Library-style ontologies to support varying model views

Hermi J.M. Tabachneck-Schijf and **Linda C. van der Gaag**
Department of Information and Computing Sciences, Utrecht University,
P.O. Box 80.089, 3508 TB Utrecht, The Netherlands
e-mail: {hermi,linda}@cs.uu.nl

Abstract

The next development in building Bayesian networks will most likely entail constructing multi-purpose models that can be employed for varying tasks and by different types of user. We argue that the development of an ontology to organize the knowledge needed for such a multi-purpose model is crucial for the management of the model's content. This ontology should preserve all elicited knowledge and be accessible to both domain experts and knowledge engineers. Based on the different ways in which people learn and gain expertise, we further argue that knowledge elicitation will result in task-specific knowledge mostly, although some task-neutral knowledge will emerge as well. To support varying model views, this combination of knowledge is best stored in a library-style ontology of task-specific and task-neutral modules.

1 Introduction

While in the early years of the field of Bayesian networks attention focused primarily on algorithmic issues, the last decade has seen an increasing interest in methods to support the construction of such networks. The field also has become more and more experienced in building decision-support systems that include a Bayesian network. Bayesian networks by now have evolved beyond laboratory settings and are being employed by non-academic users. In turn, users of these network-based decision-support systems are starting to see the possibilities that these systems offer, and begin to ask for more. For example, for various of our biomedical applications, we have been asked whether we could perhaps adapt the model for teaching purposes. It is therefore likely that the next development in the field of Bayesian networks will entail building multi-purpose models which can be employed for varying tasks and, in all likelihood, by varying types of user.

In this paper we argue that to support model views for varying tasks, a suite of Bayesian networks should be built rather than a single network. We further argue that in the first step of developing such a suite, knowledge elicitation will necessarily result in task-specific information mostly, although also some task-neutral knowledge may emerge. Structuring the elicited knowledge into a library-style ontology of task-specific and task-neutral modules then is best suited to empower reuse of knowledge segments and to facilitate composition of model views. We reiterate our view that this ontology should capture all elicited knowledge and be accessible to domain experts and engineers alike.

We begin by defining different types of model view in Section 2, and outline the *task model view* under discussion in the current paper. We argue that a single multi-purpose model would quickly become too large and unyieldy to afford the knowledge engineers and the domain experts an overview of its contents. We therefore advocate building a *suite of models* to support multiple task model views, rather than a single Bayesian network.

In Section 3 we outline our view of ontologies. We rationalize why an ontology should be constructed of the elicited knowledge, before actually developing a suite of Bayesian networks. This rationalization is much in line with our earlier arguments for developing ontologies for single networks [9]. The ontology provides as a well-structured documentation of all elicited knowledge and includes also any background information that is not captured explicitly in a network. This background information supports, for example, viewing the elicited knowledge from different perspectives, as required for different tasks. The well-structured documentation then scaffolds the building of different task model views for a suite of Bayesian networks.

We are not the first to suggest the use of ontologies. Ontologies are being developed for a variety of purposes, ranging from providing a portal for the semantic web to documenting elicited knowledge for the development of knowledge-based models; see for example [4, 6, 8, 18]. For many of these purposes, a rigorously formal logic-based or other mathematical ontology language is used to allow for automated processing. For our purpose of supporting the development of a suite of networks by well-structured documentation, however, the ontology should provide as a medium for communication between the engineers and the experts involved in the suite's construction. Based upon the observation that a rigorously formal language is not easily accessed by non-mathematical experts, we advocate, in Section 3, the use a less formal language for our ontologies.

We address the knowledge content of our ontologies in Section 4. In order to align the content of our ontologies with elicited knowledge, we consider the processes by which humans learn and structure their own knowledge. We observe that the elicited professional knowledge of practicing experts is mostly both task- and domain-specific, although also some task-neutral information may emerge during knowledge elicitation.

In Section 5, we argue that knowledge is best stored in the fashion in which it is obtained from the experts. We further argue that the elicited knowledge is best organized into *modules*. An organization of knowledge in modules is well suited for storing task-specific knowledge to support multiple tasks. Organizing the modules in a *library-style ontology* further encourages reuse of the knowledge elicited for one task model view for the construction of another task model view. We would like to note that in our earlier work we proposed the development of a meta-library of generic knowledge structures complemented with example network derivations [11]. To support the evolvement of an ontology for a suite of Bayesian networks, such generic knowledge structures can guide and speed up entering knowledge into the various modules.

The paper ends with a discussion and some perspectives for further elaboration of the presented ideas to a practicable knowledge-engineering approach to developing multipurpose Bayesian networks.

2 Model views of Bayesian networks

We distinguish two types of model view, namely *task model views* and *interaction model views*. To explain the difference between the two types, we distinguish three different states in the development of a suite of models. The first state consists of a stored pool of knowledge relevant to all tasks to be carried out. The second state encompasses the actual suite of models that allows computations to be carried out for the various tasks. The third state comprises concrete means that allow users to work with the suite of models. In view of these three states, we also consider the steps that need to be taken to proceed from one to the next state. The first step reaches the first state and involves eliciting and structuring knowledge. The second step necessitates first selecting, from the pool of all elicited knowledge, the knowledge that determines the content and the structure of the suite of models to be developed, and then representing this knowledge in the mathematical formalism of Bayesian networks. The final step is characterized by designing interfaces to the suite of models, that is, the different ways the models can be presented to someone interacting with it, be this an engineer or an end-user.

