

In God We Trust



Course Report

Presentation Project

Advanced Artificial Intelligence

By: Arman Didandeh

Student NO.: 874105

Course Tutor: Dr. Mehzen Afsharchi

Semester: Fall 2008

IASBS, Gava Zang

Zanjan, Iran

Preface

As the main approach of our course, "Advanced AI" is springing in this semester around uncertainty and specifically on Bayesian Networks; I've decided to choose some related paper to work on. As I started to meet some topics in relation with BN and my previous studies, I met the concept "Ontology", which really made my mind. Some more efforts on these topics guided me to the matter of integrating these two in some applications. Therefore I searched more and more and at last found some recent papers with the above-told pattern in them.

The main paper I'm dealing with is titled as "COSTRUCTING BAYESIAN NETWORKS AUTOMATICALLY USING ONTOLOGIES"—which is called the main paper from now on-. It has been published in the Applied Ontology 2006, IOS Press, by Ann Devitt, Boris Danev and Katarina Matusikova from Ericsson Research Center.

Here in this paper, the authors tried to represent an approach to utilize the knowledge and inference capabilities which ontology can provide them, in order to automate the construction of some graphical structure on the domain of interest. While this structure is dealing with uncertainty and especially probabilistic inference in the future (posterior probability reasoning) it should be met under the name of a BN.

I am going to let an introduction at first, then explain the previous efforts of the integration matter, then talk mostly on the main paper. After these, the information which I gathered by studying some other related papers and books will help us understand the progress of the main paper better. A little critic and some ideas of my own will come at the end.

Introduction

The main concepts which I correspond to in this report are "Ontology" and "Bayesian Networks". Here I provide a simple introductory section on the concept of ontology and while the BN is talked out in the course, the BN introductory would be just as a quick view.

Graphical Models, Bayesian Networks

Graphical Models are told to be a marriage between probability and graph theory, which provide a natural tool for dealing with "uncertainty" and "complexity". As Michael Jordan mentioned in 1998: "GM formalism provides a natural framework for the design of new systems"

Any GM could be in the shape (representation) of these three types:

1. Undirected GMs: which we refer to as Markov Random Fields or Markov Networks;
2. Directed GMs: which we refer to as Bayesian or Belief Networks;
3. Chain Graphs: which are classes of hybrid GMs.

The most interest here is on the second class, which we know as BN in future.

Some most important matters of talk are listed in coming:

- ✓ Inference: using Baye's rule to compute the posterior probability of each explanation.
- ✓ Explaining Away: sometimes marginally independent causes are conditionally dependent over the model. It's equivalent to Berkson's paradox (Selection Bias).

- ✓ Reasoning: any reasoning in a BN could be modeled in two ways. These are:
 - Diagnostic / Bottom-Up / Effect to Cause
 - Causal / Top-Down / Cause to Effect
- ✓ Influence Diagrams: generalizations of BNs that could represent and solve decision problems under uncertain conditions.
- ✓ Dynamic BNs: are to model sequences of variables, i.e. speech signals.
- ✓ Common Inference Methods: mostly go in below three classes:
 - Variable Elimination
 - Clique Tree Propagation
 - Recursive Conditioning
- ✓ Common Approximate Inference Methods: mostly go in below two classes:
 - MCMC (Markov Chain Monte Carlo)
 - MBE (Mini Bucket Elimination)
- ✓ Applications of BN: Bayesian networks are used for modeling knowledge in Bioinformatics, Medicine, Document Classification, Information Retrieval, Image Processing, Data Fusion, Decision Support Systems, Engineering, Law and very many other fields of utilization.
- ✓ Temporal Models: mostly go in below three classes:
 - HMMs (Hidden Markov Models)
 - LDSs (Linear Dynamical Systems) known by Kalman filters
 - BAT (Bayesian Automated Taxi)

Bayesian AI

The ultimate goal of BAI is "to create a thinking agent which does as well or even better than human in reasoning tasks, which can adapt to stochastic and changing environments, recognize its limited knowledge and cope sensibly with these varied sources of uncertainty" as it's being told in "Bayesian AI, 2004, Korb & Nicholson, Chapman & Hall/ CRC, page 21".

Ontology

Ontology is a formal representation of a set of concepts within a domain and the relationships between those concepts. It's used to reason about the properties of that domain and may be used to define the domain. Tom Gruber from Stanford has mentioned ontology as: "a description of the concepts and relationships that can exist for an agent or a group of agents"

In Computer Science, we are concerned with establishing fixed and controlled vocabularies, in which the ontology concept may be a great help to us. Therefore a set of large and robust ontologies were introduced in 20th century like WordNet or CYC.

Common components of ontologies are listed below:

- ✓ Individuals
- ✓ Classes
- ✓ Attributes
- ✓ Relations
- ✓ Function Terms
- ✓ Restrictions
- ✓ Rules
- ✓ Axioms

- ✓ Events

An ontology could go into two categories:

1. Extensional (ISO 15926): the ones who are characterized solely by their membership;
2. Intentional: which mostly are to put distinctions between classes and conditions to permit decision-making.

There is sometimes a concept dealing in the matter, mostly know as a Partition. Partition is a set of related classes and associated rules that allow objects to be classified by the appropriated subclass (a similarity to fork can be distinguished here). Partitions go into two categories:

- ✓ Disjoint Partition (XOR)
- ✓ Exhaustive Partition (OR)

The most known relations in the ontology applications are:

1. Taxonomy: Partially Ordered Sets
2. Meronymy: DAGs

Another categorization of ontologies which is more common and more important comes as below:

1. Domain Ontology: which represents the particular meaning of terms as they apply to the domain;
2. Upper Ontology: which is a model of common objects applicable across a wide range of domain ontologies in addition to a core glossary;
3. Hybrid: which a good example could be known as Gellish ontology.

Several ontology languages have been prepared during time to be used as a formal language to encode the ontologies, such as: CYC, Gellish, IDEF5, KIF, RIF, and OWL.

After this overview on ontology, here we go somehow deeper in the concept of using ontologies in BN construction. A more precise and full knowledge on ontologies is prepared and represented in Appendix I.

The Besides Paper Discussion

Some previous efforts on the tasks use using ontologies to help constructing the BN structure of a domain led to some papers. I've been studying some of them and as a good example, the below paper is going to be discussed:

BUILDING BAYESIAN NETWORKS THROUGH ONTOLOGIES; Helsper & van der Gaag, 2002

In this paper –which is called the besides paper from now on-, the authors used ontologies to facilitate the building of BN in the medical domain. However ontologies were used only as means of knowledge representation and the creation of BNs were done manual.

Here I provide an overview of this paper as an introductory for using ontologies and then I'll discuss the main paper for we have general idea of what it's going to do.

