Ontology-driven Automatic Construction of Bayesian Networks for Telecommunication Network Management

Ann Devitt, Boris Danev and Katarina Matusikova
Network Management Research Centre
Ericsson, Ireland.
E-mail: {ann.devitt,boris.danev,katarina.matusikova}@ericsson.com

Abstract. Bayesian Networks are probabilistic structured representations of domains which have been applied to monitoring and manipulating cause and effects for modelled systems as disparate as the weather, disease and mobile telecommunications networks. Although useful, Bayesian Networks are notoriously difficult to build accurately and efficiently which has somewhat limited their application to real world problems. Ontologies are also a structured representation of knowledge, encoding facts and rules about a given domain. This paper outlines an approach to harness the knowledge and inference capabilities inherent in an ontology model to automate the construction of Bayesian Networks to accurately represent a domain of interest. The approach was implemented in the context of an adaptive, self-configuring network management system in the telecommunications domain. In this system, the ontology model has the dual function of knowledge repository and facilitator of automated workflows and the generated BN serves to monitor effects of management activity, forming part of a feedback look for self-configuration decisions and tasks.

1 Introduction

In today's world of digital compression and storage, there is a wealth of data accessible from a vast range of domains and topics and in a huge variety of formats and structures. Over the last number of years, there have been great strides made in how this data can be accessed, indexed and searched. The challenge remains how such data can be exploited as knowledge, facilitating and enhancing new and existing applications. Ontologies have emerged as a means of providing a structured representation of knowledge which can range from generic real world to strictly domain-specific. The purpose of employing an ontological representation is to capture concepts in a given domain in order to provide a shared common understanding of this domain, enabling interoperability and knowledge reuse but also machine-readability and reasoning about information through inferencing. They are deterministic in nature, consisting of concepts and facts about a domain and their relationships to each other. Bayesian Networks have emerged as a means of estimating complex probabilities of states based on graphical models of a domain. They also are a structured representation of knowledge and specify relationships between concepts (or variables) of a domain. These relationships denote the dependencies and independencies that hold between the concepts or variables. They are probabilistic in nature, encoding the probability that variables assume particular values given the values of connected variables in the Bayesian Network structure. These two tools for knowledge representation and manipulation have independently been used to facilitate machine reasoning and decision-making. This paper describes an approach to harness the knowledge representation and inference capabilities of ontologies in order to construct automatically a Bayesian Network which accurately represents a given domain and can then be used to support machine decision-making processes.

The research is being undertaken as part of a program to develop adaptive, self-configuring functionality for devices within the mobile telecommunications network management domain. The research program is addressing the need for efficient management solutions and more automation of mobile networks. Recent advances in wired and wireless networking technology have resulted in an explosion in size, complexity and heterogeneity of such networks. At the same time, network operators are struggling to keep their running costs to a minimum in a highly competitive market. More autonomy of networks, network devices and network management systems presents a means of resolving this conflict between the ever-increasing demands of running large, complex and heterogeneous networks and the ever-decreasing operational expenditure (OPEX) budgets of network operators. The adaptive, self-configuring architecture proposed in [1] exploits emerging technologies such as ontology modelling and bayesian AI to attain the goal of autonomy for network devices. The machine-learning Bayesian Network component of the architecture is designed to provide the *adaptive* functionality, monitoring and learning the effects of configuration actions and closing the feedback loop on management activity. The ontology model component is designed to provide the *self-configuring* functionality, facilitating automation of configuration workflows. In addition to this primary function, the knowledge represented in the ontology model is leveraged in the construction of the Bayesian Network. Both components rely on expert knowledge for their content. Our approach means that this expert knowledge need only be elicited and modelled once as an ontology and can subsequently be manipulated automatically to construct a second model with different properties and functions, the Bayesian Network.

Section 2 gives a brief introduction to Bayesian Networks and outlines current approaches to Bayesian Network construction and how ontologies have been exploited to date for this purpose. Section 3 sets out how in this research the structure and inference capabilities of ontologies have been exploited to automate the construction of a Bayesian Network. Section 4 describes an implementation of this approach in the telecommunications network management domain. Finally, section 5 discusses some conclusions of this research and directions for future work.

2 Background

Korb and Nicholson state that the ultimate goal of Bayesian AI, of which Bayesian Networks are an integral part, is to:

> create a thinking agent which does as well or better than humans in such [reasoning] tasks, which can adapt to stochastic and changing environments, recognize its own limited knowledge and cope sensibly with these varied sources of uncertainty.[2, p.21]

Bayesian Networks provide a means of capturing existing knowledge about a domain, learning the stochastic properties of that domain and thereby adjusting its model of the domain over time. They are currently being exploited in several application areas, notably for estimating effects of different types of behaviour and as support for human or automated decision tasks. Some sample applications include using BNs to reduce power consumption of machines with reference to user behaviour [3] or to diagnose faults in industrial processes [4].

