Generic Knowledge Structures for Probabilistic-Network Engineering

Eveline M. Helsper

Linda C. van der Gaag

institute of information and computing sciences,
utrecht university

technical report UU-CS-2005-014

www.cs.uu.nl

Generic Knowledge Structures for Probabilistic-Network Engineering

Eveline M. Helsper and Linda C. van der Gaag

Institute of Information and Computing Sciences, Utrecht University,
P.O. Box 80.089, 3508 TB Utrecht, The Netherlands
{eveline,linda}@cs.uu.nl

Abstract. The process of engineering probabilistic networks can be supported by a library of generic knowledge structures. Such a knowledge structure is instantiated with domain-specific knowledge and is used to derive, in a number of steps, a segment of the graphical structure of a network. To provide for customisation to the application at hand, the structures are based on an in-depth knowledge analysis and capture, in an appropriate representation, the intricate details of the knowledge involved. We present, as an example, the generic knowledge structure that captures the relations between a test result and the underlying true value. As a guideline for its application we provide the derivation of a network segment in the field of oncology.

1 Introduction

An increasing number of knowledge-based systems build upon the formalism of probabilistic networks for their knowledge representation. A probabilistic network consists of a graphical structure, representing statistical variables and the (in)dependence relations between them, and an associated numerical part, describing a joint probability distribution over the represented variables [1]. Engineering a probabilistic network for an application usually has to be done with the help of domain experts and is generally considered a hard and time-consuming task. Methods and associated tools are called for to support this task.

We consider, as an example, the construction of a probabilistic network for medical diagnosis. In the medical domain testing plays an important role and the probabilistic network should capture the results of the various tests employed. We observe that a test is typically performed to reveal an underlying true value that is not directly observable. The test's result may not unambiguously reflect this true value and may depend, for example, on the skills of the laboratory technician. Because the relation between the true value and the test result can be quite intricate, modelling the domain-specific knowledge involved is not straightforward.

We propose in this paper to support the task of engineering probabilistic networks by a library of generic knowledge structures, where each such structure represents a unit of knowledge in domain-independent terms. This library contains, for example, a generic knowledge structure for tests that captures all knowledge that plays a role in explaining test results in terms of their underlying true value, independent of any specific application domain. Upon using the library, the knowledge engineer selects appropriate generic knowledge structures and instantiates these with the domain-specific knowledge for the application at hand. The instantiated knowledge structures are subsequently used in the design of the probabilistic network. Since building a probabilistic network involves taking design decisions that may depend on the requirements of the application, the represented knowledge may be modelled at different levels of abstraction. The generic knowledge structures should therefore allow for flexible customisation by the knowledge engineer.

To allow for customisation, a knowledge structure should be based on a thorough understanding of the knowledge involved. An in-depth analysis of the knowledge is therefore indispensable. The structure should further capture the intricate details of the knowledge in a representation that is easy to understand and that is capable to reflect those intricacies. Since capturing knowledge in the formalism of probabilistic networks may result in a representation from which the knowledge is not easily recognizable [2], we represent our knowledge structures independently of this formalism. Since a generic knowledge structure, after having been customised and instantiated with knowledge in the application domain, is used to derive a segment of the graphical structure of a probabilistic network, it further has associated, as a guideline to this end, an example derivation.

In this paper, we detail, as an example, a generic knowledge structure that captures the relations involved in explaining or predicting test results. This knowledge structure is based on an in-depth analysis of the knowledge involved. To provide support to the task of building a network for an application domain, we detail, with the structure, an example derivation of a segment of the graphical structure of the network.

The remainder of the paper is structured as follows. In Section 2, we describe how a library of generic knowledge structures can support a knowledge engineer in building a probabilistic network. In Section 3, we analyse the relations between test results and the underlying true values, and present a generic knowledge structure capturing this knowledge. In Section 4, we demonstrate how a segment of the graphical structure of a probabilistic network is derived from an instantiated knowledge structure in the domain of oncology. Next, we discuss related work, followed by our conclusions in Section 6.

