Knowledge based Bridge Engineering - Artificial Intelligence meets Building Information Modeling

Dominic Singer, Maximilian Bügler, André Borrmann
Chair of Computational Modeling and Simulation
Leonhard Obermeyer Center, Technische Universität München, Germany
dominic.singer@tum.de

Abstract. In the research project "KBE4Infra", knowledge-based engineering (KBE) techniques for capture, storage and reuse of engineering knowledge in bridge engineering are investigated. A rich data source of bridge engineering knowledge, are Bridge Management Systems (BMS). The work present addresses the question, whether it is possible to gain knowledge by the use of data in bridge management systems and how this knowledge can be used beneficially in conceptual bridge design phases. To achieve this, we propose the use of a Bayesian network. Thus, we introduce and review well known network learning and inference algorithms for Bayesian networks. Then, promising methods are matched to the needs of the proposed approach. Finally, a proof of concept will show the practicality of the chosen approach.

1. Introduction

In the research project "KBE4Infra", knowledge-based engineering (KBE) techniques for capture, storage and reuse of engineering knowledge in bridge engineering are investigated (Singer et al. 2015). In the course of this project, generation and comparison of various design alternatives in early design phases are mainly taken into account. KBE aims at the *"the use of advanced software techniques to capture and re-use product and process knowledge in an integrated way"* (Stokes 2001). Compared to previous approaches, the present system is strongly based on building information modeling. Furthermore, visual programming is used for the geometric semantic modeling of the bridge. This enables the possibility of real-time modelling, evaluation, and optimization of bridge structures. While setting up a knowledge base and an inference engine, formalizing engineering knowledge in computer interpretable rules is the most important and difficult step.

In recent years, Artificial Intelligence (AI) techniques have been intensively researched in civil engineering (Lu et al. 2012; Schneider et al. 2015). In the course of these studies, the use of expert systems, rule-based systems, knowledge-based engineering, case-based reasoning, neural networks, machine learning, and their underlying technologies like first order logic, fuzzy logic, forward and backward chaining, semantic networks, genetic programming were investigated to enhance typical civil engineering tasks. In bridge engineering, the application of AI methods was considered for various tasks throughout bridge lifecycle phases, like in conceptual design, analysis, construction, operation, and maintenance. To the present day, these systems have hardly reached a broader use in practice for several reasons (Reich 1995): The research was often focused on narrow and specialized engineering subtasks, not on larger and more integrated problems. In addition, a building product model was not yet established and therefore not used. However, a product model, acting as a communication base, is indispensable to gain reliable design solutions, since in bridge engineering design tasks interact closely and influence each other.

Currently, a fundamental change in the Architecture, Engineering and Construction (AEC) industry by the introduction of the Building Information Modeling (BIM) technology (Eastman et al. 2011) takes place. BIM aims to represent the complete building facility in a digital product

model (Eastman 1999), which is used throughout the whole life-cycle. Furthermore, parametric modeling is more and more incorporated for the design of infrastructure facilities (Boeykens 2012). Apart from that, the Industry Foundation Classes (IFC), a non-proprietary data-format for the exchange of product models, has already reached a convenient level. This data format has the potential to increase the integration of existing analysis tools in the design workflow of bridges dramatically in the near future. Figure 1 shows a building information model of reinforced concrete road bridge in a low to mid level of detail.
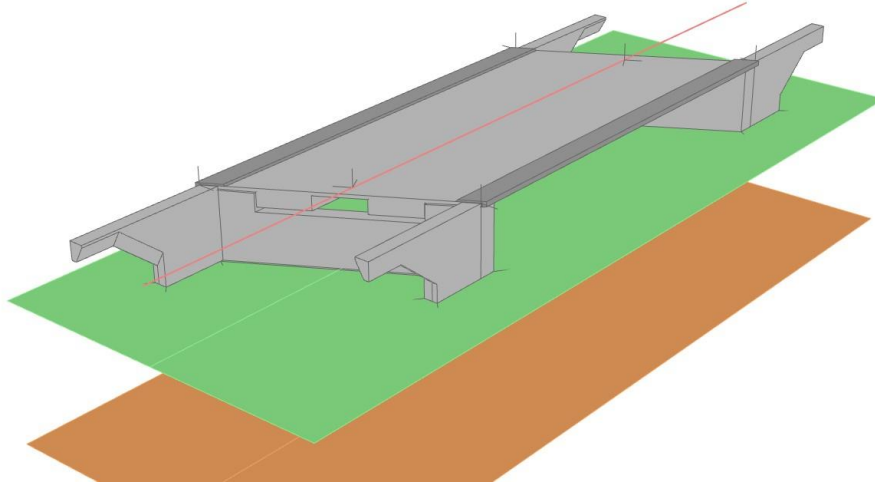