The ultimate goal of an autonomous thinking agent is not yet realised but BNs are currently the state-of-the-art for modelling, monitoring and adapting stochastic processes.

BNs consists of a Directed Acyclic Graph (DAG) structure. The nodes of this graph represent variables from an application domain,[1] for example, performance counters in a telecommunications network or weather indicators in the climate domain. The arcs represent the dependencies that hold between these variables, for example, a drop in parameter X triggers alarm Y or high atmospheric pressure is associated with warm weather. Additionally, there is an associated conditional probability distribution over these variables which encodes the probability that the variables assume their different values given the values of their parent variables in the BN graph structure. For example, the probability of alarm Y being triggered when parameter X is above a given threshold is $p = 1$ or the probability of the weather being good when the atmospheric pressure is high is $p = 0.65$. It should be noted that the arcs of the Bayesian Network do not necessarily denote a causal relationship between two variables but only that the distribution of the child variable values is *dependent* on its parents value. In some instances, this may be a causal relationship but not in all cases. Figure 1 shows a sample Bayesian Network for a set of eight variables from the telecommunications network domain. It consists of a Key Performance Indicator (KPI) for a telecommunications device (*SuccessfulSetupRate*), the performance counters which contribute to that KPI (*NoOfRlAdditionalFailures*, *SetupFailures*, *SetupAttempts*), a service workflow (*ATMConnectionService*) which is triggered by degradation in the KPI levels and two temporal variables, day of the week and peak time. It is a structure such as this which the approach outline in this paper aims to build based on an ontology model of the telecommunications domain.

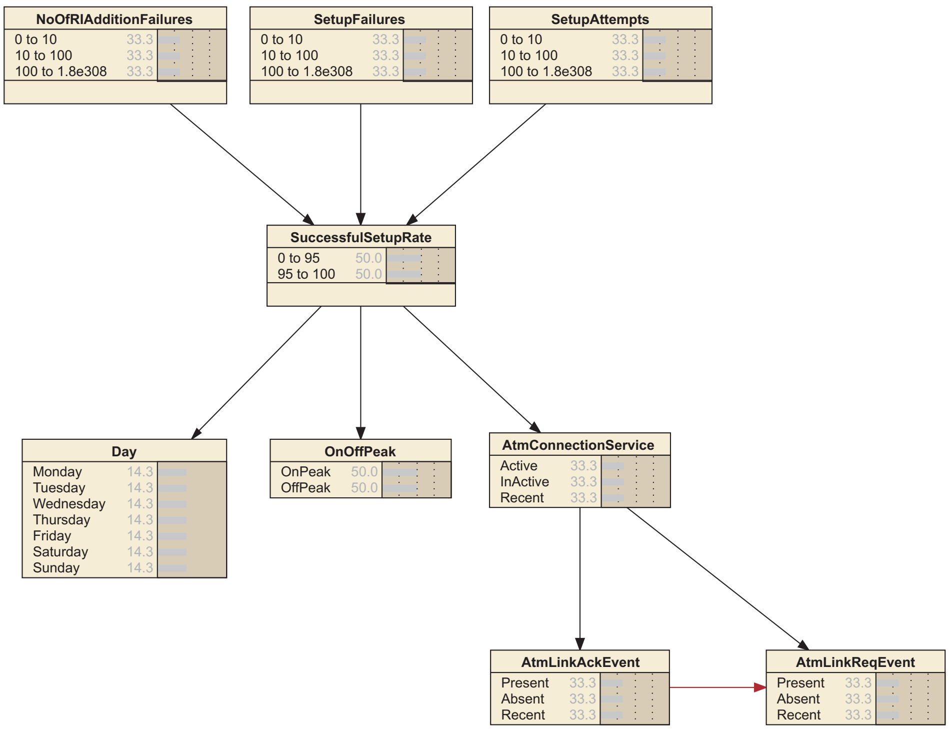

Figure 1: Sample Bayesian Network for Telecommunications Network Management Domain

[1]Terminology: Ontologies represent knowledge in terms of concepts and relations. Ontological concepts, in Bayesian Network terms, are domain variables which can take certain values and have an associated probablility distribution which are represented as nodes in the BN graph. In this paper, we use the terms concept, variable and node interchangeably to denote concepts in the ontology model and variables in the Bayesian Network directed acyclic graph.