2 Generic knowledge structures in network engineering

The design of a probabilistic network involves three basic steps [3]. First, the relevant statistical variables in the domain are identified, with their possible values. Next, the relations between these variables are specified, resulting in the network's graphical structure. Lastly, probabilities are specified for all variables involved. These tasks are usually performed in close cooperation with domain experts. Recently, we have recommended to develop a conceptual model for a domain of application before actually designing the network's graphical structure

[2]; this model then serves as a documentation of elicited knowledge and as a means of communication during further knowledge acquisition. We have further argued that the conceptual model can be used to derive the graphical structure of the probabilistic network [4]: it is used to derive an initial graphical structure which is subsequently improved and optimised with respect to, for example, the probabilities to be assessed and the running time of inference.

The construction of the conceptual model and its use upon deriving a network structure can be supported by the availability of a library of generic knowledge structures. A generic knowledge structure describes the structure of a unit of knowledge, that is, it describes the relevant concepts and their interrelations independent of a specific application domain. The knowledge engineer first selects a generic knowledge structure that is appropriate for the domain knowledge that is to be modelled. She instantiates the structure with the domain-specific knowledge, thereby obtaining a part of the conceptual model; in addition, she may customise the knowledge structure to meet the requirements of the application at hand. Both the selection of the generic structure and its customisation are based upon a thorough understanding of the knowledge involved. To this end, the various structures are constructed from an in-depth knowledge analysis and represented in depictions and tables that are easy to understand and that are capable of capturing the relevant details of the knowledge involved. The instantiated and possibly customised knowledge structure then is used to derive a segment of the network structure, as indicated above. Since the standard knowledge structure may have been customised and, moreover, the design decisions to be taken in the derivation in some aspects depend on the application at hand, we do not provide a fixed network structure with the generic knowledge structures in the library. Instead, as a guideline, a domain-specific example is associated with each generic knowledge structure, illustrating a step-by-step derivation of a network structure.

The idea of providing generic structures to support network engineering was addressed before [5]. Neil, Fenton and Nielsen introduced the concept of *idiom* for this purpose. Their idioms are modelled directly as segments of a network's graphical structure and are not easily customised without knowledge of the modelling decisions. Our generic knowledge structures, on the other hand, are represented independently of the formalism of probabilistic networks and have associated an in-depth knowledge analysis and example derivation, thus allowing for optimal flexibility in their application.

3 A generic knowledge structure for test results and the underlying truth

Performing tests and interpreting the results play an important role in numerous domains. Tests are typically performed to reveal an underlying true value that is not directly observable. However, due to several aspects, the result of a test may not unambiguously reflect the underlying truth. To enable the knowledge engineer to take well-considered decisions as to whether or not represent these aspects, we designed a generic knowledge structure describing the relations between test results and their underlying true values, based on an in-depth analysis of the knowledge involved. We decided to keep the structure general enough to apply to different fields of biomedicine, yet specific enough to provide valuable support. Further abstraction and adaptation will render it applicable also to other domains. In this section we present the structure; its use to derive the graphical structure of a probabilistic network is discussed in the next section.

Since we illustrate our in-depth analysis of the knowledge related to test results and their underlying true values with examples taken from the domain of oesophageal cancer, we briefly introduce this field of medicine. Due to various factors, a tumour may develop in a patient's oesophagus. Its presence may cause the patient to have difficulties with swallowing food. The extent to which the passage of food is impaired depends on such factors as the tumour's circumference, length and shape. The primary tumour typically invades the oesophageal wall and, upon further growth, may invade neighbouring organs. In time, the tumour may give rise to secondary tumours, or metastases. The depth of invasion and the extent of metastasis, summarised in the cancer's stage, are important factors in deciding upon a therapy. To determine the cancer's stage, typically a number of diagnostic tests are performed. For example, a gastroscopic examination, that is, letting a camera into the oesophagus, is performed to gain insight in such properties of the primary tumour as its circumference. The result of the examination will only be available, however, after the physician has observed and interpreted the gastroscopic image. The result then is an assessment of the tumour's actual circumference.

