab plugin within the web-based collaborative version of the antipattern ontology [8] and antipattern contributors will be able to use the approach proposed in this paper simultaneously through the Web. This will ensure that all antipattern ontology users are editing the most up-to-date version of the ontology. Ultimately, SPARSE [7] can benefit from this approach by including associated BNs with the resulting detected antipatterns in order to visualize the uncertainty of the data used for the detected antipatterns.

References

[1] W. Brown, H. McCormick, and S. Thomas. *AntiPatterns in Project Management*. Wiley Computer publishing, 2000.

[2] R. N. Carvalho, K. B. Laskey, and P. C. G. Costa. Compatibility formalization between pr-owl and owl. In *In Proceedings of the First International Workshop on Uncertainty in Description Logics (UniDL 2010), held at the International Joint Conference on Automated Reasoning (IJCAR 2010)*, July 2010.

[3] Z. Ding, Y. Peng, and R. Pan. A bayesian approach to uncertainty modelling in owl ontology. In *In Proceedings of the 2004 Int.Conference on Advances inIntelligent Systems*, 2004.

[4] S. Fenz. http://stefan.fenz.at/ontology-based-generation-of-bayesian-networks/.

[5] S. Fenz, A. M. Tjoa, and M. Hudec. Ontology-based generation of bayesian networks. In *In Proceedings of the Third International Conference on Complex, Intelligent and Software Intensive Systems – International Workshop on Ontology Alignment and Visualization*, pages 712 – 717, 2009.

[6] D. Settas, S. Bibi, P. Sfetsos, I. Stamelos, and V. Gerogiannis. Using bayesian belief networks to model software project management antipatterns. In *4th ACIS International Conference on Software Engineering Research, Management and Applications (SERA 2006)*, pages 117–124, 2006.

[7] D. L. Settas, G. Meditskos, I. G. Stamelos, and N. Bassiliades. Sparse: A symptom-based antipattern retrieval knowledge-based system using semantic web technologies. *Expert Systems with Applications, Elsevier*, 38(6):7633–7646, 2011.

[8] D. L. Settas, G. Meditskos, I. G. Stamelos, and N. Bassiliades. Detecting antipatterns using a web-based collaborative antipattern ontology knowledge base. In *In Proceedings of the 5th ONTOSE 2011 Workshop (International Workshop on Ontology, Models, Conceptualization and Epistemology in Social, Artificial and Natural Systems)*, To Appear.

[9] T. Tudorache, J. Vendetti, and F. N. Noy. Web-protege: A lightweight owl ontology editor for the web. In *In Proceedings of the fourth workshop in the The OWL: Experiences and Direction (OWLED)*, 2008.