The task of building the structure and assigning the probability distributions of a Bayesian Network is complex and knowledge-intensive. It requires the identification of relevant statistical variables in the application domain, the specification of dependency relations between these variables and assignment of initial probability distributions. Both the structure and parameters, or probability distribution, of such a network can be assigned by an expert or learnt off-line from historical data. The parameters may also be learnt on-line incrementally from a live feed of data. Parameter estimation is a mature field of study and several algorithms exist to derive conditional probability tables for a fixed network structure efficiently and accurately from data [5, 6, 7]. In more recent years, structure learning for Bayesian Networks has become a hot topic in the data mining community. Initial BN applications involved defining the network structure and learning the parameters from data. There are a number of methodologies proposed for facilitating building BNs by hand [8, 9] and indeed most BN software tools today include a GUI component for defining BN structures. However, in addition to expert knowledge in the application domain, the human may require some understanding of the statistical principles and in particular the notion of conditional dependence and independence underlying a Bayesian Network in order to correctly specify relations between the variables in the domain. To address this knowledge bottleneck and the inherent difficulties of building BNs by hand, several algorithms have been developed to derive the structure of the network from data, such as the K2 algorithm [10], MDL (Minimum Description Length) [11] and CAMML [12]. While learning causal structure for Bayesian Networks can eradicate some of the bottlenecks which impede the application of BNs to real world problems, the learning algorithms are not without their own drawbacks such as making over-simplifying assumptions about the input data or output structure, inability to deal with missing data, requirement for huge input datasets or intractability for complex multivariate input data. Furthermore, although there may exist a knowledge source representing the domain of interest, it can be a complex task to integrate these sources which in today's data-rich world constitutes an unnecessary waste of resources.

Ontologies provide a potential knowledge source which could be exploited to build the BN structure. Helsper and Gaag [13] outline an approach which uses ontologies to facilitate the building of Bayesian Networks in the medical domain. However, the ontologies are used only as means of representing knowledge to facilitate the manual creation of the BN structure. The ontology constitutes shared and agreed domain knowledge to be used to derive the BN graph structure that is close to the ontological descriptions in the given domain. Due to the complexity of the medical domain and the high impact of misclassifications, the derivation of the graph structure is still done manually by expert analysis. The following sections outline how the BN–from–ontology building process in the telecommunications domain has been automated and enhanced using the inference capabilities offered by the formal ontological representation.

3 Automating Bayesian Network Construction

This section outlines how our approach uses the inheritance structure and inference capabilities of an ontology to build the structure of a Bayesian Network (BN). This addresses two challenges associated with the construction of BNs: the complexity of hand-coding BN structure, requiring both domain and statistical knowledge and the need to integrate existing knowledge sources. As with manual BN construction, the algorithm relies wholly on expert knowledge, rather than evidence derived from data. However, here the work of the expert is simplified. They no longer have to master the statistical principles of BNs, they must only classify their domain knowledge in the familiar ontological world of concepts and relations.

While this is not a trivial task, it is a more straightforward one. Furthermore, where there is an existing ontology knowledge source, little extra input is required to build a BN for this domain. In this implementation, the primary function of the ontology model is workflow automation and the BN construction algorithm is merely a side benefit of having this rich knowledge store.

The complex task of construction a BN for a given domain can be decomposed into four subtasks: 1) identifying the variables of interest; 2) specifying the values these variables can take; 3) defining the relations the hold between the variables and 4) assigning a conditional probability distribution to the variables. The following sections set out how the use of an ontology model facilitates each of these steps.

3.1 Identifying Variables of Interest

The approach described here assumes that an ontology of concepts for the domain of interest has been (as in this project) or can be built. Some of the concepts may be of interest to include in a BN which models causal relations in that domain and some may not. In order to distinguish between these, we have defined a new ontology of BN concepts and link this to the original domain ontology. All concepts of interest for the Bayesian Network then inherit from a node in this BN ontology. The root concept of the BN ontology is the BNnode. In order to create the Bayesian Network, an instance of each leaf class which inherits from the BNnode class is created. The description of the generic BNnode concept, its properties and relations are set out in figure 2. The concept has two types of relations:

- **hasParentNode**: BN nodes have a directed link from themselves to at least one parent node;

- **hasDelayParentNode**: this is a directed time-delay link which can be used to generate a Dynamic Bayesian Network (a BN which includes a temporal dimension) [14].