We consider a *task model view* to be one view of a suite of models. The task model view is the result of carrying out the elicitation and structuring of task-neutral and task-related domain knowledge and of making selections of the elicited knowledge to support a single or a few closely related tasks. In the medical field, for example, one task model view might support diagnostic reasoning, while another task model view could support teaching diagnostics, which requires additional modeling of underlying mechanisms so that deeper 'why' and 'what if' questions can be posed and answered. *Interaction model views*, on the other hand, comprise the interfaces of a model that are tailored to task and user. For example, for a diagnostics model view, one interaction model view could be optimized for data entry and another might support maintenance of the model by the knowledge engineer.

In sum, for different tasks to be carried out by different types of user, a suite of models can require several task model views, each of which can need several interaction model views. In last year's workshop, we laid out some methods to construct effective interaction model views [16]. In the current paper, we concentrate on the elicitation and structuring of knowledge, in order to support the development of multiple task model views.

3 Ontologies for Bayesian networks

A suite of Bayesian networks that supports several tasks with different task model views, is likely to be of a complexity necessitating development over multiple years, involving possibly different engineers and experts. Building and maintaining models of such complexity is a hard and time-consuming process. The knowledge elicited from domain experts constitutes a rich pool of knowledge, segments of which can play varying roles in the domain under study. All this elicited knowledge has to be carefully reviewed and structured, and ultimately captured in the formalism of Bayesian networks. In this process, a multitude of modeling decisions are taken as well as numerous decisions to demarcate the scope of the model. Such decisions tend to forestall an overview and thorough comprehension of the model by anyone who has not been intimately involved in its construction. We have experienced already for single larger networks, that construction and maintenance are seriously hampered if the elicited domain knowledge and the decisions taken are not made explicit by proper documentation [9]. This problem is bound to grow worse if a suite of networks is to be developed and maintained.

Having observed the advantages of developing an ontology before building a single Bayesian network in our earlier work [9], we feel that the construction of a suite of models will especially benefit from an explicit ontology, which then serves not just as a documentation of all elicited knowledge but also as a means of ensuring consistency over the models within the suite and as a medium for communication between the experts and engineers involved.

3.1 The role of ontologies

Most generally applicable knowledge-engineering methodologies, among which is the well-known CommonKADS methodology [13], strongly recommend the development of a conceptual model before actually constructing a model in the knowledge-representation formalism to be used. In line with this recommendation, we recently proposed to develop an ontology before constructing a Bayesian network for a domain at hand [9].

There exist many views of the concept of ontology in general; see for example [4, 7, 8, 18]. In this paper, we use the term ontology to refer to an explicit specification of the elicited domain knowledge that is to be shared by the experts and the knowledge engineers involved in a network's construction and maintenance. From this perspective, an ontology plays two distinct roles. One of these is to make all elicited domain knowledge explicit. To this end, the ontology specifies not just the knowledge that is to be captured in a network, but also the relevant background knowledge of the domain and the meta-level knowledge of its regularities and organizational structure. Note that capturing the elicited knowledge directly in a Bayesian network would result in a representation from which not all types of domain knowledge are easily recognizable as a result of the modeling decisions taken. Also, some of the elicited knowledge may not be captured at all in the network. The other main role of an ontology is to provide as an explicit medium for communication between experts and engineers alike for further knowledge acquisition, network validation and maintenance.

3.2 The ontology language and an example

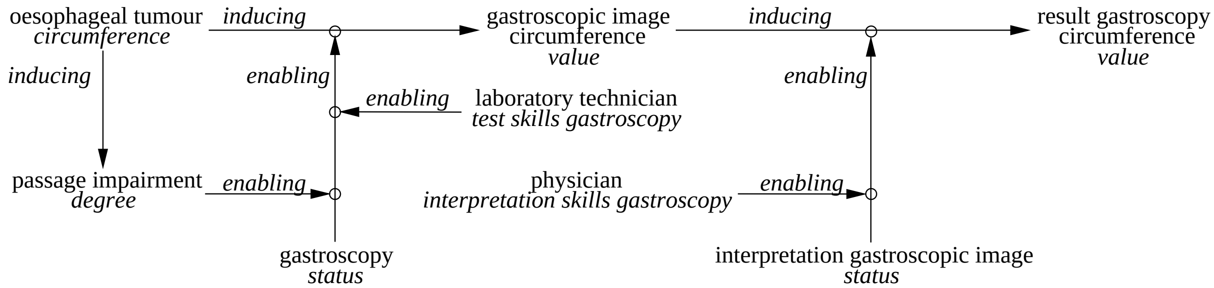
To support the two roles mentioned above, the representation language to be used for an ontology should be chosen with care. The issue of selecting an appropriate ontology language has been addressed by many researchers. Some suggest that domain knowledge should be represented by a language that is highly informal, semi-informal, or semi-formal [18]; others argue that ontologies should be specified in a rigorously formal language and, in fact, should be machine readable [14].