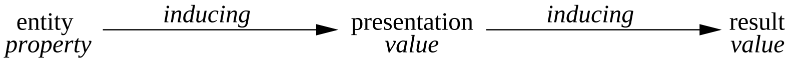

Fig. 1. The relations between entities and test results

We abstract from the example test in our domain of application and describe the concepts and relations involved in more general terms. The tumour in the example is an *entity*. The gastroscopic image is a *presentation* of the tumour's circumference; the physician's assessment of the circumference from the image is a *result*. Entities, presentations and results have properties that may adopt a value. The oesophageal tumour, for example, has a circumference, that may be *circular* or *non-circular*; we represent this type of relation by standard *object-attribute-value* tuples. We now say that a test is performed to gain insight in a property of an entity; (the value of) the property may then *induce* (the value of) a presentation which, in turn, may *induce* (the value of) a result. The relations involved are depicted in Figure 1.

From the above considerations, we have that a property of an entity can only induce a presentation if an appropriate test is performed. The *status* of the test, being *performed* or *not performed*, basically *enables* the relation between

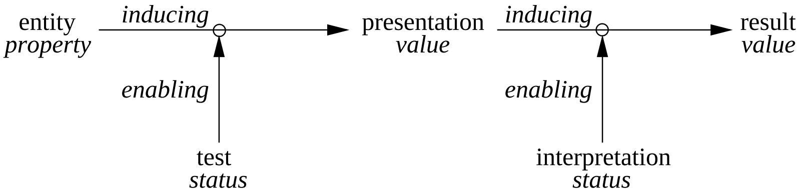

Fig. 2. The roles of test and interpretation

the property and its presentation; this enabling relation is shown in Figure 2 on the left. Now, if the test has been performed, the attribute *value* of the object *presentation* adopts one of its possible values; if the test has not been performed, the attribute cannot adopt any value. Likewise, a physician must interpret the presentation to obtain the test result. The relation between the presentation and the result therefore is *enabled* by the status of the interpretation, as shown in Figure 2 on the right. Again, the attribute *value* of the object *result* can only adopt one of its possible values if the interpretation has been performed. We specify as a validity constraint that an interpretation cannot be performed if the test has not been performed and, hence, no presentation is available.

A test does not always succeed upon performance. For example, if a patient's oesophagus is obstructed, caused by such properties of the tumour as its circumference and length, the camera cannot pass the obstruction upon a gastroscopic examination; the test may then not give the image aimed for. The obstruction is a *manifestation* of the primary tumour and has a certain *degree*. This manifestation may now *disable* (or negatively enable) the enabling relation from the test, as depicted in Figure 3 on the left. The extent to which a test succeeds upon performance depends also on the skills and experience of the laboratory technician performing the test [6]. These properties are summarised in the *test skills* of the technician, shown in Figure 3; note that these skills may vary between

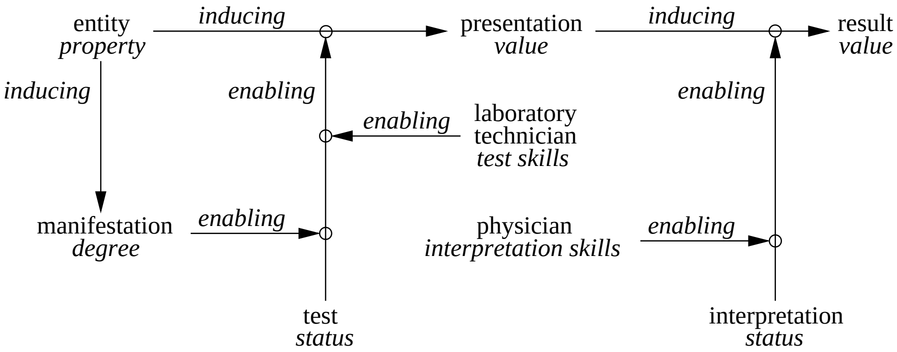

Fig. 3. Relations between test results and the underlying true values

presentation	entity	test	manifestation	laboratory technician
value	*property*	*status*	*degree*	*test skills*
yes, *no*, *indeterminate*	*yes*, *no*	*performed*	*low*, *high*	*low*, *high*
no value	–	*not performed*	–	–

Table 1. The attribute *value* of the object *presentation*

tests. Now, if the test has been performed, the attribute *value* of the object *presentation* adopts, in essence, either one of the values that the attribute *property* of the object *entity* may adopt, or the value *indeterminate*, to indicate that the test has failed. The value it adopts depends on the property of the entity, the degree of the manifestation, and the technician's test skills. These relations are summarised in Table 1. For ease of presentation, we assume that the latter three attributes have two possible values each; the table can easily be generalised to include additional values, however.