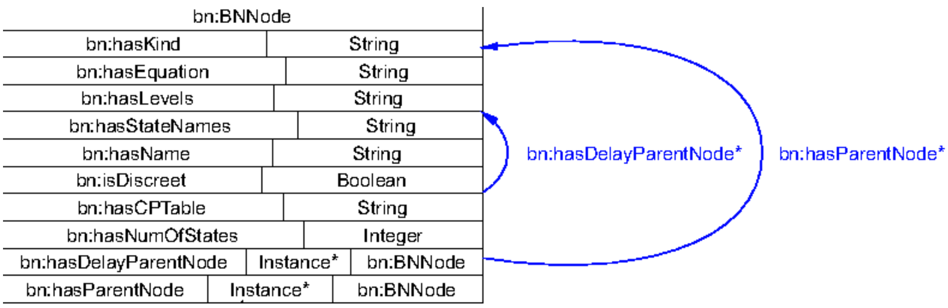

Figure 2: Bayesian Network Node concept

When a BNnode instance is created, these relations define the influential links between this BNnode instance and other BNnode instances (see section 3.3). The BNnode concept properties listed in the figure constitute the set of possible attributes which a BN node can contain. These include name, conditional probability table (CPT), state names (for discreet variables, the list of values which the variable the node represents can take), levels (for continuous variables, the ranges of values which the variable can take). When a BNnode instance of a domain concept is created, the BN attributes are derived from properties of that concept in the domain ontology (see section 3.2). By defining inheritance relations between concepts of interest in the domain ontology and the BNnode concept, it is possible to automate the creation of BN nodes, their attributes and the arcs that connect them, as set out in the sections that follow.

The inheritance relation expresses that a class of the domain model is to be included as a node in the behaviour model. This combined ontology is enriched with facts which describe how a domain can be represented as a Bayesian Network.

The combined domain and BN ontology can be further enriched to constrain BN creation. In addition to the basic BNnode concept, the BN ontology may contain additional BN concepts which are more specific either to the BN application or to characteristics of the domain ontology. Figure 3 illustrates a simple ontology for the telecommunications network management application for which this approach has been implemented. The root concept of the BN ontology remains the BNnode. The domain ontology in this figure consists of the concepts *domain:SubConceptOfNoInterest* and *domain:SubConceptOfInterest* and their parent concept *domain:Concept1*. The *domain:SubConceptOfInterest* concept inherits both from the domain ontology and the BN ontology and only this concept node will be included in an output BN for this domain. In this figure, however, between the root node and a *conceptOfInterest* node, there are two additional, intermediate concepts: *BehaviourModelNode* and *bmConcept1Node*. The BehaviourModelNode concept represents the characteristics of BN nodes required for a particular application, in this case the network management application.[2] This separation between pure BN and BN for an application allows the original generic BNnode ontology to be re-used for other applications which require a BN component by defining a different *ApplicationNode* concept. The *Concept1Node* concept defines characteristics of *Concept1* instances which should be treated in a particular way. The ontology can define a hierarchy of more specific BNnode classes for any domain concepts which should be included in the output BN, if these concepts would benefit from additional processing. This additional level is not a requirement of this approach. However, the structure enables tailored processing of domain ontology concepts in the generation of the Bayesian Network, for example, setting ranges for continuous variables or default probability values.

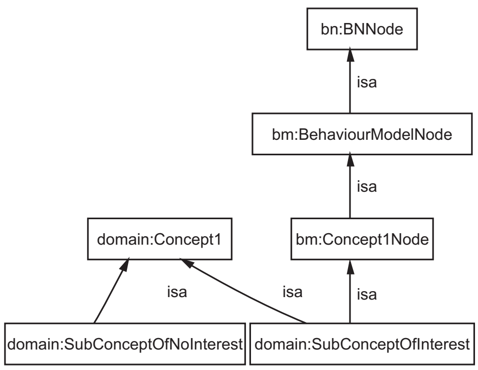

Figure 3: Generic Domain Ontology with BN concepts

3.2 Specifying the Attributes of the Bayesian Network Node

The properties of the BN nodes created at the previous step are derived from the combined BNnode and domain ontology. Properties, such as name, stateNames and kind of bayesian network node, are specified in the BNnode concept and their values are instantiated from the relevant domain concept using the constraints offered by ontology restrictions. The domain concepts specify restrictions on their properties and these are used to populate the BNnode properties of newly created BN nodes. In particular, the *hasValue* restriction specifies the

[2]See section 4 and Baliosian et al [15] for an outline of the application architecture and the BN "Behaviour-Model" component function.

values which a property can assume. For example, the *hasStateNames* BNnode property, which all sub-concepts of BNnode inherit, can be constrained in the sub-concept class to a specific value of the domain ontology concept property using the *hasValue* restriction.