Usage of Ontologies to construct BNs

Building a knowledge-base system is mostly complex for real-life application domains and knowledge has to be extracted from domain experts or data. The first is usually not available in short time and it has a great deal of expense. The second while having some even standard algorithms and approaches of parameter estimation is not so optimized to be reliable and is also a NP problem at least (on-line data extraction from live feeds). Some of these algorithms to derive CPTs for a fixed network structure from data are being discussed in the below papers –as introduced in the main paper- and I'll refer to them as parameter estimation algorithms in the main paper discussion:

- ✓ BAYESIAN APPROACHES TO MULTIPLE SOURCES OF EVIDENCE AND UNCERTAINTY IN COMPLEX COST-EFFECTIVENESS MODELLING; Spiegelhalter & Best, 2002
- ✓ A BAYESIAN METHOD FOR THE INDUCTION OF PROBABILISTIC NETWORKS FROM DATA; Cooper & Herskovitz, 1992
- ✓ STOCHASTIC RELAXATION, GIBBS DISTRIBUTION & THE BAYESIAN RESTORATION OF IMAGES; Geman & Geman, 1984 (revised in 2002)
- ✓ MAXIMUM LIKELIHOOD FROM INCOMPLETE DATA VIA THE EM ALGORITHMS; Dempster & Laird & Rubin, 1977

Along with the knowledge elicitation, some formalism is needed to capture this knowledge, where BN is making itself more favorite. Building a network involves three basic tasks:

1. Identifying variables and their possible values;
2. Identifying the relations between these variables and encoding them in a graphical structure;
3. Obtaining the CPT.

Building network-based systems in comparison with knowledge-base systems involves specific modeling issues, i.e. modeling domain concepts as statistical variables; so some tailored methodologies are needed to be prepared.

The approach in the besides paper is like this:

The knowledge of the domain is modeled into an ontology. Then a sequence of issues gets a BN structure. At the end, a qualification is done by means of some available probability elicitation methods.

Domain ontology is constructed with expert knowledge plus a real-life BN (generated with no use of any methodology). This will lead to various BNs / alternatives from the ontology.

A quick view on the domain of interest lets us know that the main problem springing around esophagus cancer and its metastasis. The network includes some 40 variables and about 1000 probabilities.

An overview on the methodology: most methodologies need domain knowledge capture which may lead to a knowledge-base creation and a knowledge model is needed to prevent inconsistencies which hamper system's construction and maintenance. For exploiting such a model, ontology could be used for knowledge specifying, while all the knowledge is to be captured in a network plus the available metadata and the background knowledge.

The process of (ontology \rightarrow graphical structure + quantification) is done in two phases:

1. Directly relevant knowledge for network is selected and the remainder is a background knowledge. Now the translation to statistical variables can be started.
2. Then the generated structure would go into improvement and optimization, where the structure is restricted to include variables for which probability can be obtained (expert / data). This may lead to some added / reversed arcs, and at least to a suboptimal structure.

To guarantee a feasible running time of probabilistic inference, the number of incoming arcs may have to be reduced. This task is done via removing the weak dependencies or by divorcing parents. This is done to reduce the state space of some variables.

The optimization phase would be a trade-off between rich network and time cost of inference. Therefore the several alternatives are to be pruned during the process of their generation, as much as possible.

The Ontology of Esophagus specifies the explicitly captured knowledge in the system as well as more implicit background knowledge and the meta-level knowledge of its regularities. It consists of several components to specify different levels of abstraction and perspectives. Here we use a glossary in addition to a static perspective component and a dynamic perspective component. Below comes an image of a simplified fragment of the hierarchy of this ontology:

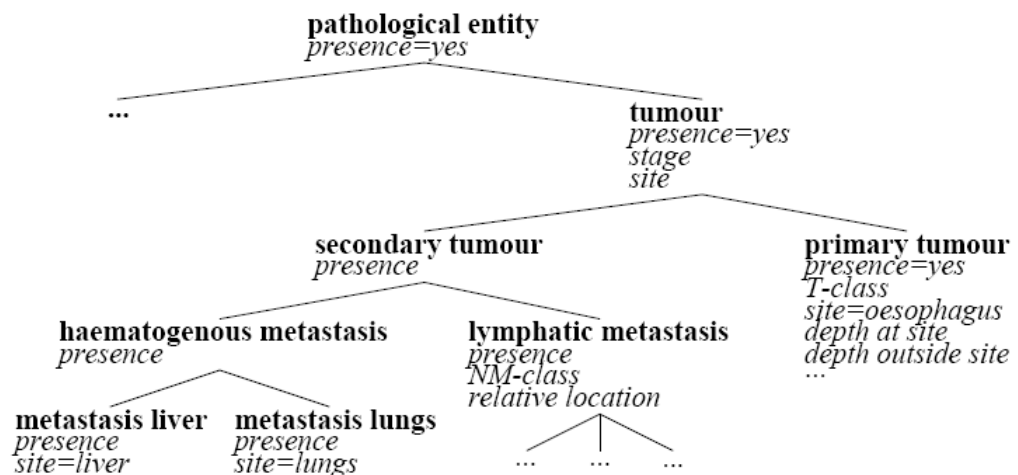


Figure 1: A fragment of the hierarchy of pathological entities

Also in the below image we can see some of the relations:

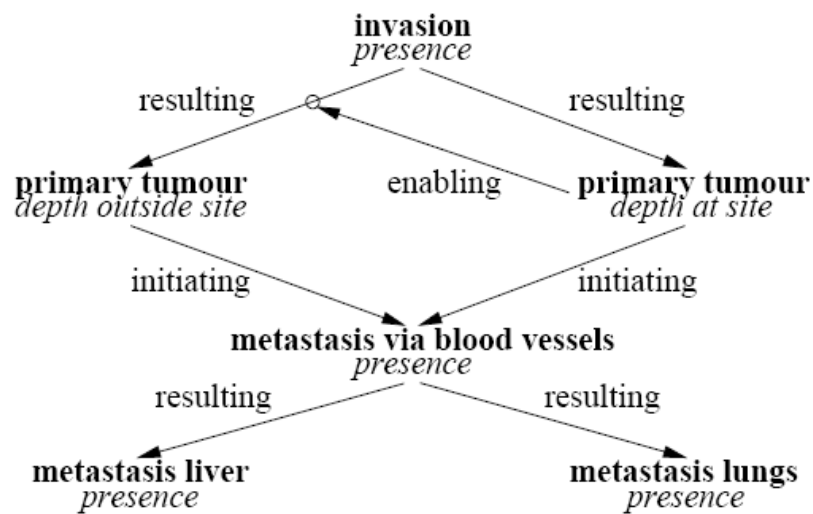


Figure 2: A fragment of the attribute-level graph of dynamic relations

In order to derivate the structure, we start with selecting the knowledge, we'd like to address from our ontology. The central concepts and relations of this knowledge are then represented in several depictions.