An important argument for using a formal ontology language is that it allows a highly structured and unambiguous representation of the elicited knowledge. Such a formal representation in addition may provide for (semi-)automated derivation of segments of the Bayesian networks under construction. While rigorously formal languages often have limited expressiveness, an ontology language should come with a rich semantics to introduce as little bias as possible in the represented contents. If the language introduces biases, for example as a result of not allowing the representation of specific knowledge constructs, then the ontology may not properly reflect the intricacies of the domain. Since the ontology is to be used for the construction of a network, the resulting model may then be biased as well, maybe even in unforeseen ways. The development of an independent knowledge model, recommended by most knowledge-engineering methodologies, in fact has its origin in this observation.

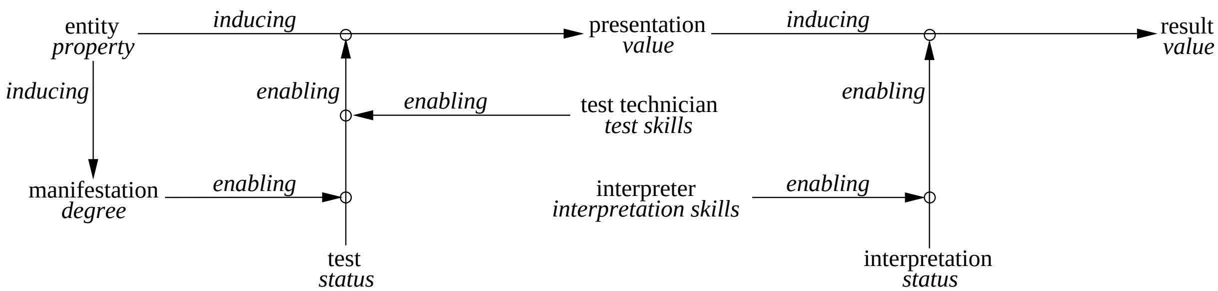
The purpose of knowledge sharing provides a strong argument for using a less formal language. The ontology should be represented in a language that is understandable for both the knowledge engineers and the domain experts involved in a network's construction. We argued before that the mathematical language of Bayesian networks, for example, is very difficult to grasp by non-mathematical persons [16]. In our opinion in fact, many of the formal languages commonly used for ontologies are unsuitable for checking the accumulated knowledge with non-mathematical experts. If the use of a formal language is uncommon in a domain of application, then a rigorously formal language is unsuited for the purpose of knowledge sharing between the knowledge engineers and the domain experts in the domain at hand and a less formal language had best be used.

To support developing Bayesian networks in the biomedical domain, we use a semi-formal ontology language composed of well-structured tables, depictions, graphs and hierarchy representations combined with text [9], which can be understood by both the domain experts and the knowledge engineers. As an example, Figure 1(a) shows part of an ontology for the medical domain of oesophageal cancer. The depicted graph captures the relationships between the result of a gastroscopic examination of the circumference of a patient's tumour and the underlying true circumference. It describes, for example, that a gastroscopic examination may not result in an image from which the circumference can be established, as a result of a patient's impaired swallowing capabilities.

Upon establishing the stage of a patient's cancer, not only the circumference of the primary tumour is investigated. Other diagnostic tests are performed as well. In addition to the knowledge pertaining to these tests separately, the domain's ontology specifies the high-level regularities of the knowledge involved. The graph capturing these regularities for the various diagnostic tests is depicted in Figure 1(b). Note that this graph can be exploited upon extending the network with the results of a new test, as it provides for guiding the elicitation of the knowledge pertaining to



(a)



(b)

Figure 1: Relations between test results and the underlying true values, (a) for a gastroscopic examination of the circumference of an oesophageal tumour, and (b) for a diagnostic test in oncology in general

the new test. For further details of the oesophageal cancer ontology, we refer to [9].

3.3 Ontology-supported construction of networks

Of course it is a daunting prospect to have to capture all elicited knowledge in two ways, that is, first in an ontology and then in a suite of Bayesian networks. A carefully structured ontology, however, can be used to derive the graphical structure of the suite in a semi-automated fashion. First, the knowledge that is to be captured in the suite is selected from the ontology; the remainder of the ontology then serves as background knowledge to the suite. Note that this step involves a reflection on the elicited knowledge which must be performed and documented by the knowledge engineer. In the next step, the central concepts and relations from the selected parts of the ontology are combined into a single depiction for each envisioned network. From this depiction, an initial graphical structure is derived that adheres to the syntax of Bayesian networks. To this end, the domain concepts from the depiction are translated into stochastic variables, which may involve for example re-defining multi-valued variables. The relations from the depiction are translated into arcs between variables in the initial graphical structure. Note that many of these steps can be performed in an automated way. Figure 2 shows, as an example, part of the initial graphical structure that is derived from the graph of Figure 1(a). In the final step, the engineer has to verify that the resulting structure correctly captures probabilistic independence. Also, the initial structure may need further optimization [10].