(1)
```
<Restriction>
        <onProperty hasStateNames/>
        <hasValue "Present, Absent, Recent"/>
</Restriction>
```

Other restrictions are used to control correct node notation. For example, the requirement that a node must have exactly one name is expressed by a cardinality restriction:

(2) $hasName\ property = 1$

Figure 4 shows the restrictions on a sample EventNode for the telecommunications ontology, expressing for particular node properties what values this property can take for this class and its subclasses. For example, for all BN event nodes the stateNames property must have values $\{Present, Absent, Recent\}$. The ontology inheritance structure allows some restrictions to be specified at a very generic level (e.g. notational restrictions) and others at a lower level in the ontology (e.g. value specifications), thereby maximising the generalizability of the ontology. For each BNnode instance created, the ontology reasoner compiles all restrictions from the class and all its superclasses and the BNnode properties are generated from these.

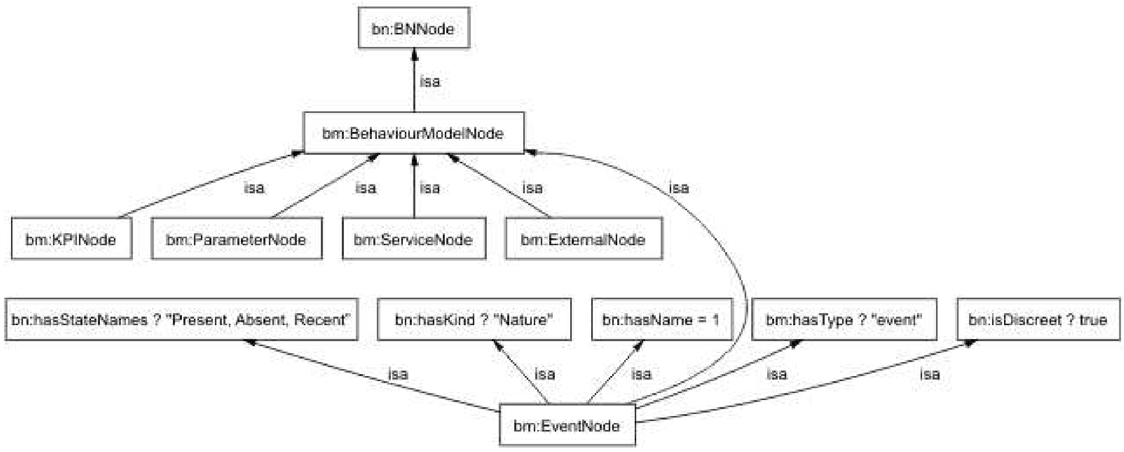

Figure 4: Restrictions on Event Node in telecommunications ontology

3.3 Finding Parent Nodes

In building the Bayesian network from the behaviour model ontology, the ontology reasoner operates over the hierarchy and its restrictions to create Bayesian Network nodes and to populate the properties of those nodes. As noted above, an instance is created for each BehaviourModelNode subclass, in the domain ontology. At the same time, the algorithm also generates a node in the Bayesian Network representation with appropriate property values. To create arcs between these nodes, the algorithm relies on rules. These rules are specific to the application domain and define which ontology properties or relations between concepts correspond to arcs in the Bayesian Network. For example, in the medical domain, a disease may present as one or more symptoms, a single rule can express this causal relation from disease to symptoms for all sub-classes of disease and the symptoms associated with them. The rules are then used to generate arcs in the Bayesian Network from a node representing a disease variable to the node representing its symptoms. Section 4.3 details the rules used to generate BN arcs for the network management application. This rule-based approach over ontology classes provides a means of specifying generic BN relationships which are then generated automatically when the nodes are initialised. Finally, the reasoner is used to check that the generated Bayesian Network is valid by checking all BNnode instances and the domain ontology for consistency.

3.4 Conditional Probability Table (CPT) Estimation

As noted in section 2, estimation of Conditional Probability Tables for Bayesian Networks is and has been a field of intensive research for several decades. The approach set out in this paper does not delve into this area. Indeed the network management application for which this approach was designed exploits existing parameter learning algorithms and the Bayesian Network CPTs are learnt incrementally and on-line from a live feed of network event data. However, the knowledge resource of a domain ontology can be exploited to estimate initial probability distributions for some concepts that lend themselves to this interpretation. Some relations between parent and child nodes can be assigned an initial probability value based on the nature of the concepts involved. For example, a deterministic relation where the value of the parent entails the value of the child variable can be encoded directly in the CPT of the child variable. Like the arc construction method, this can be encoded as a rule in the ontology.