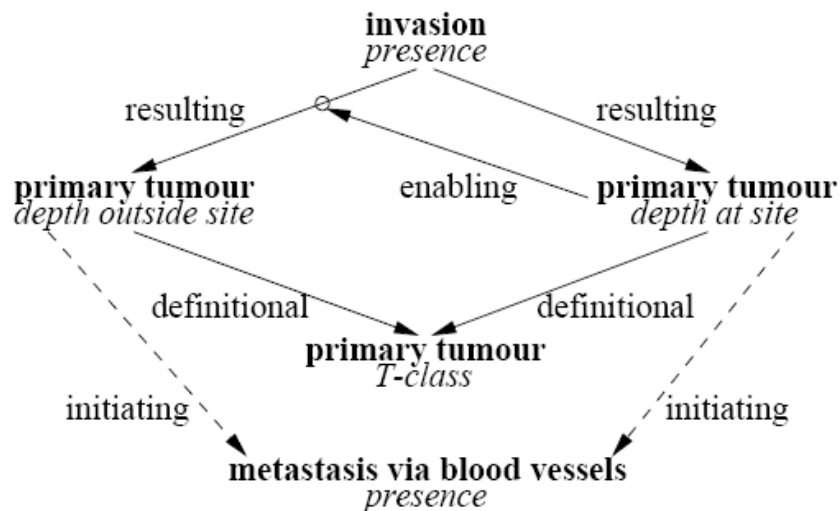


Figure 3: The depiction for the depth of invasion

The depictions describe the knowledge at attribute-level which provides a convenient point of departure for deriving an initial structure. Further the depiction combines the relations from different components of our ontology. This may lead to several derivations.

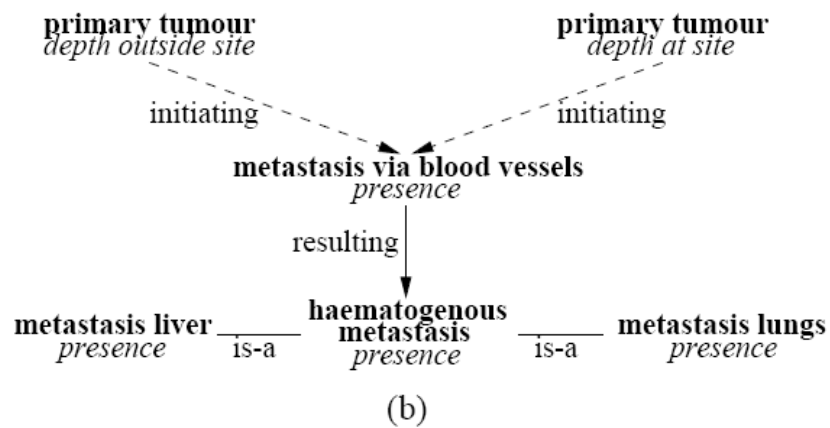
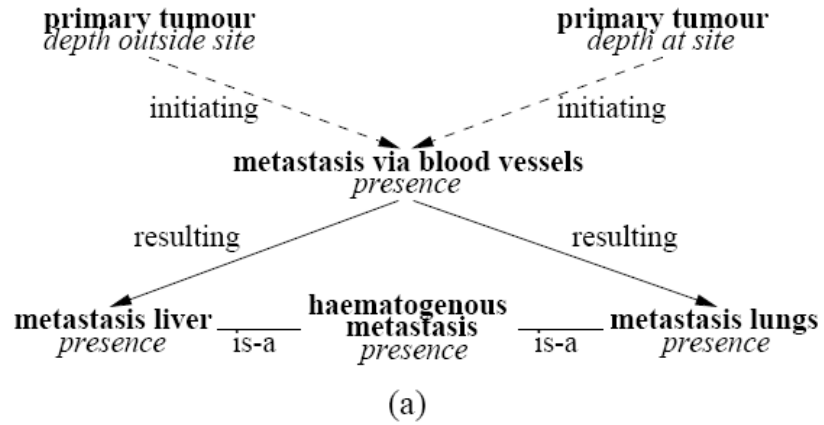


Figure 4: Alternative depictions for haematogenous metastasis

These alternative depictions may lead to several alternative structures, as shown in the below image:

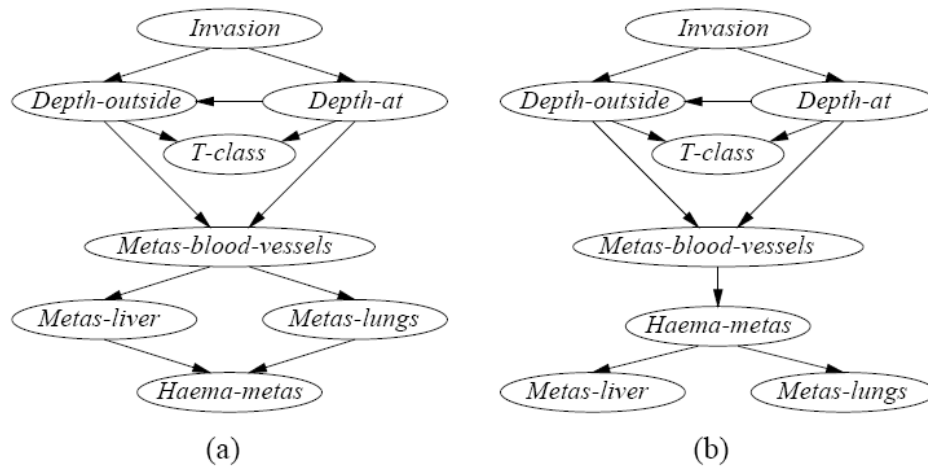


Figure 5: The alternative initial structures

These derived structures are to get restricted, as shown in the below image:

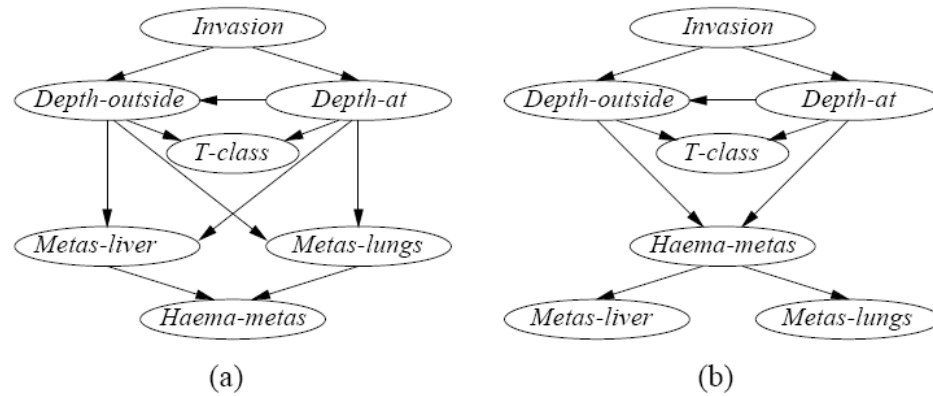


Figure 6: The alternative restricted structures

Then it's time for optimization:

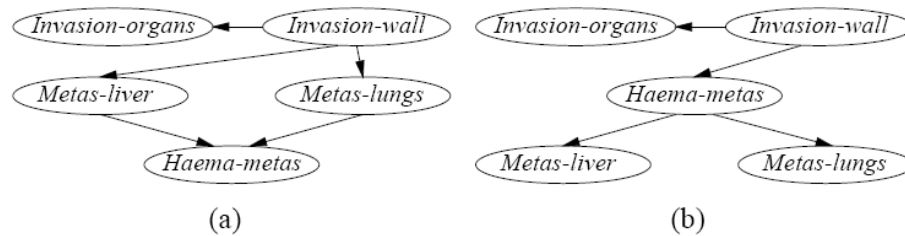


Figure 7: The alternative optimized structures

To decide about the final structure, we consider the nature of the inaccuracies in the representation of independence. It'd be better to check the independence as early as possible then repeat it upon optimization phase. Usually it's been offered to make a feedback loop of optimization where the final structures go into some expert-analysis capsule and then get another optimization phase. This may lead to rich optimized networks to deal among the domain of interest.

The Main Paper Discussion

As told before, the main task of this paper is to present an approach to automate the construction of BNs on a domain of interest by means of some in-hand ontologies on the domain. This approach is to harness the knowledge and inference capabilities inherent in an ontology model to do its task and to accurately represent the domain. This approach is implemented on the context of an adaptive self-configuring network management system in the telecommunications domain. The ontology model in this approach is used as a knowledge repository as well as a facilitator of automated workflows. The generated BN serves to monitor the effects of management activities, forming part of a feedback loop for self-configuring decisions.

The structure of this main paper is like below:

We first provide a sample BN for the domain of interest to get a familiarity with the domain. Then the discussion goes around utilizing the structure and inference capabilities of ontologies for the automation task. Hence, the implementation of the

methodology on the above told domain is provided. After all, the conclusion and future work of the main paper authors are discussed.

Some sample BN structure for the domain of telecommunication is shown in the image below:

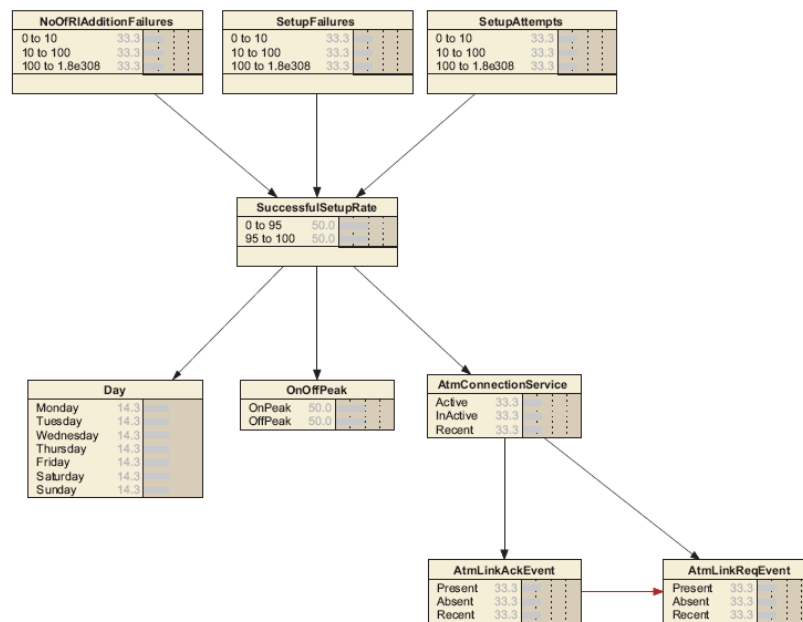


Figure 8: Sample Bayesian Network for the

Telecommunication Network Management Domain

I talked before about the task of building the structure and assigning the probability distribution of a BN and its complexity. I also said that all the phases of this task can be done with the help of three below facilities:

1. Expert knowledge
2. Offline learning from historical data
3. Online learning from a live feed of data

In the phases of a BN generation, two of them are most important and of most interest among the people who study it. One is the CPT derivation, which I mentioned some algorithms in the besides paper. The other is learning the causal structure of the BN. There are also some known algorithms in this subfield too:

- ✓ K2: Cooper & Herskovitz, 1991;
- ✓ MDL (Minimum Description Length): Lam & Bacchus, 1993;
- ✓ CAMML: Wallace et al., 1996.

The learning algorithms for the structure may remove some bottlenecks but they also have their drawbacks:

- ✓ Over-simplification on assumptions about the data (input or output);
- ✓ Inability to deal with missing data;
- ✓ Requirement for huge amount of input data;
- ✓ Intractability for multivariate input data;
- ✓ Waste of resources for integrating multiple existing data sources.

Here, ontologies are to provide a knowledge source which may help reducing these problems as a formal representation. This is the way to automate the BN-from-Ontology process.

Automating BN Construction

There are two difficulties regarding the BN construction:

1. Complexity of the hand-coded BN structure, which needs both BN and domain knowledge;
2. Integration of existing knowledge sources.

The algorithm discussed here relies wholly on expert knowledge rather than evidence-extraction. While this seems to be no progress, it can be seen that the method simplifies the expert work. It's while the need for mastering the principles of BNs is eliminated and the expert just needs to classify the domain knowledge in a familiar ontological world.

Subtasks of this construction task is listed here and discussed further:

1. Identifying the variables of interest;
2. Specifying the values that these variables can take;
3. Defining the relations between these variables (Finding parent nodes);
4. Assigning the CPD.