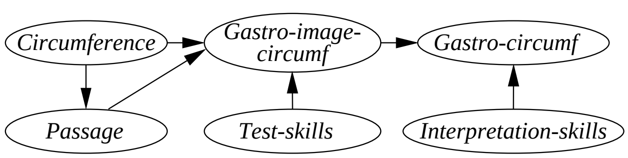


Figure 2: The initial segment of the graphical structure

4 Eliciting ontology knowledge

Given the prospective advantages of constructing a domain ontology before building a suite of Bayesian networks, we now turn to the question of how to organize the elicited knowledge in the ontology so that it most usefully supports different task model views for the suite.

Many researchers recommend that ontologies be constructed independently of the projected use of the ontology and its contents; see for example [3]. Underlying this recommendation is the argument that any commitment to the problem-solving method that will be applied to the domain knowledge for example, will influence and thereby bias the contents of the ontology. Such commitments thus hamper the extendibility and reuse of the ontology. However, constructing an ontology without any commitments to a particular task requires either eliciting task-neutral knowledge from domain experts, or stripping the task-specific aspects from the elicited knowledge. In this section, we address the feasibility of the first option; the second option is briefly addressed in Section 5.

We consider eliciting task-neutral information, that is, eliciting knowledge from experts without them having a particular task in mind. To provide task-neutral information, experts should be able to gather such information from their minds, which implies that the knowledge should be stored in their brains in such a way that task-neutral aspects are readily separated from task-specific aspects. We now briefly lay out the different ways in which people learn information and argue that these learning processes imply that the knowledge stored in the human brain is largely both domain- and task-specific. We then conclude that, given how knowledge is learned and stored, it would be extremely difficult to elicit task-neutral knowledge from an experienced professional.

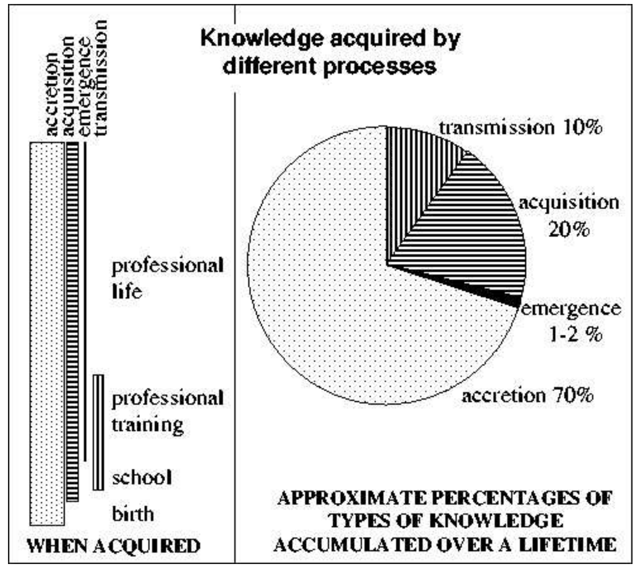


Figure 3: Knowledge acquired by different processes

4.1 Human knowledge acquisition processes

Humans acquire knowledge in four different ways: transmission, acquisition, accretion, and emergence [19]. Usually people start gathering professional knowledge from books and teachers: the knowledge is explicitly *transmitted* to them. Except in vocational training, such transmitted knowledge is mostly task-neutral. Over the course of a lifetime, transmission accounts for some 10% of our knowledge. Further learning done by conscious choice is termed *acquisition learning*, which is good for about 20% of our knowledge. Acquired knowledge is gathered by our own initiative: by exploring, experimenting, self-instruction, inquiry and the like. *Emergence* is the result of self-constructing new ideas and meanings that did not exist before, which in current educational practices is said to account for just 1-2% of our knowledge.

When people are asked to describe learning processes, they generally mention explicit processes akin to transmission and acquisition, and perhaps emergence. *Accretion*, which accounts for about 70% of what we know, however, does not commonly come to mind. Accretion is the gradual, unconscious and implicit process by which we learn for example language, culture, social behavior, and whatever other knowledge comes on our path. Accretion knowledge is picked up simply by living and interacting with the world. Within limits, we process and react to all we see, hear, smell, taste and experience. By processing the information and reacting to it, it is stored in the brain without our being conscious of the learning process. People consequently often are not even aware they possess this type of knowledge. Because it is unconsciously experienced and learned in particular situations, accreted knowledge is largely both task- and domain-specific. Figure 3 summarizes the four processes by which humans acquire knowledge.

4.2 Example: the acquisition of medical knowledge

While the four learning processes reviewed above relate to general educational practices, they are easily mapped onto what happens in the course of gathering professional knowledge. Although the exact percentages may vary a little, the different processes will create roughly the same proportions of the knowledge that our domain experts possess. We illustrate this observation with an example from medicine [2], and also argue that transmission and acquisition learning in college does not prepare a student for medical practice, because of the task-neutral nature of the material learned in medical school.