4 Application in Telecommunications Network Management domain

As noted in the introduction, the work described here was conducted as part of large-scale project to develop a self-adapting auto-configuration management functionality for network management systems in the mobile telecommunications domain. The ever-increasing size, complexity and heterogeneity of communications networks today is driving intense research activity in the area of adaptive, autonomous networks and network devices. One crucial obstacle to increased autonomy in today's networks which must be overcome is that network management systems have only partial models of the networks they manage and these models are semantically empty. In deployed commercial systems, the Management Information Base (MIB) records device attributes and the current state of network parameters but it does not explicitly represent constraints that hold for individual managed elements and even less *between* elements of the MIB. Recent approaches propose using ontologies to capture Management Information Base models [16], annotating, constraining and making machine-readable the Operations and Maintenance (O&M) descriptions of the network in order to enable automation of O&M activities, as in [17]. However, a system which allows automation of management decisions and tasks could be prey to instability. A viable autonomous management solution must include a feedback loop to observe and deal with the consequences of its activities. This can be provided using the machine learning capabilities offered by a Bayesian Network to monitor the effects of human, semi-automated or fully automated O&M activities and feed the derived knowledge back into the management system, as described in [15].

The construction algorithm described here was implemented in Java. The Bayesian Network software is the Netica API. The choice of ontology language and support tools for modelling was a matter for much debate. This project builds on previous work which used an ontology defined in F-Logic and the Ontobroker reasoner to validate the results of configuration activities. Having evaluated the merits of F-Logic, WSML and OWL variants (OWL-Lite, OWL and OWL-DL), we decided to employ the W3C standard recommendation OWL. OWL-DL provides the level of expressiveness desired for workflow definitions and validation. Combined with the Jena framework with rule-based inference engine, this OWL flavour allows direct work with subclasses, definition of property ranges and domains and use of restrictions. Furthermore, its guarantee of completeness and decidability is a requirement for any real world application. The debate remains open as to which modelling language will emerge as the true standard for telecommunications and more generally web services applications. In the meantime, this solution provides more than adequate reasoning capabilities to illustrate the benefits of our approach.

4.1 *Domain Ontology*

The domain ontology model is a super-enhanced MIB for a single telecommunications network device. It stores the current configuration of the device, its relationships with other objects in the network and constraints on its possible configuration imposed by the hardware and software deployed on the network element. It also stores the service workflows associated with configuration tasks, i.e., the sequence of actions affecting a network element that need to be completed in order to automatically fulfil a given service request, for example, a request to configure a port. In addition, it models the performance and fault metrics associated with that node (i.e. alarm types, performance counters and KPIs) and any associations between these (e.g. KPI equations, alarm triggers). Figure 5 shows a subsection of this ontology which is focused on the Service concept and other concepts connected with it. The relations between these concepts are expressed by directed links (blue arcs with arrows) representing object properties of concepts. Links lead from property domain to property range concepts. As noted previously, the primary goal of this model is to facilitate automation of management tasks by explicitly representing the events which trigger a service, the sequence of atomic tasks and complex processes that constitute the service and any events issued or expected while the service is active. The secondary function of the model is to generate an accurate BN representation of the domain to be used for monitoring effects of these management services.