Figure 1:   Building information model of a bridge

As part of knowledge modeling, knowledge representation and inference, suitable existing AI methods have to be chosen. In this paper, we propose to use a Bayesian network to achieve these tasks. Therefore, we introduce and review well known network learning and inference algorithms for Bayesian networks. Then, promising methods are matched to the needs of the proposed KBE approach for bridge engineering. Finally, a proof of concept will show the practicality of the chosen approach.

2. Problem definition

Preliminary design focuses on the exploration of various design options leading to an optimal or near-optimal design solution with regard to technical, ecological, creative, and economic aspects of the planned structure. The generated design solution serves as a basis for determining the construction costs, preparing the tendering, elaborating the detailed design, and finally executing the construction work itself. The design of bridge structures is a highly iterative process, which is heavily dependent on external conditions, such as the location and the intended function of the structure.

As part of the bridge design, a bridge structure is described by the essential design elements like superstructure, substructure, foundation, abutment, bearing, bridge equipment, materials and construction methods. For each component a very large number of corresponding regulations and guidelines specifying the design and function exists. The design for each bridge is mainly dependent on the boundary conditions like alignment and traffic type or the existing terrain and soil conditions found in-situ. Beyond the engineering knowledge in codes and guidelines, knowledge is implicitly represented by existing bridges. It is desirable to get useful hints towards possible design solutions for the present circumstances. We aim to use this data to gain knowledge for new design problems and narrow the design space.

A rich data source to achieve this, are Bridge Management Systems (BMS). BMSs are software systems to manage large infrastructure assets. They are usually operated by the road authorities of each country. A BMS consists of a database where all information belonging to a bridge structure is documented. The database typically holds general information like the name of the bridge structure, the location, as well as construction data, the condition of bridge parts, pictures, and drawings. For our work, data like bridge type, bridge length, number of piers and superstructure type are important information stored in the database. In Germany the bridge management system "SIB Bauwerke" depicted in Figure 2 is well known; All over the world several systems are used, which do not differ greatly from one another.

Figure 2: General view in the bridge management system SIB-Bauwerke (WPM - Ingenieure 2015)

This paper addresses the question, whether it is possible to gain knowledge by the use of data in bridge management systems and how this knowledge can be used in conceptual bridge design phases. With the help of the present approach we want to support bridge engineers throughout their planning tasks, for example a feasibility analysis, by answering questions like the following ones: (i) Which construction type, materials, superstructure types are usually used to build a bridge crossing a river with a width of 10m? (ii) Which superstructure height needs to be reckoned for a certain span width and superstructure type? (iii) Which design options cannot be used for the conditions in-situ? The information delivered by the system and further settings given by the bridge engineer are then used to generate BIM models for each variant of the bridge. With the help of these models further simulation and analysis processes could be performed.

3. Methods

3.1 Bayesian network

In artificial intelligence various methods for knowledge representation exist. To represent knowledge in an uncertain domain and to perform probabilistic reasoning Bayesian networks are well known. They are based on Bayes' theorem, which describes the conditional probability $P(A|B)$ of an event A for a given event B. It is defined as:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