The basics for medical knowledge are taught by *transmission* in universities. This type of knowledge is explicitly task-neutral and consists of biomedical knowledge, which is mostly causal and definitional in nature and describes the functioning and possible dysfunctioning of the human body. It is this transmitted knowledge that upon elicitation would result in task-neutral knowledge segments.

Next, students are confronted with patients in internships, where they have to link the transmitted task-neutral information to clinical knowledge. In contrast to biomedical knowledge, clinical knowledge is task-specific in nature. It consists of knowledge of symptoms, classification and treatment of diseases, all embedded in medical situations. In internships, some transmitted information is still offered, but students are also *acquiring* knowledge by trying to figure out diagnoses and treatment plans themselves. *Accre-*

tion then is also at work, continually recording knowledge from all perception instruments. Examples of accreted knowledge are how to read symptoms from patients' look, smell, utterances and behavior, and how to communicate with colleagues, patients and their next of kin, yet also how to get around in the hospital and many other aspects of work. All that is learned is now embedded in the task at hand and in the medical culture and practices. In cognitive-science terms, the knowledge is *situated*.

It is taking the step from employing task-neutral knowledge in college to having to apply task-specific knowledge in a hospital setting that makes the transition from the university classrooms to practice so problematic for many medical students [2]. Students may have learned which disease causes which symptoms, and maybe even have seen pictures of such symptoms. However, recognizing the symptoms when exhibited by a patient is a very different matter. Each patient is unique, and may or may not exhibit all of the symptoms. Patients also may exhibit symptoms differently. Patients may further have more than one disease, which may result in an indistinct mixture of symptoms. Last but not least, the reasoning required now goes diagnostically from symptoms to disease, not causally from disease to symptoms. The difficulty of this re-representation is supported by research in various other contexts, from which it is also clear that switching information from one representation to another is very difficult. Switching representations, in fact, does not occur spontaneously and must be explicitly and extensively taught [1, 17].

Professional learning in medicine does not stop with the internship phase. It continues by a mixture of accretion and acquisition during the entire professional career. All knowledge picked up in this phase is in a task-specific format, because it is learned while carrying out specific tasks. The theory of situated learning describes this phenomenon and argues that learning as it normally occurs is a function of the activity, context and culture in which it occurs [12, 15]. In fact, the theory argues specifically that learning *never* occurs in a task-, context-, and culture-neutral manner.[1] In a physician, for example, interaction with patients is typically stored as examplars of sick people complete with diagnosis, treatment plan, and outcomes.

From the above observations, we conclude that the bulk of the professional knowledge of an expert is stored in the mind in a task-specific format.

4.3 Eliciting task-specific knowledge

Since professional knowledge is largely task-specific, it is reasonable to assume that most of the knowledge that comes to the fore upon elicitation is task- and domain-specific. Of course an engineer can explicitly ask a domain expert to provide task-neutral knowledge. If experience from practice is requested, however, the engineer is asking for extra information processing from the expert: the expert has to relate his or her knowledge in a different way than is stored in the brain. This, as argued in the example above of the medical students' transition from book knowledge to diagnostic and treatment knowledge, requires non-trivial effort, which, as it is to be done real-time, will at least considerably slow down the elicitation. More potentially damaging, however, asking people to relay knowledge in a way that requires them to reason about their stored knowledge, as is done when asking an expert for task-neutral information, always increases the risk of introducing errors [5]. We conclude that, except for information that was transmitted in a task-neutral fashion, it will be difficult, time-consuming and error-prone to try to elicit task-neutral knowledge from domain experts.

Two examples from our own research will serve as illustrations. As a first example, when we asked veterinarians to supply us with average disease symptoms for pigs that were sick, most of them provided us with symptoms belonging to one particular illness rather than a context-free average; some gave symptoms associated with a particular group of closely related diseases such as infections of the respiratory tract. What happened is that the veterinarians called a pig having a particular disease to mind, of which they provided the symptoms. The veterinarians providing a few more symptoms ostensibly generalized but actually were doing exactly what their colleagues did: they provided the symptoms of diseases encountered within the same differential diagnosis. The veterinarians unwittingly rendered their knowledge in the same situated way it was stored, rather than following our instructions.

As a second example, we relate a knowledge-elicitation session where we asked a group of veterinary experts to reason out loud about particular pig cases of which the clinical symptoms were described in terms of variables and values. When asked what would happen to their assessment of the case when a particular symptom was changed from present to absent, one of the participants asked, in earnest, how he could possibly change the symptoms of a pig. Clearly, the veterinary expert had called the case to mind as a concrete pig for which he had to come to a diagnosis. Thinking in this task-related setting, he could not imagine physically changing a pig's symptoms.

5 Storing the elicited knowledge

Having established that it will be rather unlikely that an engineer will elicit knowledge from a domain expert that is altogether task-neutral, we now address how the elicited

[1] According to this theory, the knowledge transmitted in medical school is also not task-neutral: the task is passing the exam. For our purpose, however, the issue is that the knowledge is independent of specific medical tasks.

knowledge is best stored in an ontology. More specifically, we compare constructing a single task-neutral ontology that is free of task biases, with constructing multiple task-specific ontologies. We then argue that a library-style ontology best supports the development of a suite of Bayesian networks for multiple tasks. This library-style ontology is composed of various modules that are task-specific as well as domain-specific, supplemented with modules that are either task-neutral or domain-neutral.