Similarly, the extent to which the interpretation gives rise to a result depends on the *interpretation skills* of the physician, shown in Figure 3 on the right. Since performing a test and interpreting a presentation are different tasks that are often performed by different clinicians, the skills involved are modelled as attributes of separate objects. If both tasks are performed by the same clinician, however, the skills can be modelled as separate attributes of the same object. The attribute *value* of the object *result* may now adopt one of the possible values of the attribute *value* of the object *presentation*, or the value *not decided.* The latter value represents that the physician has difficulties in establishing a result, for example, in establishing whether an oesophageal tumour is actually circular or rather almost circular and therefore non-circular. We assume that a physician performing the interpretation recognises whether a presentation is *indeterminate* or not. Table 2 summarises the relations involved. We would like to note that the relations described in the Tables 1 and 2 in general are not deterministic in nature, but involve some uncertainty.

The result of our analysis is the generic knowledge structure in Figure 3 and its associated Tables 1 and 2, describing the relations between test results and their underlying true values in the biomedical domain. It can be instantiated with a specific entity and test, and used in the design of a probabilistic

result	presentation	interpretation	physician
value	*value*	*status*	*interpretation skills*
indeterminate	*indeterminate*	*performed*	–
yes, *no*, *not decided*	*yes*, *no*	*performed*	*low*, *high*
no value	–	*not performed*	–

Table 2. The attribute *value* of the object *result*

network. Upon instantiating the structure, the various concepts may be further differentiated to capture more details. Both clinicians' skills, for example, can be substituted by more fine-grained properties. The structure can also be extended to include additional aspects such as the patient's health status. On the other hand, further abstraction and adaptation will render it applicable also to non-biomedical domains.

4 The derivation of a network structure

The generic knowledge structure capturing the relations between a test result and its underlying true value is used in the design of a probabilistic network or, more specifically, in the derivation of a segment of a network's graphical structure. The graphical structure of a network is an acyclic, directed graph. Its nodes represent statistical variables, which have an exhaustive state space of mutually exclusive, discrete values. The arcs in the graph capture influential relationships between the variables. More formally, the graph represents probabilistic independence: two variables are considered independent given the available evidence if every chain between the two variables contains a variable with at least one emanating arc that has been observed, or a variable with two incoming arcs such that neither the variable itself nor any of its descendants in the graph have been observed.

In deriving a segment of the graphical structure of a network, the selected generic structure is instantiated with domain-specific knowledge and possibly customised to the application at hand. Next, the segment is derived in a sequence of steps. As an example, we illustrate the derivation of a segment for a gastroscopic examination as a test to gain insight in the circumference of an oesophageal tumour. This example derivation serves as a guideline associated with the generic knowledge structure.

The first step is to instantiate the generic knowledge structure with the relevant concepts from the domain of application. In our example, this results in the structure shown in Figure 4. As a gastroscopic examination may result in images of several properties of the tumour, the interpretation and the interpretation skills mentioned in the figure should be taken to pertain to the tumour's circumference.

Building upon the instantiated knowledge structure, the knowledge engineer must now decide upon the statistical variables to be included in the probabilistic network under construction. In our example, the statistical variable *Circumference* is created to capture the attribute *circumference* of the object *oesophageal tumour*. The values of the variable are the possible values of the attribute. The creation of this variable is straightforward, because the attribute from which it is created is single valued and its possible values are mutually exclusive and exhaustive. Therefore, no additional design decisions are required to ensure that the properties of a statistical variable are adhered to. Similarly, the statistical variables *Passage*, *Test-skills*, and *Interpretation-skills* are created.