A Bayesian network is defined by a directed acyclic graph (DAG), where each node N represents a discrete or continuous random variable X $= \{X_1, \ldots, X_n\}$ and the edges E represent conditional dependencies among variables. A directed link from node X to node Y says that X is a parent of Y. An arrow between two variables is usually interpreted as a direct influence of X (cause) on Y (effect). If no connection between two nodes exists, they are conditionally independent of each other. This is very intuitive and makes it easy for domain experts to formalize dependencies in the domain. Once the topology of the graph is defined, each node of the graph is associated with a conditional probability distribution $P = (Xi \mid parents(Xi))$, which determines the impact of the parents of one node. The probabilities are denoted in conditional probability tables (CPT). The joint probability for a set of variable values x $= \{x_1, \ldots, x_n\}$ can be calculated by multiplication of each probability distribution:

$$P(X_1 = x_1 \wedge \ldots \wedge X_n = x_n) = P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i|parents(x_i))$$

To set up the Bayesian network, two main tasks occur. First, one needs to learn the network structure, second is to calculate the A-priori conditional probability distributions for each node. In order to perform these steps, a sufficient amount of training data sets is needed. In the following sections, algorithms are introduced that allow to automatically generate the topology of the graph and the conditional probability distributions. After setting up the Bayesian network, it is possible to perform probabilistic reasoning tasks. For the present application this means that we take existing data sets from BMS to learn the network structure and to calculate the probability distribution tables and then perform probabilistic reasoning for a new bridge design problem.

3.2 Structure learning

Beside the possibility to define the graph structure by the help of a domain expert, it is also feasible to define the graph structure of a Bayesian network by learning from training sets in an automated manner. Therefore, algorithms like the CI-based search algorithm (De Campos 2006) or the K2 algorithm (Cooper et al. 1992) exist. The Tree Augmented Naive Bayes (TAN) method was introduced by (Friedman et al. 1997). This algorithm creates a Bayesian network structure the following way.

1. Compute the mutual information function $I(X_i, X_j)$

   for each pair of variables where $i \neq j$

$$I(X,Y) = \sum_{x,y} P(x,y)\log\frac{P(x,y)}{P(x)P(y)}$$.

2. Generate a undirected graph where the edges are weighted by $I(X_i, X_j)$.

3. Find a maximum weighted spanning tree, applying Kruskal's algorithm.

4. Set a root variable and set the direction of all edges to be outward from it.

3.3 Inference in Bayesian networks

After setting up the Bayesian network, we would like to reason on this network. First, we need to differentiate between exact inference methods and approximate inference methods for Bayesian networks. Part of exact inference are the variable elimination algorithm and clustering algorithms. Common known algorithms for approximate inference in Bayesian networks are direct sampling methods, rejection sampling and likelihood weighting.

Also well-known are the Markov chain Monte Carlo (MCMC) algorithms. Unlike the before mentioned algorithms, samples are not generated randomly from scratch. Here, based on the current sample, the next sample is generated by performing small random changes. A particular form of a MCMC is called Gibbs sampling (Neal 1993) which is well suited for inference in Bayesian networks.

Gibbs sampling in Bayesian networks works as follows:

1. Start with an arbitrary state. Fix the evidence variables.
2. Initialize all other variables.
3. Repeat n times:
    a. Randomly choose a non-evidence variable X
    b. Resample this variable X from P (X | all other variables)
    c. Count current state
4. Normalize.

Usually a pre-sampling or burn-in is performed to begin the Gibbs sampling from a more realistic starting state.

4. Proof of concept

In the following is to be demonstrated that the use of a Bayesian networks permits the determination (at least with a certain probability) of bridge design parameters on the basis of given boundary conditions. Up to now, it has not been possible to get access to real data from a bridge management system. However, in order to proof our approach, we generated random test data sets by the help of a knowledge-based system. We assume, that the generated output data is compliant to the current codes and guidelines. We focused on reinforced concrete road bridges. We choose the following variables to generate our test sets, also depicted in Figure 3: terrain depth, bridge length, cross section, horizontal curvature, bridge crossing angle, bridge type, number of piers, material type, superstructure type and superstructure height. In Table 1 the chosen minimum and maximum values as well as the discretization for each variable is shown. Furthermore, the use as an input or output design parameter is indicated.