5.1 Single or multiple ontologies

We begin by comparing capturing all elicited knowledge in a single task-neutral ontology or in multiple task-specific ontologies. For the construction of a *single* ontology, be it composed of task-neutral or task-specific knowledge, plead that no duplication is needed and that it will be easier to ensure internal consistency upon maintenance and extension. In spite of these advantages, however, we reject building a single ontology. A single ontology is likely to become quite large in size for a suite of Bayesian networks supporting multiple task model views. Even if it is well organized and highly structured, its mere size will cause the knowledge engineers and the domain experts to quickly lose track of its contents. Another argument against the construction of a single ontology is that it may be much more difficult to build multiple task model views from a single entity than from a collection of task-focused entities.

Having rejected developing a single ontology, we now address the format of the ontology's content. There are quite strong arguments for storing knowledge in a *task-neutral* fashion. Task-neutral knowledge need not be captured multiple times for use for varying tasks, as would be required if the knowledge were captured in a task-specific fashion. Also, when new task model views need be developed, it is likely that these can already be supported using the available task-neutral knowledge. If the knowledge would have been stored in a task-specific fashion, developing a new task-specific ontology would be required.

Although there are strong arguments for storing the elicited knowledge in a task-neutral fashion, it generally will be highly infeasible to do so. In Section 4, we argued that the bulk of the elicited knowledge will be available in a format that is both task- and domain-specific. Constructing a task-neutral ontology would thus require stripping the elicited knowledge from its task biases and integrating the resulting segments of neutral knowledge. The task of stripping the elicited knowledge from its task-related context is non-trivial, however. Our opinion in fact is that it is infeasible since not just the experts but also the engineers will have particular tasks in mind when surveying the various segments of knowledge. The engineers moreover are likely to be insufficiently knowledgeable in the domain of application to recognize the various task biases included.

From the above observations, we conclude that although storing knowledge in a task-neutral fashion is prefered, it is infeasible to do so for the bulk of elicited information. Some of the elicited knowledge may be available as task-neutral information, however, for example if originating from the transmission phase of learning professional knowledge. Also, some of the elicited information can be abstracted to segments of task-neutral knowledge. An example from our veterinary applications pertains to the stress effects of handling a pig. Catching a pig will cause stress to the animal, regardless of the task for which it is being caught. The knowledge elicited in the contexts of the various tasks thus is explicitly reusable and can be stored in a task-neutral fashion.

5.2 A library of ontology modules

Alternative to either a single task-neutral ontology or a collection of multiple task-specific ontologies as discussed above, is a *library* consisting of multiple *ontology modules*. Some of the library's modules contain background knowledge that is common to all tasks in the domain under study yet independent of a specific task. Other modules contain knowledge that is common to one task but holds across domains; the graph from Figure 1(b), in fact, showed a segment of such knowledge, pertaining to the interpretation of the results of diagnostic tests in biomedicine. The majority of the modules, however, capture knowledge that is both task- and domain-specific. A segment of knowledge may thus be captured in more than one module, described from the varied perspectives of different tasks. A task-specific ontology aimed at supporting a particular task model view, then is constructed by combining various modules.

We illustrate the concept of a library-style ontology using our earlier example in medicine. A library of modules for medical applications would include, for example, anatomical knowledge. Anatomical knowledge is descriptive and definitional in nature and summarizes the elements of the human body. Anatomical knowledge is common to most medical tasks yet is independent of any specific task. In the library, it would therefore be included in one or more task-neutral ontology modules. Knowledge of which diseases typically occur in the differential diagnoses of which other diseases is closely linked to the task of diagnosis, and would be included in a task-specific ontology module for diagnostic tasks. Note that gradations of task specificity may be supported. Knowledge of the relationships between diseases and symptoms, for example, is common to both diagnosis and prognostication, and could be included in a single ontology module subserving both tasks.

To construct a concrete task-specific ontology for supporting a model view of teaching diagnostics, information from the task-neutral modules of anatomical knowledge would be pulled in as well as information from modules related to