A knowledge structure may include attributes of objects that do not allow for translation into statistical variables. Examples in our domain are the *status* at-

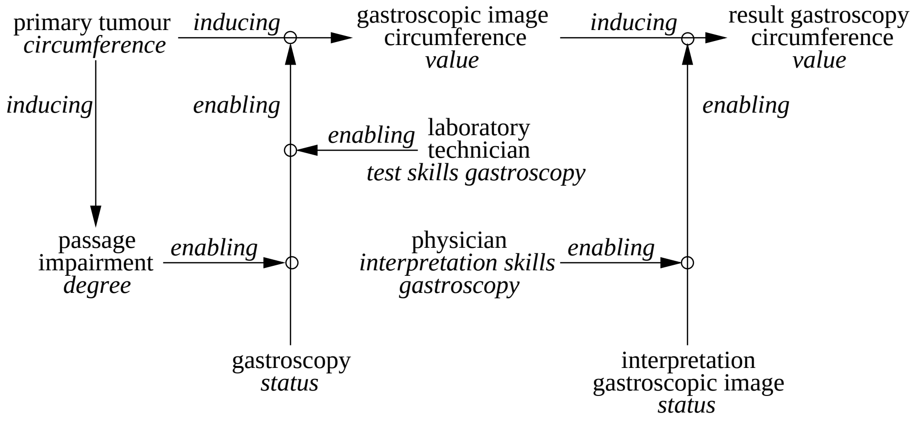

Fig. 4. The instantiated knowledge structure

tributes of the objects *gastroscopy* and *interpretation gastroscopic image*. These attributes are not stochastic in nature, as their values can be decided upon: the attributes in essence are decision variables which cannot be modelled in a probabilistic network. Instead, their meaning is captured by the (simplifying) assumption that *the test and the interpretation have been performed*. Creating statistical variables to describe the remaining attributes is now straightforward. For example, the variable *Gastro-image-circumf* is created, with the possible values *circular*, *non-circular* and *indeterminate*. Note that this variable always adopts a value, in contrast with the attribute *value* of the object *gastroscopic image circumference*, since the network is based on the assumption that the test has been performed.

The next step in deriving the network segment is the specification of the arcs to capture the influences between the statistical variables. The relations between the attributes in the instantiated knowledge structure provide guidance to this end. In our domain of application, for example, the influence from the attribute *circumference* of the object *oesophageal tumour* on the attribute *degree* of *passage impairment* is represented in the network structure as an arc from the variable *Circumference* to the variable *Passage*. The enabling relations from

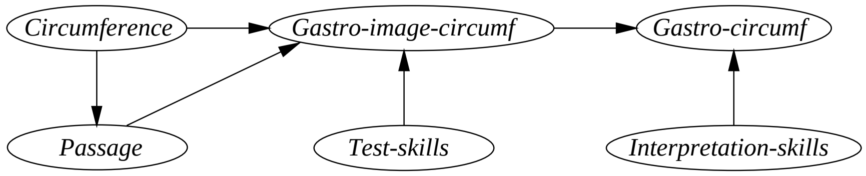

Fig. 5. The initial segment of the graphical structure

the instantiated pattern cannot be represented directly in the graphical structure, however, since a probabilistic network cannot contain an arc pointing onto another arc. However, we observe that in essence these are indirect influences. The enabling relation originating at the attribute *degree* of the object *passage impairment*, for example, constitutes an indirect influence on the attribute *value* of *gastroscopic image circumference*. It is therefore captured in the network by an arc from *Passage* to *Gastro-image-circumf.* The other enabling relations are represented in a similar manner. Figure 5 shows the resulting initial segment.

The segment is then restricted so as to include only variables for which probabilities can be reasonably obtained. The initial segment constructed for our domain, for example, includes the variable *Gastro-image-circumf.* This variable cannot be observed in reality, since an image cannot be observed without being interpreted. Hence, it is practically impossible to obtain the probabilities required for this variable. The variable is therefore removed from the graph, along with its incident arcs, by marginalisation. To retain the indirect influences from the variables *Circumference*, *Passage* and *Test-skills* on *Gastro-circumf*, arcs are added from the former three variables to the latter. Figure 6 shows the optimised segment. It must now still be verified whether or not the resulting segment correctly captures the probabilistic independences in the application domain. The structure in Figure 6 states, for example, that the variables *Passage* and *Test-skills* are independent, yet may become dependent given a value of *Gastro-circumf.* These (in)dependences actually hold in the domain. As the same observation pertains to the other represented (in)dependences, no correction of the segment is necessary.