The minimal necessary amount of data sets is mainly dependent on the number of connections in the network structure. In order to count a sufficient number of samples for all possible combinations and each variable, an either high number of training data or a very heterogenic set is necessary. In order to get meaningful results, we generated 10,000 training data sets. SIB Bauwerke contains about 8,000 data sets for bridges in the state Bavaria along motorways. The number would be much higher if all states in Germany and the other road types would be taken into account. For those data sets, the proposed approach could be used the same way. Comparable BMS systems in other countries typically hold a similar number of data sets.
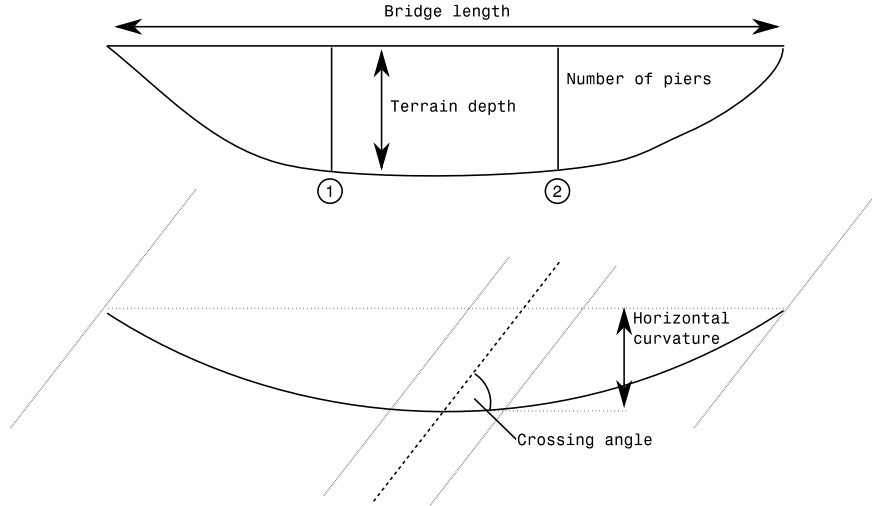
Figure 3:   Chosen bridge parameters

Table 1: Discretization of variables

Variables	Type	Min value	Max value	Discretization	Used as
Bridge length	Continuous	0	100	25	Input
Crossing angle	Continuous	0	60	5	
Cross section	Discrete	{RQ 7.5, RQ 9.5, RQ 10.5, RQ 15.5}			
Horizontal curvature	Continuous	0	13	1	
Terrain depth	Continuous	-50	0	10	
Bridge type	Discrete	{frame bridge, continuous girder bridge, arc bridge}			Output
Material type	Discrete	{reinforced concrete, pre-stressed concrete}			
Number of piers	Continuous	0	6	1	
Superstructure height	Continuous	0.5	5	0.5	
Superstructure type	Discrete	{plate, T-beam, girder box}			

We considered two variants for the generation of the Bayesian network. First is the generation by the help of the Tree Augmented Naive Bayesian Network Algorithm (TAN). Therefore, the algorithm was extended to work with multiple class nodes (Van Der Gaag et al. 2006). The spanning tree is generated as usual, excluding all class nodes. Instead of then connecting all class nodes to all nodes in the spanning tree, which would create a very strongly connected network, only the edges to the class nodes with high mutual information are connected. Strongly connected networks cause large probability tables which require larger amounts of data to be populated. Edges are added in order of mutual information values until the desired connectivity is reached. This can be a maximum information spanning tree. Otherwise a threshold can be set on the minimal MI to create an edge. Furthermore, the interconnections among the class nodes have been set by an expert a-priori, in order to reflect the order in which the design choices are commonly made.

The second considered variant was the completely manual creation of the Bayesian network by a knowledge engineer. Both the necessary nodes as well as the edges in between are solely defined by a domain expert. In both cases the same class and non-class variables were chosen. Non-class nodes are bridge length, terrain depth, cross section, crossing angle, and horizontal curvature. The non-class nodes are used as input values and describe the boundary conditions