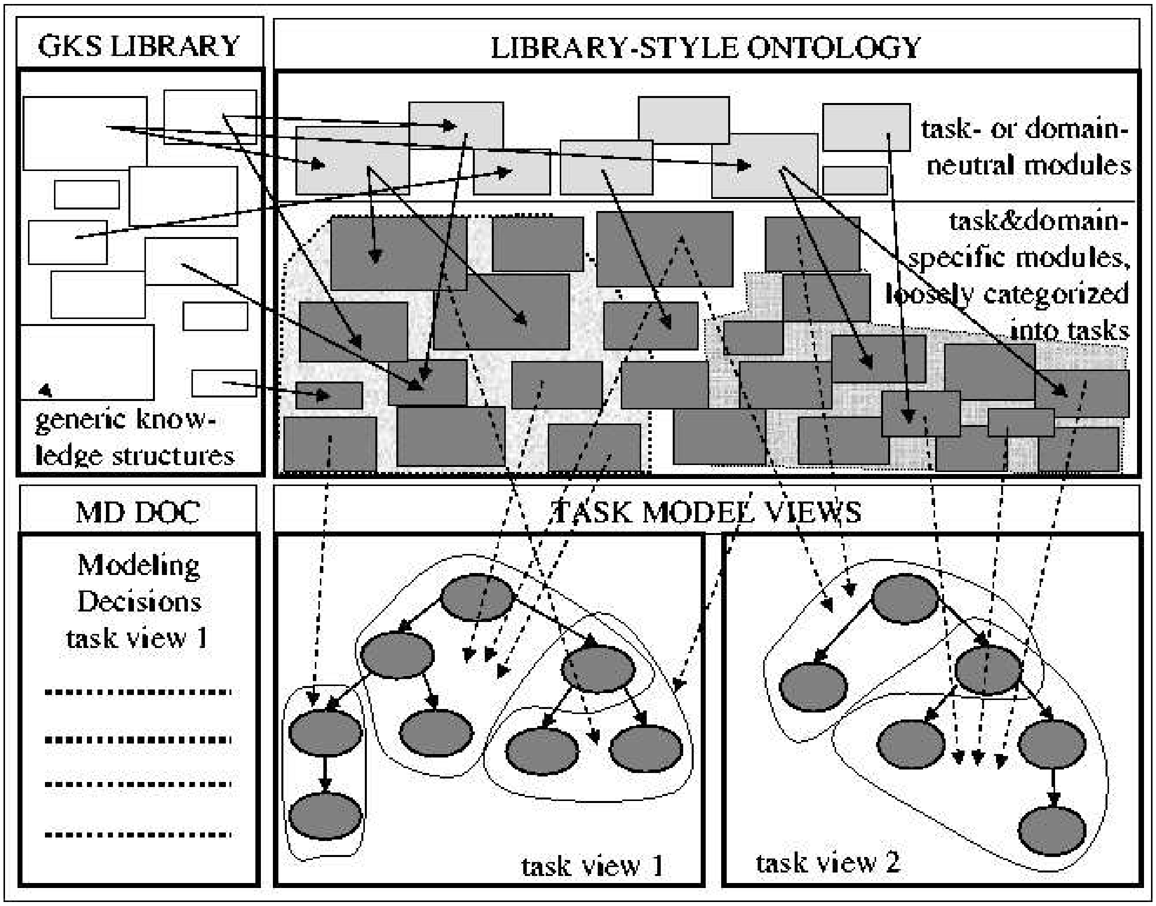


Figure 4: A library-style ontology for developing task model views: the library of ontology modules is supplemented with a library of generic knowledge structures and a document of modeling decisions; drawn arcs indicate instantiation of modules, dashed arcs indicate selection

the tasks of diagnosis and prognostication. The modules of anatomy and prognostication would then subserve simulation purposes and answering in-depth 'what-if' questions. Note that the other, unrelated modules of the library need not be considered upon constructing the task-specific ontology. For supporting a model view of diagnosis, on the other hand, the knowledge from the task-neutral modules of anatomy would most likely not be included explicitly in the task-specific ontology, as the model to be developed could leave this knowledge implicit. Now suppose that an ontology for the new task model view of predicting the effects of treatment is to be developed. Any task-neutral knowledge required for the new model view ideally is already present in the library and can be readily pulled in. Also the ontology module of prognostication, which is already present in the library, captures some of the knowledge for the new task and can be used. In addition, however, a new task-specific module needs to be developed and included in the library. The knowledge for this new module, describing the physiological effects of treatment, is elicited from domain experts, focusing on just the task at hand.

6 Concluding observations

In this paper, we argued that multiple task model views for Bayesian networks are best supported by a library-style ontology composed mainly of task-specific knowledge modules, but also including task-neutral modules.

In summary, this paper addressed several issues. We began by reiterating the need for documenting all elicited knowledge. If this knowledge is not properly documented, construction and maintenance of large suites of networks inevitably becomes problematic. We recommended building an ontology to provide a well-structured explicit specification of the elicited knowledge and a medium for communication for the knowledge engineers and the experts involved in the networks' development. We argued that the ontology should not only store the knowledge needed for the different model views, but also any relevant background knowledge; in addition, a modeling-decisions document should be maintained. Documentation of the information that cannot be read off the suite of networks directly is especially important when the development of the suite extends over several years of research and the suite ultimately is handed off to industry.

The paper also attended to the language to be used for our ontologies. The necessity of including all types of relevant knowledge demands a language that allows for a rich semantics and permits semi-automated model building. We stressed that the language used should be accessible for non-mathematical domain experts. Earlier research had shown that rigorously formal representations, be they logic-based or stated in another mathematical language, cannot readily be understood by domain experts who are not trained in such representations. When stated in a semi-formal language that is accessible for the experts, the ontology can provide as a means of communication between the knowledge engineers and the experts, which serves to minimize the risk of omitting important information and of including erroneous information.