Identifying the variables of interest

Here we need the assumption that an ontology of the concepts of the domain of interest has been in hand or can be built. Not all of these concepts are liked to take a role in the BN to model the causal relations. To recognize them, a new ontology of the interested concepts is generated and linked to the domain ontology. All of these concepts are then descendents of a node in the new BN ontology, called the BNNode. To give birth to any concept –leaves classes- there are two prominent relations to debate:

- ✓ hasParentNode: a directed link to a least one node;
- ✓ hasDelayParentNode: a directed time-delay link, used to generate a Dynamic BN (a BN with temporal dimension)

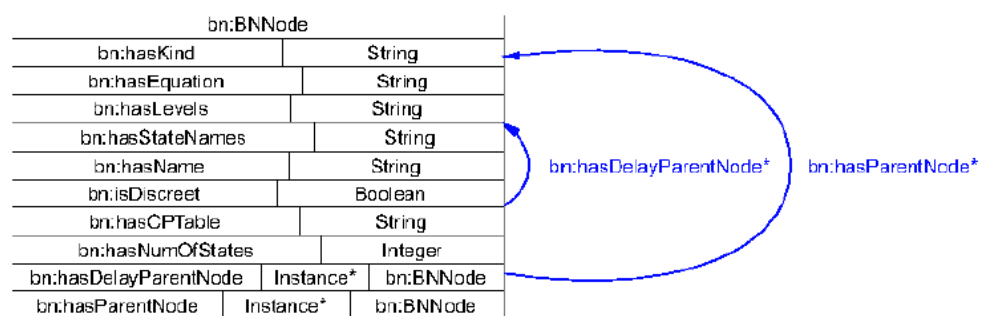


Figure 9: BN node concept

When a node is instantiated, attributes are extracted from properties of the main concept in the domain ontology. Here, the inheritance relation which is defined below helps to automate the node creation and their attributes and arcs. The domain ontology and this BN ontology are used in combination to force the BN creation.

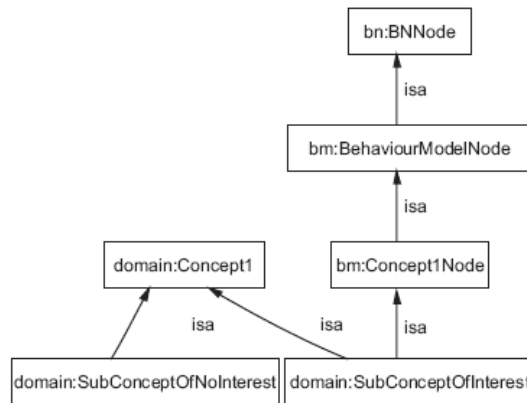


Figure 10: Generic Domain Ontology with BN concepts

There also can be an additional level. The ontology can define a hierarchy of more specific BNNode classes for any domain concept which should be included in the output BN. It may overcome to a tailored processing of domain ontology concepts. An example could be setting ranges or default values for some variables.

Specifying the values that these variables can take

The attributes of the created nodes should be derived from combined BNNode and the domain ontology. These attributes may contain name, state names, type of BN nodes and etc. This is done using the restrictions and constraints offered by the ontology. These restrictions are specified by the domain concepts:

```

<restriction>
  <onProperty hasStateNames/>
  <hasValue "Present"/>
  <hasValue "Absent"/>
  <hasValue "Recent"/>
</restriction>

```

Other restrictions are used to control correct node notations:

```

<hasProperty class="Name" cardinality=1/>

```

Inheritance makes the restrictions to be provided in a generic level. Therefore, the ontology reasoner may use this to get all restrictions and generate the BNNode with properties:

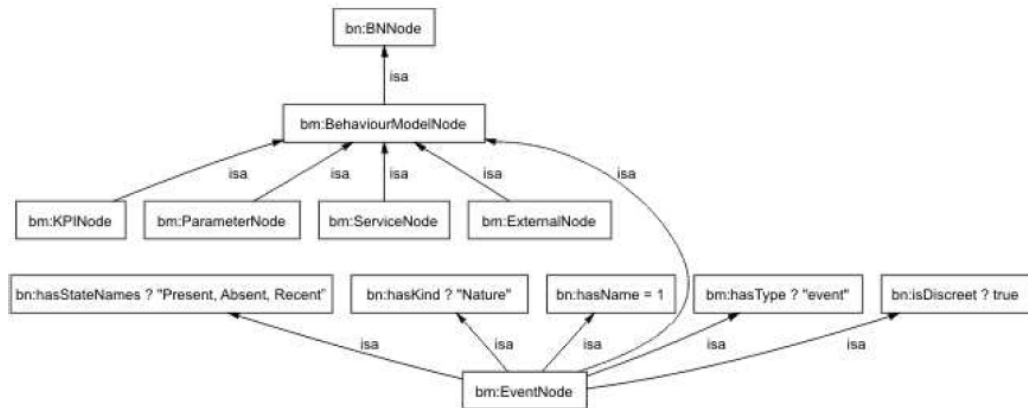


Figure 11: Restrictions on EVENT node in TC ontology

Defining the relations between these variables / Finding parent nodes

Arc creation relies on rules which are specific to the application domain and define which ontology properties or relations between concepts correspond to arcs in BN. An example is that any disease can have several symptoms. This can be presented as a class of disease and a rule to express its causal relation to symptom classes.

This rule-based approach leads to a means of specifying generic BN relationships which are then generated automatically in the time that nodes are initialized.

Finally, the reasoner checks for the validity of generated BN nodes and consistency of domain ontology.

Assigning the conditional probability distribution

As I've mentioned before, this is a field of intensive interest. Here in this paper, the authors use an approach that exploits existing parameter-learning algorithms to learn the CPT incrementally and online from a live feed of network data source.

Some tip here is that the ontology knowledge resource can be utilized to estimate initial probabilities for some concepts. Furthermore, some parent-child relations have probability values based on their nature. These two latter could be modeled / ruled too as an encoded rule in the system.

Application to Telecommunication Network Management Domain

The area of adaptive autonomous networks has been and is a subject of extreme research activity. The most important reasons are:

- ✓ Current explosion in size, complexity and heterogeneity of networks is set to continue in future at an exponential rate;
- ✓ Network operators are struggling in a highly competitive market.

Therefore a conflict is set to prepare between "the ever-increasing demand of running huge, complex heterogeneous networks" and "the ever-decreasing OPEX (operational expenditure) budgets of network operators". A critical obstacle to increased autonomy to overcome is that nowadays network management systems have only partially access to the networks' dataflow for the networks they're managing and unfortunately these models of dataflow are semantically empty.

The ontology here is modeled into OWL and uses OWL-DL reasoner with additional JENA rules. The BN software is Netica API. The algorithm is in Java. Below I've gathered the implementation data in the above-told phases:

Domain ontology

The domain ontology stores the below:

- ✓ current configuration of the device;
- ✓ its relationships with other objects in the network;
- ✓ constraints on its possible configuration;
- ✓ workflows associated with the configuration tasks, i.e. sequence of actions affecting a network element that need to be completed in order to fulfill a given task.

It also models the below:

- ✓ performance and fault metrics associated with the node, i.e. alarm types, performance counters, KPIs;
- ✓ any association between these metrics, i.e. KPI equations, alarm triggers.