6

of a certain design problem. Class nodes are bridge type, superstructure type, number of piers, superstructure height, and materiel type. They provide the output values for the design parameters we search for. In Figure 4 and Figure 5 the graphs for both variants are depicted. Class variables have continuous border; the non-class node borders are dashed. Continuous styled edges occur in both graphs; the directions do not matter.
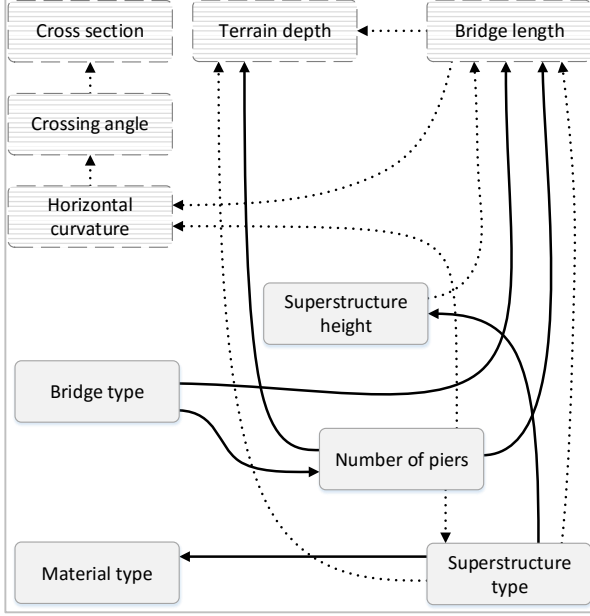

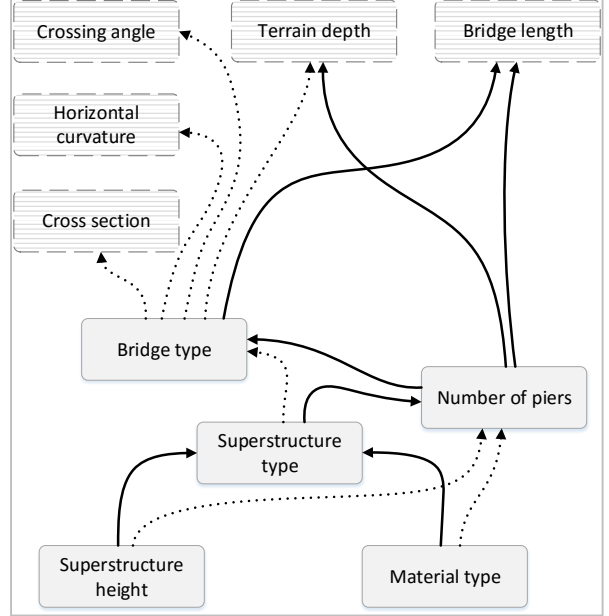Figure 4: Variant 1 - TAN generated graph

Figure 5: Variant 2 - Manually created graph

As you can see in the diagrams above, the input variables terrain depth und bridge length are the most important variables and the main conditions for the output variables in both variants, which reflects the reality as expected. The input variables horizontal curvature, crossing angle und cross section have minor influence on the output variables in both networks. Since we work with generated test data to learn the network structure in variant 1, the way of their generation could have main impact on the appearance of the network. It is possible that some conditions are implicitly effected by the generation algorithm. The manually created graph fits better to the design procedure bridge engineers usually perform than the automatically generated one (bridge length → bridge type → number of piers → superstructure type → superstructure height → material type).

In order to sample results from the network, all non-class nodes are assigned with the respective input values, and all class nodes are then sampled using the Markov Chain Monte Carlo (MCMC) algorithm, counting the occurrences of all combinations of values for all class nodes. Therefore, a likelihood for each possible class parameter combination can be calculated. Combination that never occur are discarded and the remaining combinations are ranked accordingly (Zheng et al. 2011). In Table 2 the conditional probability table for the class node superstructure type is exemplarily shown for variant 1. Given the number of piers equal to 2, a probability distribution of 37%, 33%, and 30% for the given superstructure types can be derived. This means that in the training data set an almost equal number of bridges with two piers for each superstructure exists. For variables with more than one condition the table gets more complicated. For bridges with more than four piers only bridges with a plate superstructure occur.

Table 2: Conditional probability table for superstructure type for variant 1

Number of piers	Plate	T-beam	Girder box
0	31%	37%	32%
1	41%	13%	46%
2	37%	33%	30%
3	60%	30%	10%
4	95%	5%	0%
5	100%	0%	0%

In order to test the generated Bayesian network, we set up two test scenarios. The scenario No. 1 describes the use case of an ordinary road (RQ 10.5) crossing a valley or river with a bridge length of 100m and a maximum height of 37m. Scenario No. 2 specifies a design problem for a bridge with a length of 57m and a maximum height of 15m for a motorway (RQ 15.5). To simplify the test, the variables crossing angle and horizontal curvature are set to 0 for both scenarios.