Next, we pled for aligning the content of the ontology with how practicing experts learn and store knowledge in their minds. Some knowledge, we argued, is stored in a task-neutral fashion, and should also be stored in this way in the ontology. However, we contended that most knowledge of domain experts is inherently related to specific tasks and is stored in that way in their brains. Constructing a task-neutral ontology would thus require stripping the task-specific professional knowledge from its task biases. This, however, is highly demanding, either on the part of the expert or on the part of the knowledge engineer, and error-prone. We therefore proposed storing task-specific knowledge in a task-specific fashion.

Lastly, we proposed to develop a library-style ontology, composed of the aforementioned task-neutral and task-specific knowledge modules which subsequently are combined into task-specific ontologies to support concrete task model views for a suite of Bayesian networks. We illustrated the ease of development of multiple views and demonstrated that reuse of information is encouraged by organizing the domain knowledge in modules.

In the near future, we intend to further develop our concept of ontology library by using it in the development of a suite of Bayesian networks in the field of veterinary science. By doing so, we hope to initiate a publicly available collection of ontology modules and inspire the uncertainty community to contribute.

References

[1] H.P.A. Boshuizen, H.J.M. Tabachneck-Schijf (1998). Problem solving with multiple representations by multiple and single agents; an analysis of the issues involved. In: M.W. van Someren, P. Reimann, H.P.A. Boshuizen, T. de Jong (eds). *Learning with Multiple Representations*. Amsterdam: Elsevier Science, Ch. 8, pp. 137 – 152.

[2] H.P.A. Boshuizen, M.W.J. van de Wiel (1998). One person, multiple representations: An analysis of a simple, realistic multiple representation learning task. In: M.W. van Someren, P. Reimann, H.P.A. Boshuizen, T. de Jong (eds). *Learning with Multiple Representations*. Amsterdam: Elsevier Science, Ch. 12, pp. 237 – 263.

[3] T. Bylander, B. Chandrasekaran (1988). Generic tasks for knowledge-based reasoning: the 'right' level of abstraction for knowledge acquisition. In: B.R. Gaines, J.H. Boose (eds). *Knowledge Acquisition for Knowledge-based Systems*, vol. 1. Academic Press, London, pp. 65 – 77.

[4] P.C.G. da Costa, K.B. Laskey, K.J. Laskey (2005). PR-OWL: A Bayesian ontology language for the Semantic Web, *International Semantic Web Conference, Workshop Uncertainty Reasoning for the Semantic Web*, Galway, pp. 23-33.

[5] K.A. Ericsson, H.A. Simon (1993). *Protocol Analysis: Verbal Reports as Data*. MIT Press, Cambridge, MA.

[6] T.R. Gruber (1993). A translation approach to portable ontologies. *Knowledge Acquisition*, vol. 5, pp. 199 – 220.

[7] Th.R. Gruber (1995). Towards principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies*, vol. 43, pp. 907 – 928.

[8] G. van Heijst, A.Th. Schreiber, B.J. Wielinga (1997). Using explicit ontologies in KBS development. *International Journal of Human-Computer Studies* vol. 46, pp. 183 – 292.

[9] E.M. Helsper, L.C. van der Gaag (2002a). 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.

[10] E.M. Helsper, L.C. van der Gaag (2002b). Building Bayesian networks through ontologies. In: F. van Harmelen (editor). *Proceedings of the 15th European Conference on Artificial Intelligence*. IOS Press, Amsterdam, pp. 680 – 684.

[11] E.M. Helsper, L.C. van der Gaag (2005). Generic knowledge structures for probabilistic-network engineering. *Proceedings of the Third Bayesian Modeling Applications Workshop*, held in conjunction with the

Twenty-first Conference on Uncertainty in Artificial Intelligence, Edinburgh.

[12] J. Lave (1988). *Cognition in Practice: Mind, mathematics, and culture in everyday life*. Cambridge, UK: Cambridge University Press.

[13] 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.

[14] R. Studer, V.R. Benjamins, D. Fensel (1998). Knowledge engineering: principles and methods. *Data & Knowledge Engineering*, vol. 25, pp. 161–197.

[15] L. Suchman (1988). *Plans and Situated Actions: The Problem of Human/Machine Communication*. Cambridge, UK: Cambridge University Press.

[16] H.J.M. Tabachneck-Schijf, P.L. Geenen (2006). Preventing knowledge transfer errors: probabilistic decision support systems through the users' eyes. In: L.C. van der Gaag, R. Almond (eds). *Proceedings of the 4th Bayesian Modelling Workshop: Bayesian Models Meet Cognition*, Boston, pp. 74 – 82.

[17] H.J.M. Tabachneck-Schijf and H.A. Simon (1996). Alternative representations of instructional material. In: D. Peterson (editor). *Alternative Representations: an Interdisciplinary Theme in Cognitive Science*. London, England: Intellect Books, pp. 28 – 46.

[18] M. Uschold, M. Gruninger (1996). Ontologies: principles, methods and applications. *The Knowledge Engineering Review*, vol. 11, pp. 93 – 136.

[19] L.O. Wilson (1997). *New View of Learning: Types of Learning*. Retrieved Jan 25, 2007, from `www.uwsp.edu/education/lwilson/learning`