A subsection of this ontology –modeled in OWL- is shown in image below. It is focused on SERVICE concept.

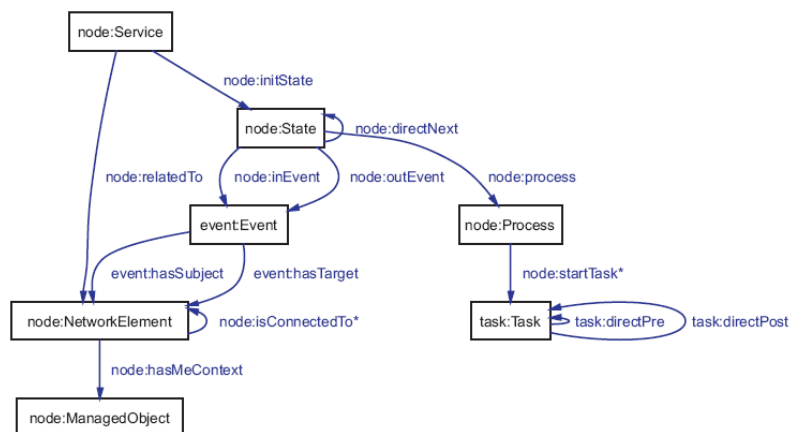


Figure 12: Part of TC Network Management Domain Ontology

Behavior model ontology

The BehaviorModelNode is the root class for any node to be included in the BN and below this node is a hierarchy of more specific nodes / classes for each node type to be included (KPI, Performance Parameter, Service, Event). A part of the combined ontology is shown in below image. Notice that this ontology is based on two node classes: ServiceNode and EventNode. These are subclasses of BMN and describe the properties of service and event concepts. Every service and event of interest should inherit from these two nodes.


```

[
Service-Event_arc_rule:

(?s type Service)      //if there is a service

(?e type Event)        //and an event

(?s ?p ?e)             //that is in relation with the service

→                      //then

(?e hasParent ?s)      //the event is a child of the service

]

```

Notice that if no relation is defined in domain ontology, it's also possible to define rules that explicitly specify arc creation.

This model is going to a consistency check and then recreates a BN using Netica API, as shown in figure 8.

Conclusion and Further Work

To mention the job in this paper again, we saw the automatic building of a BN from an ontology model of a given domain. Future work as the authors claim has the below avenues:

- ✓ Evaluation of the current system:
 - Subjective evaluation: compare how the task is perceived by the ontology or BN builders to estimate about any saving in time and effort of domain experts;
 - Objective evaluation: assess the quality of the generated structure by performing a comparison with other data-learned models on a selected metric.
- ✓ Specification of CPT priors for BN on the basis of ontology models' properties, i.e.:

$$\Pr(\text{service} \mid \text{KPI} \geq \text{threshold}) = 1$$

- ✓ Modification of ontology-built structure on basis of learnt data: existing methodologies involve learning an entirely new structure from data (offline / online) using original structure as a prior in the learning process.

My Own Viewpoint

Up to now, we have seen the approaches of two different recent papers to use from ontologies in BN construction. The importance of the task is well understood and the patterns of the authors are to be respected, though a bit cloudy to see the sun. I'm going to revise the matter of constructing BNs from the window of knowledge engineering as it may help to get the big picture.

As before told, the BN construction matter can go into below phases:

1. Identifying the set of variables and their states the make up the system;

2. Identifying the graphical structure, that is, the qualitative structural assumptions of the system;
3. Identifying the conditional probability tables of each variable, that is, the quantitative effects of variables parent nodes on it.

Along with properly defining the problem domain, an objective of the first two steps is to express the problem in its simplest yet sufficiently complete form. This is done to reduce the number and form of probabilities to be entered into the CPTs which is to be done in the third task, and is often considered the most difficult task.

To select the variables from the domain, it'd be of the great level of importance to do the selecting with reminding to limit the number of variables. This is done to let the model be tractable. For this matter, I introduce here four categories to include the important variables of ours:

1. Query / Objective / Target variables: end users want to know about them
2. Evidence / Observation / Controlling variables: input variables of the system that help to infer the states of querying.
3. Context / Intermediate variables: link the query and evidence variables.
4. Controllable / Intervention variables: could potentially interfere the domain system.

These classes of variables could be modeled in the automated model of ontology, but of course by the help of expert knowledge. This can be done in the phase of generating the BN ontology –the Behavior Model- mentioned in the main paper.

After this, it's time to decide on the states of each variable. A simple guide comes below to help the decision of a variable:

- ✓ The state it is currently in.
- ✓ The state(s) toward which it may move to due to possible managing plans.
- ✓ Any intermediate state.

Mostly only states are included in the knowledge management that are of interest to the user, but in the automation by means of ontologies, it's a bit difficult to do so. It is clear now that any approach, whether manual or automatic, would be a trade-off of simplicity and accuracy.

Another must to make the system tractable is to limit the relationships of the variables. This is done in the phase of graphical model elicitation. It's also important to make the BN understandable for some out-of-domain viewer. Four important relationships between two variables that manage this phase are:

1. Independent
2. Dependent
3. Conditionally Independent
4. Marginally Independent

These relations could be cleared by asking direct questions about any variables. A support tool for this type of elicitation is called MATILDA.

The most important, effective and interesting phase of all is the last one: CPT elicitation. It has been great efforts recently to provide approaches which lead to better estimation of the CPTs for BNs; however it didn't get an optimal way to work out in every domain of interest. Some approaches, due to probabilistic efforts were introduced before. Now I'm going to visit the matter again on the window of knowledge engineering.

In fact, CPT for each node specifies the value of the node given the values of its parents. For each possible instantiation of parent values, there's a probability distribution. This means that the probability estimation is exponential in the number of parents. It's another reason for keeping the structure the simplest whilst properly representing the system. There are three possible elicitation sources:

- ✓ Domain Experts: which may be the only source in some domain; a support tool is called VE which aids by mapping verbal terms into probabilities.
- ✓ Experimental Data: enough data is needed to automate the process of training the CPTs, though there may be problems of noisy data, missing values, bias and not matching values on the model.
- ✓ The Literature: it's usually very unlikely that this source could cover all the needed probability distributions, while it also has the problem of bias.

Another way of CPT estimation could be done by means of HMMs (Hidden Markov Chains). There are some sorts of classes of HMMs which Hierarchical and Layered classes are mostly better for modeling a system under uncertain cause-and-effect circumstances. There are also some known algorithms to be taken in custody for this kind of pure-data-to-information tasks. Some of them, which I've listed below, are based on Estimation Theory and some on Maximum Likelihood:

- ✓ Markov Chain Monte Carlo
- ✓ Forward Backward Algorithm
- ✓ Baum-Welch Algorithm
- ✓ Viterbi Algorithm

Notice that none of these CPT extraction methods which I called here are not standard methods for BNs and the only reason for calling them is unavailability of standard approaches worldwide and also the similarities observed among the two tasks –BN CPT extraction and these classes of methods- which led to an understanding of this usage.