This experiments showed that the analysed input parameters returned a small set of feasible design variants with strong preferences that can be determined by those with distinctively high numbers of occurrences. The network generating small sets of possible designs implies that the number of possibilities is reduced successfully. If the information in the network would be sparse, all possible design combinations would be generated with uniform probability in the worst case. Since the truth is far from that, the results are meaningful. Exemplarily, the best four sample results for two different design scenarios and the two different graph generation variants are shown in Table 3 to 6. The two design scenarios differ in the values assigned to the non-class nodes which are denoted in the table description.

Table 3: Design scenario No. 1 – variant No. 1: Best sample results for evidence variables bridge length=100, horizontal curvature=0, terrain depth=-37, crossing angle=0, cross section = RQ10.5

Bridge type	Superstructure Type	Material Type	Superstructure Height	Number of piers	Probability
frame	plate	pre-stressed	1.0	4.0	32,95%
frame	plate	pre-stressed	1.0	6.0	11,97%
frame	plate	reinforced	1.0	4.0	10,22%
arc	plate	pre-stressed	1.0	4.0	7,37%

Table 4: Design scenario No. 1 – variant No. 2: Best sample results for evidence variables bridge length=100, horizontal curvature=0, terrain depth=-37, crossing angle=0, cross section = RQ10.5

Bridge type	Superstructure Type	Material Type	Superstructure Height	Number of piers	Probability
frame	T-beam	pre-stressed	1.0	3.0	10,83%
frame	T-beam	pre-stressed	1.0	2.0	8,60%
frame	girder box	pre-stressed	2.0	2.0	8,32%
frame	plate	reinforced	1.0	3.0	6,03%

8

Table 5: Design scenario No. 2 – variant No. 1: Best sample results for evidence variables bridge length=59, horizontal curvature=0, terrain depth=-15, crossing angle=0, cross section = RQ15.5

Bridge type	Superstructure Type	Material Type	Superstructure Height	Number of piers	Probability
continous girder	girder box	pre-stressed	1.0	0.0	61,78%
continous girder	T-beam	pre-stressed	1.0	0.0	12,62%
continous girder	plate	pre-stressed	1.0	0.0	9,05%
continous girder	girder box	pre-stressed	2.0	0.0	5,78%

Table 6: Design scenario No. 2 – variant No. 2: Best sample results for evidence variables bridge length=59, horizontal curvature=0, terrain depth=-15, crossing angle=0, cross section = RQ15.5

Bridge type	Superstructure Type	Material Type	Superstructure Height	Number of piers	Probability
continous girder	girder box	pre-stressed	1.0	0.0	23,29%
frame bridge	T-beam	pre-stressed	1.0	2.0	7,68%
frame bridge	girder box	pre-stressed	2.0	2.0	7,51%
frame bridge	T-beam	pre-stressed	1.0	3.0	5,55%

The outcome for the chosen test scenarios looks promising and showed that the approach works in principle. However, we expected more suitable advices. For example, in scenario No. 1 variant No. 1 only superstructure types of type plate are listed Alternatives with less piers and therefore other superstructure types are not mentioned. In Table 5 continuous girder bridges are combined with a number of piers equal to 0 for all entries and several superstructure types, which makes no sense from a bridge engineering perspective. The chosen discretization step of 1m for the variable superstructure height is too big. To achieve more feasible advices, more design variables and a finer discretization of those must be taken into account. Furthermore, a Bayesian network based on real-life data and not on randomly generated test data will provide much more realistic outcomes. Generally, the calculated probabilities for variant 1 are higher than for variant No. 2. This is because the generated network fits better to the generated training data sets than the manually created one. However, the manually created is the one which supports the bridge design process in a more intuitive way.