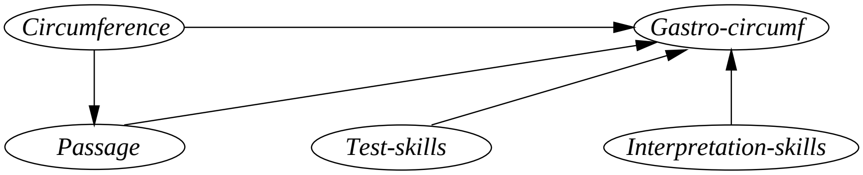

Fig. 6. The optimised segment

The optimised segment can be further reduced by making some (simplifying) assumptions about, for example, the clinicians' test or interpretation skills. For example, if the probabilistic network is being developed for a specific hospital in which all clinicians involved have comparable skills in performing the gastroscopy and in interpreting the gastroscopic images with respect to the tumour's circumference, then the variables *Test-skills* and *Interpretation-skills* can be removed from the structure, along with their emanating arcs. Removing these variables again amounts to marginalising over them, this time without giving rise to additional arcs. Note, however, that upon assessing the probabilities for the variable

Gastro-circumf, the influences from the removed variables should be taken in account. The reduced segment is shown in Figure 7.

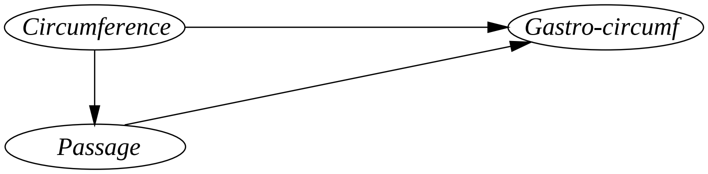

Fig. 7. The reduced segment

We have demonstrated how our generic knowledge structure pertaining to test results and their underlying true values can be used for deriving a segment of the graphical structure of a network for our domain of application. The example derivation is associated with our generic knowledge structure, and serves as a guideline for deriving segments pertaining to other biomedical domains. We would like to note that the resulting segment distinguishes explicitly between test results and hidden true values. This contrasts the approach taken in most probabilistic networks. Including just the variable that represents the test result into a network, amounts to assuming that all modelled tests unambiguously reveal the underlying truth and therefore are perfectly reliable. Experience has shown that this assumption may rarely be realistic, however [7]. In addition, by including just the variable that represents the test result, the modelled independence relation may be inaccurate, especially if the true value itself plays a role in other parts of the network.

5 Related work

We are not the first to propose the use of a library of structures or patterns for engineering purposes. A structure for capturing tests, more specifically, has also been proposed by Neil, Fenton and Nielsen [5]. In their *measurement idiom*, represented in the formalism of probabilistic networks, however, all concepts inducing uncertainty are represented by a single statistical variable. Our generic knowledge structure offers much more detail. We feel that as a consequence it can offer more detailed support. The application of general *analysis patterns* has also been proposed in the context of business modelling and software engineering [8]. The analysis pattern proposed for *observations and measurements* in the medical domain is rather general, however, and does not distinguish explicitly between true values and test results. We feel that therefore the pattern does not provide much support in modelling the detailed knowledge involved in tests and their results. The generic knowledge structure that we propose, in contrast, explicitly represents the various sources of uncertainty. The use of reusable knowledge structures is also common in the field of knowledge engineering [9]. Both these structures and the analysis patterns are meant for support of the construction of a model, independently of specific knowledge representation formalisms or programming languages. They are therefore rather general. Our approach, on the other hand, is tailored to network engineering, and therefore provides more dedicated support. Our knowledge structures allow for flexible use, yet their practical application in network engineering is supported by example derivations that serve as a guideline.