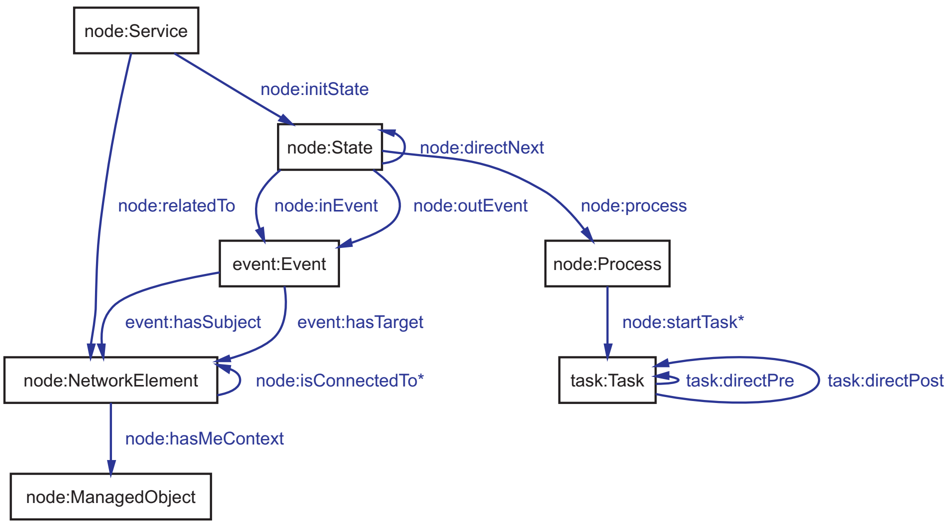

Figure 5: Part of Telecommunications Network Management Domain Ontology

4.2 *Behaviour Model Ontology*

The BehaviourModelNode is the root class for any node to be included in the Bayesian Network. Below this root, there is a hierarchy of more specific node classes for each node type to be included (KPI, Performance Parameter, Service, and Event) to allow custom processing of the various node types. A part of the combined Bayesian Network and Domain ontology for the application domain is shown in figure 6, where *is-a* links represent the inheritance hierarchy. This ontology subsection defines two classes, ServiceNode and EventNode, as subclasses of BehaviourModelNode to describe properties of the service and event domain concepts which are specifically relevant to a Bayesian Network representation. For example, all service nodes in the Bayesian network share the same state names $\{Active, InActive, Recent\}$. This data and other shared property values are recorded using hasValue restrictions on the corresponding ServiceNode properties. Likewise, all event nodes share the same state names $\{Present, Absent, Recent\}$, different from service nodes state names. The EventNode subclass contains this information in the form of hasValue restrictions. Every service and event which is of interest to the Bayesian Network inherits from the ServiceNode subclass and the EventNode subclass respectively.

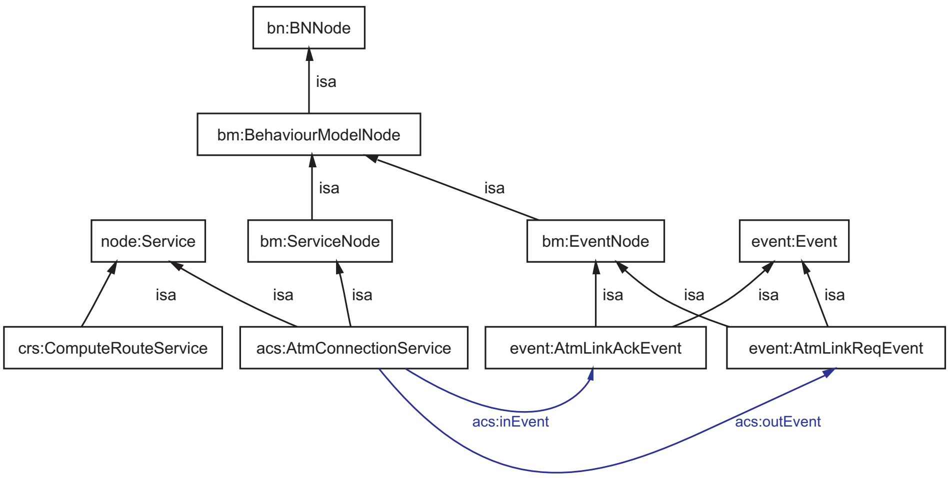

Figure 6: Domain plus Behaviour Model Ontologies

4.3 Rules for BN arc construction

In order to generate arcs automatically from the domain ontology, rules are defined using the Jena rule language. Rules specify how to create arcs from relations between domain concepts. The reasoner infers *hasParentNode* and *hasDelayParentNode* relations from inter-concept relations such as those represented by the blue arcs in figure 6. After rule inference, the *hasParent* relations appear in the behaviour model ontology as shown in figure 7. Example 3 shows a sample rule for generating arcs between the Event and Service concepts.

(3)
```
[Service-Event_arc_rule:
(?s type Service) // if there is a service
(?e type Event)   // and an event
(?s ?p ?e)        // that is related to this service
 ->               // then
(?e hasParent ?s) // the event has the service as a parent
]
```

Likewise, each KPI has its relevant performance parameters defined in the ontology as properties of the KPI concept. A generic rule for all KPIs generates Bayesian Network arcs to each KPI from their associated performance parameters. If there is no relation defined between classes in the domain ontology, it is also possible to define rules that explicitly specify arc creation. This final model is checked for consistency and recreated as a Bayesian Network, such as the BN shown in figure 1, using the Netica API.