Also the matter of evaluation could be seen from the window of knowledge engineering. Evaluation methods are useful to grade the BN and to identify the errors and also possible environments. Again in this field, we can talk about expert help and the statistical methods which may help automating the evaluating the structure:

- ✓ Domain Expert Evaluation: includes feedbacks from experts who were not involved in the BN creation process. Four types of evaluation can be done in this category:
 - Elicitation review
 - MATILDA
 - Sensitivity analysis
 - Case based evaluation
- ✓ Automated Evaluation: availability of a large amount of data is needed on the domain, better to split the data to 90/10 or 80/20 and use the later section for evaluation. Two types of evaluation can be done in this category:
 - Predictive accuracy
 - Expected value

I believe that here is the time and place for a little critic while I'm running out of data to provide! As I've made an overall study in the field of BN construction, whether manual or automatic, esp. by means of ontologies, all the methods and approaches seem to have some leaks in their body.

First is that they all seem to be a bit young and abstract, though claiming to be implementable. This means that providing a more detailed method will be in interest of some working groups.

The other is that every approach I've met really stands on the shoulders of a group of domain experts to be started or to be ended. This is really not the meaning of automatic or even by-means-of methods which I'm familiar with in concepts of knowledge engineering or software engineering. It seems that everyone is trying to use any help -like ontologies- just to better managing the data from experts in knowledge classification / representation, or better to say: Any method just seems to be automatic!

And at last, every method is only demonstrated on its exact domain of interest or up to the neighbor domains. It means that on any other domains at least a tailor-expert is needed to make the changes on the approach. For example the concepts in the medical domains are mostly causal and in numerous relations to many other concepts, while the models on network error management are in first look the same. But a great difference can be seen in the structure of their ontologies. So the first steps of encapsulating the knowledge into a formal language and identifying the needed variables really differ on these two examples.

As I've contacted to some people in the realm, it seems that everyone is hopeless to grant a full, optimal, over-domain method. So am I!

Acknowledgement

Here, I'd like to express my best regards to the followings who really made me get the big picture by their unsparing help and guidance: Prof. Ann Nicholson, Mr. Boris Danev and Dr. Mohsen Afsharchi.

Appendix I

We have a level of information about ontologies due to the overview in the introduction part. Here we go some deeper:

Why ontologies are needed?

There are some below problems that an ontology as a formal representation could solve:

- ✓ The consequent lack of "a shared understanding" which leads to:
 - Poor communication between people / organizations / agents / ...
 - Difficulties in identifying requirements and defining the specification of system
- ✓ Such an understanding functions as a "unifying framework" for different viewpoints and serve as the basis for:
 - Communication
 - Inter-Operability
 - System Engineering Benefits: reusability, reliability, specification

What exactly is ontology?

Ontology is the shared understanding of some domain of interest, which may be used as a unifying framework to solve the above-told problems. Ontology is a conceptualization which can be implicit or explicit. It usually includes a vocabulary of terms and some specification of their meaning (definition). Four degrees of formality are known to be dealt among ontological models:

1. Highly Informal: like natural languages
2. Semi-Informal
3. Semi-Formal
4. Rigorously Formal: like TOVE

Ontologies to communicate

We consider several aspects of using ontology to facilitate communication:

- ✓ Ontologies provide a normative model of the system and its objectives. This allows semantic transformations between different contexts;
- ✓ Ontologies lead to create a network of relationship and to keep track of what is linked and explore and navigate through this network. This onto-network-of-relationship explicitizes the implicit network by identifying the logical connections between elements across models of system.
- ✓ Ontologies help to have consistency and lack of ambiguity. Where ever we have ambiguity problem, a "meaning mapper" could come in handy.
- ✓ Ontologies help to have integration among different user perspectives, which assists participants in communication and coming to an agreement.

Ontologies and Inter-Operability

Ontology as Inter-Lingua: any information technology environment should use integrated enterprise models which server as a common repository, accessible by multiple tool sets. Below comes the image for an example of translators between languages:

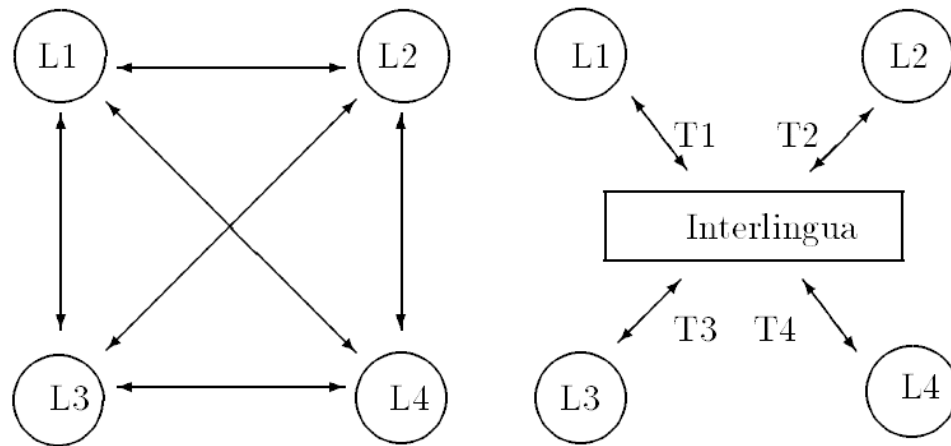


Figure A1: Utilizing an ontology makes the need of $O(n^2)$ translators to decrease to $O(n)$, given n languages

Dimensions of Inter-Operability: Several distinctions can be made in Inter-Operability matters, like the nature of relationships among organizational agents. So we study the below classes of dimensions for Inter-Operability:

- ✓ Internal I-O: all systems here are under direct control of some organizational unit. Some example is shown in the image below:

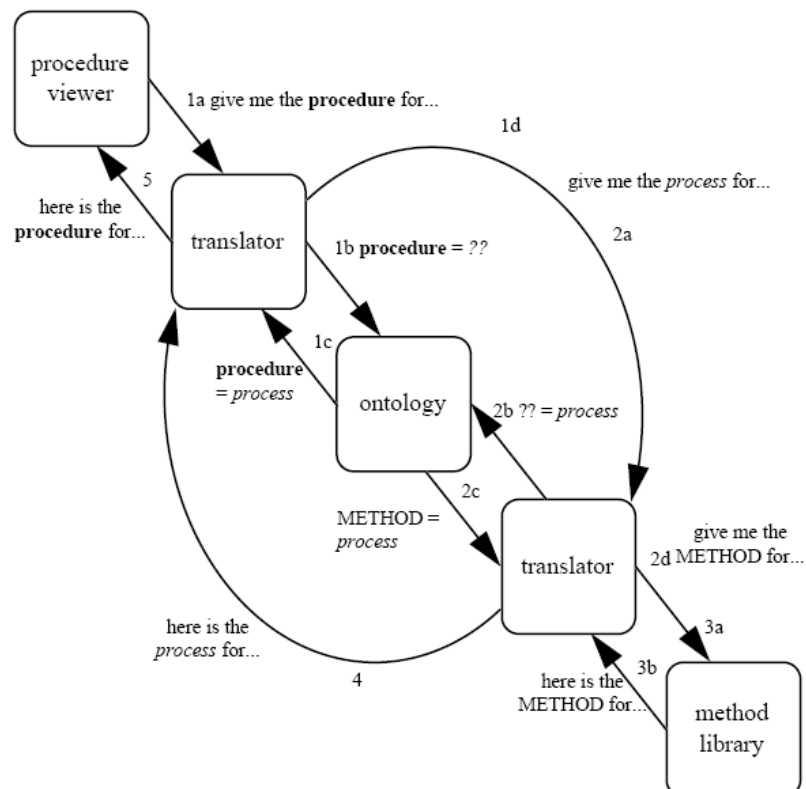


Figure A2: Utilizing an ontology as a control unit between ProcedureViewer and MethodLibrary

- ✓ External I-O: which gives us a unit that wishes to insulate itself from changes imposed on it from outside.
- ✓ Integrated ontologies among domains: different ontologies should be integrated to support some tasks, i.e. the workflow management system ontology is an integration of ontologies for processes, resources, products, services and organization, and workflow tools use this integration as a united ontology.
- ✓ Integrated ontologies among tools: we may also need to integrate different ontologies in the same domain because of legacy system, i.e. the tools which work on process ontologies which lets out the need of a common ontology that all sets of tools can utilize.

Ontologies and System Engineering

Up to now, we mostly discussed on the role of ontologies in the operation of software systems. From now on, we wish to focus on the applications of ontology that support the design and development of software systems.

Specification: The role of ontologies here varies with the degree of formality and automation within the system design methodology:

- ✓ Informal Approaches: ontologies facilitate the process of identifying the requirements of the system and understanding the relationships among the components of the system. This is particularly important for the systems involving distributed teams of designers working in different domain.
- ✓ Formal Approaches: ontologies provide a declarative specification of a software system which allows us to reason about what the system is designed for rather than how it supports this functionality.

Reliability: Here again we deal with the degree of formality:

- ✓ Informal ontologies serve as a basis for manual checking of the design against the specification.
- ✓ Formal ontologies enable the use of (semi)automated consistency checking of the software system with respect to the declarative specification. They are used to explicitizing the different and various assumptions made by different components of system, facilitating their integration.

Reusability: At the time of dealing with new systems -or new domain better to say- we may face some problems with the assumptions which relied to the previous domains. By characterizing the classes of domains and tasks within these domains, ontologies provide a framework for determining which aspects of an ontology are reusable between different domains and tasks. So it may lead to an easy-to-share library of class objects for modeling problems and domains. Ontologies must be customizable both to the class of problems and users and should be extendible. (Partially Shared Views at PIF projects)

A Methodology for building Ontologies

To be honest, there exist no field such as knowledge engineering to be known for ontologies and no standard methodology for ontology construction attempt. Overall we can find some offers as a chance to choose between. Here we come to a conclusion of these above-told offers as a unique approach:

- ✓ Identifying purpose and scope: why it's being built and what its intended uses are? This question hits a reasonably well-defined target to make.
- ✓ Building: hits on three aspects of the purpose and scope, as they come below:
 - Capture: which mean:
 1. Identification of key concepts and relationships in domain;
 2. Production of precise unambiguous text definition for concepts and relations terms to refer to concepts and relation;

It leads to an intermediate representation, which is to include assumptions, justification and definitions.
 - Coding: It makes an explicit representation of the captured conceptualization. It involves the below:
 1. A meta-ontology: basic terms or the underlying ontology
 2. A language to support meta-ontology
 3. Writing the code
 - Integrating existing ontologies: during the capture and code, there's a question of explicitizing all assumptions underlying the ontology, which is a biggest challenge in developing an ontology.
- ✓ Evaluation: is "to make a technical judgment of the ontologies, their associated software environment and documentation with respect to a frame of reference which can be the requirement specifications, competency questions and / or real world" as Gomez and Perez in 1995. It seems that the best approach is to look first at what has been done in the field of KBSs and to adopt it for ontologies.
- ✓ Documentation: a main barrier to effective sharing is lack of documents .So all important assumptions should be documented, both about the main concepts defined in the ontology and the primitives known as the meta-ontology.

References

1. <http://www.wikipedia.com/>(Dynamic Bayesian Networks, Hidden Markov Models)
2. Constructing Bayesian Networks Automatically using Ontologies; Ann Devitt, Boris Danev, Katarina Matusikova; Network Management Research Center, Ericsson, Ireland; Applied Ontology 2006
3. Using Ontologies to Simplify Wireless Network Configuration; David Cleary, Boris Danev, Diarmuid O'Donoghue;
4. Dynamic Bayesian Networks; Kevin P. Murphy; 2002
5. Estimating the Posterior Probability of Bayesian Networks; Gernot D. Kleiter; University of Salzburg
6. Learning Naïve-Bayes Tree for Conditional Probability Estimation; Han Liang, Yuhong Yan; University of Fredericton
7. A Bayesian Method for the induction of Probability Networks from Data; Gregory F. Cooper, Edward Herskovitz; Machine Learning Publishing, Boston; 1992
8. Building Large-Scale Bayesian Networks; Martin Neil, Norman Fenton, Lars Nielson
9. Online Probability Density Estimation of Non-Stationary Random Signal using Dynamic Bayesian Networks; Hyun Cheol Cho, M. Sami Fadali, Kwon Soon Lee; International Journal of Control, Automation and System; 2008
10. Naïve-Bayes Models for Sequential Data; Daniel Lowd, Pedro Domingos; University of Washington
11. Sequential Updating of Conditional Probabilities for Directed Graphical Structures; David J. Spiegelhalter, Steffen L. Lauritzen
12. Ontologies: Principles, Methods and Applications; Mike Uschold, Michael Gruninger; 1996
13. Creating an Ontology for the User Profile; Method and Application; Maria Golemati, Akirivi Katifori, Costas Vassilakis, George Lepouras, Constantin Halatsis