5. Conclusion and Outlook

In artificial intelligence Bayesian networks are well established for probabilistic reasoning in big data sources. We presented an approach to use Bayesian networks for purposes in preliminary bridge design phases, namely to gain knowledge implicitly stored in bridge management systems. It needs to be mentioned that the existing knowledge used in early bridge design phases cannot be captured in explicit if-then rules since it is highly uncertain. Thus, the use of a Bayesian networks perfectly fits this need.

It could be shown that, in principle, our approach is suitable to support the bridge design engineer by providing proper design solutions. To achieve more feasible advices, more design

variables and a finer discretization of those must be taken into account. The hypothetical case is limited on early design stages. In order to include further life-cycle aspects (e.g. operation and maintenance) into the decision making process, information like the operation cost of the bridge structure must be included. BMSs usually provide this type of information. We then can think about answering questions like, which construction type works well under which boundary conditions concerning the maintenance costs?

In future work, we will apply the shown concept to real world data from bridge management systems. Additionally, we want to develop methods to extend the Bayesian network with explicit knowledge to achieve a hybrid knowledge representation for bridge engineering. Furthermore, we will think of machine learning techniques to fill knowledge gaps in the knowledge base. Moreover, other artificial intelligence methods like artificial neural networks could be assessed for their applicability.

References

Boeykens, Stefan (2012): Bridging building information modeling and parametric design. In: *eWork and eBusiness in Architecture, Engineering and Construction: ECPPM 2012*, S. 453.

Cooper, Gregory F.; Herskovits, Edward (1992): A Bayesian method for the induction of probabilistic networks from data. In: *Machine learning* 9 (4), S. 309–347.

De Campos, Luis M (2006): A scoring function for learning Bayesian networks based on mutual information and conditional independence tests. In: *The Journal of Machine Learning Research* 7, S. 2149–2187.

Eastman, Charles M. (1999): Building product models. Computer environments supporting design and construction. Boca Raton, Fla.: CRC press.

Eastman, Charles M.; Teicholz, Paul; Sacks, Rafael; Liston, Kathleen (2011): BIM handbook. A guide to building information modeling for owners, managers, designers, engineers and contractors. 2nd ed. Hoboken, NJ: Wiley.

Friedman, Nir; Geiger, Dan; Goldszmidt, Moises (1997): Bayesian network classifiers. In: *Machine learning* 29 (2-3), S. 131–163.

Lu, Pengzhen; Chen, Shengyong; Zheng, Yujun (2012): Artificial Intelligence in Civil Engineering. In: *Mathematical Problems in Engineering* 2012 (6), S. 1–22. DOI: 10.1155/2012/145974.

Neal, Radford M. (1993): Probabilistic inference using Markov chain Monte Carlo methods.

Reich, Yoram (1995): Artificial Intelligence in Bridge Engineering: Towards Matching Practical Needs with Technology.

Schneider, Ronald; Fischer, Johannes; Bügler, Maximilian; Nowak, Marcel; Thöns, Sebastian; Borrmann, André; Straub, Daniel (2015): Assessing and updating the reliability of concrete bridges subjected to spatial deterioration - principles and software implementation. In: *Structural Concrete* 16 (3), S. 356–365. DOI: 10.1002/suco.201500014.

Singer, D.; Borrmann, A. (2015): A Novel Knowledge-Based Engineering Approach for Infrastructure Design. In: The Fourth International Conference on Soft Computing Technology in Civil, Structural and Environmental Engineering. Prague, Czech Republic.

Stokes, M. (2001): Managing engineering knowledge. MOKA: methodology for knowledge based engineering applications. London: Professional Engineering Pub.

Van Der Gaag, Linda C; De Waal, Peter R (2006): Multi-dimensional Bayesian Network Classifiers. In: Probabilistic graphical models. Citeseer, S. 107–114.

WPM - Ingenieure (2015): Übersichtsblatt SIB Bauwerke. Online verfügbar unter http://www.wpm-ingenieure.de/images/produkte/sib_bauwerke/uebersichtsblatt.jpg, zuletzt geprüft am 08.02.2015.

Zheng, Fei; Webb, Geoffrey I. (2011): Tree augmented naive Bayes. In: Encyclopedia of Machine Learning: Springer, S. 990–991.