6 Conclusions

In this paper we have proposed to support the task of engineering a probabilistic network by the availability of a library of generic knowledge structures. A generic knowledge structure captures the structure of a unit of knowledge independently of a specific domain. It is instantiated by the knowledge engineer with domain-specific knowledge and used to derive a segment of the graphical structure of a probabilistic network. As the design decisions within this derivation may depend on the requirements of the application at hand, an example derivation is associated with each generic knowledge structure to serve as a guideline. To enable the knowledge engineer to customise the represented knowledge to the requirements of the application at hand, the generic knowledge structure reflects the intricate details of the knowledge involved, thus facilitating well-considered adaptations. In addition to providing support of the process of knowledge modelling, our generic knowledge structures may also serve to guide the elicitation of the required knowledge. To provide for a generic knowledge structure containing knowledge pertaining to test results and their underlying true values, we have performed an in-depth analysis of the concepts, relations and uncertainties involved. We presented both the resulting knowledge structure and an example derivation of a network structure, in the domain of oesophageal cancer. For detailing the associated probabilities, we would like to refer to [10].

To investigate the applicability of our generic knowledge structure for test results and their underlying true value, we studied the real-life probabilistic network that we had previously developed for the field of oesophageal cancer. This network includes some 25 variables that model test results and, hence, contains many segments involving tests. Each of these segments proved to be based on knowledge that fits our generic knowledge structure. We are currently applying the structure in the development of a probabilistic network for the detection of classical swine fever in pig herds. We found so far that the use of our generic knowledge structure significantly simplifies the task of the knowledge engineer.

In the near future, we aim to develop a variety of generic knowledge structures along the same ideas outlined in the paper, to arrive at an extensive library of knowledge structures serving to support the engineering of probabilistic networks in a wide variety of application domains.

References

1. F.V. Jensen (2001). *Bayesian Networks and Decision Graphs.* Statistics for Engineering and Information Science. Springer-Verlag, New York.
2. E.M. Helsper, L.C. van der Gaag (2002). A case study in ontologies for probabilistic networks. In: M. Bramer, F. Coenen, A. Preece (eds). *Research and Development in Intelligent Systems XVIII.* Springer-Verlag, London, pp. 229-242.
3. M.J. Druzdzel, L.C. van der Gaag (2000). Building Bayesian networks: "Where do the numbers come from?" Guest editors' introduction. *IEEE Transactions on Knowledge and Data Engineering*, vol. 12, pp. 481-486.
4. E.M. Helsper, L.C. van der Gaag (2002). Building Bayesian networks through ontologies. In: F. van Harmelen (ed). *Proceedings of the 15th European Conference on Artificial Intelligence.* IOS Press, Amsterdam, pp. 680-684.
5. M. Neil, N. Fenton, L. Nielsen (2000). Building large-scale Bayesian networks. *The Knowledge Engineering Review*, vol. 15(3), pp. 257-284.
6. R.H. Fletcher, S.W. Fletcher, E.H. Wagner (1996). *Clinical Epidemiology: the Essentials.* 3rd edn. Williams & Wilkins, Baltimore.
7. L.C. van der Gaag, C.L.M. Witteman, S. Renooij, M. Egmont-Petersen (2001). The effects of disregarding test characteristics in probabilistic networks. In: S. Quaglini, P. Barahona, S. Andreassen (eds). *Artificial Intelligence in Medicine.* LNAI 2101. Springer-Verlag, Berlin, pp. 188-198.
8. M. Fowler (1997). *Analysis Patterns: Reusable Object Models.* Addison Wesley Longman, Menlo Park.
9. G. Schreiber, H. Akkermans, A. Anjewierden, R. de Hoog, N. Shadbolt, W. Van de Velde, B. Wielinga (2000). *Knowledge Engineering and Management: The CommonKADS Methodology.* MIT Press, Cambridge, Massachusetts.
10. D. Sent, L.C. van der Gaag (2003). Detailing test characteristics for probabilistic networks. In: M. Dojat, E. Keravnou, P. Barahona (eds). *Artificial Intelligence in Medicine.* LNAI 2780. Springer-Verlag, Berlin, pp. 254-263.