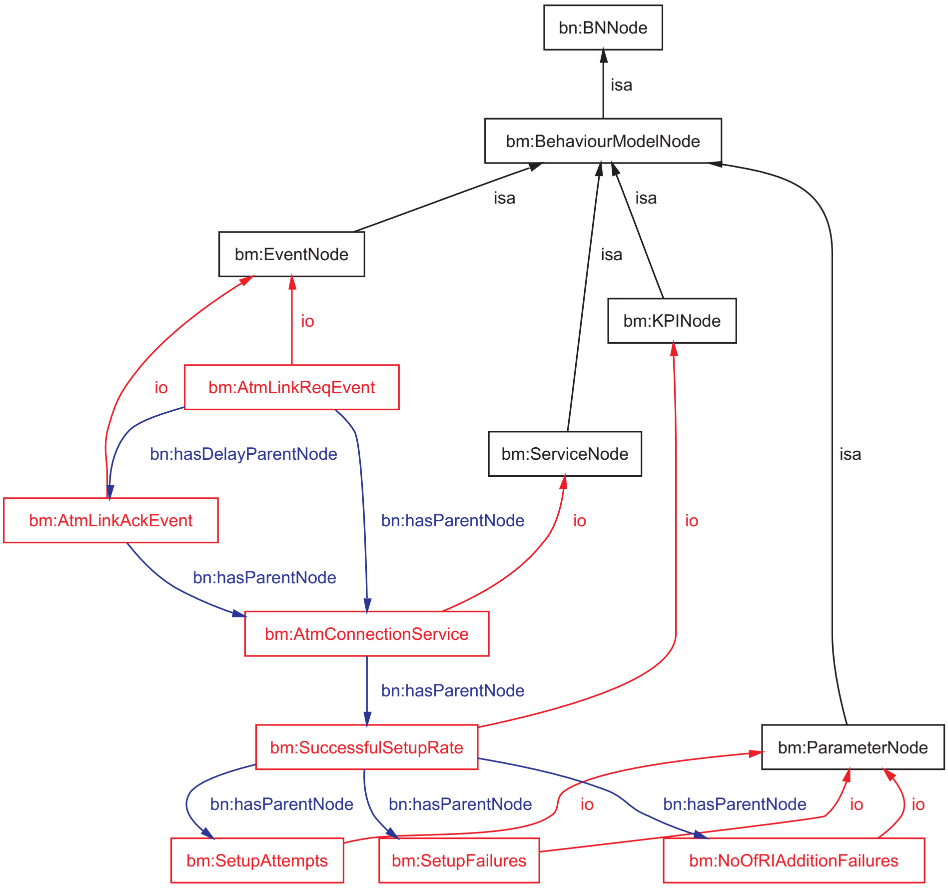

Figure 7: Domain plus Behaviour Model Ontologies after Rule Applications

4.4 Sample Use Case: ATMCrossConnection Service

The use case to evaluate this approach deals with ATM cross-connections. In a 3G WCDMA mobile network, data is transported over ATM connections from the Radio Network Controller (RNC) to its Radio Base Stations (RBSs). In a large network, the data signal may be cross-connected through an intermediate network node, a hub RBS or RXI. The domain ontology for this use case models an ATMCrossConnection service to build and configure a cross-connection on an intermediate network node. The domain ontology models the Managed Object Model of the network device (hub RBS or RXI), relevant Key Performance Indicators (KPIs) and their related performance counters and the ATMCrossConnection service itself. There are two aspects to the definition of this service: the external interface and the internal workflows. The external interface defines the type of events the service will receive and send in each of its states and which constitute the transitional conditions to move from one state to another of the service. The internal workflows define the managed objects, such as ATM ports and channels, to create or modify in the Managed Object Model in order for a node to cross-connect data. This domain ontology consists of 32 concepts and 49 properties.

In this use case, all event and service concepts together with other workflow components (e.g. tasks and processes) and the associated KPIs are "of interest" for constructing the Bayesian Network and therefore inherit from both the domain ontology and the behaviour model ontology. The initial behaviour model ontology contains 9 domain dependent concepts covering these concepts of interest. The output behaviour model ontology after inference and automatic rule based arc creation contains 21 instances of bayesian network node class with

26 hasParent and 12 hasDelayParent relations. This is then mapped to a Bayesian network with 21 nodes, 26 arcs and 12 time-delay links, an extended version of the BN in figure 1.

5 Conclusions and Future Work

This paper has outlined an approach to building a Bayesian Network from an ontology model of a given domain. Bayesian Networks are notoriously difficult to hand-code and structure learning algorithms, while useful, can have significant drawbacks. The use of a domain ontology coupled with the capabilities of an inference engine can automate the BN building task, reducing the knowledge bottleneck of expert knowledge to BN structure, while accurately representing the domain of interest. The approach was implemented in the context of an adaptive, self-configuring network management system in the telecommunications domain. In this system, the ontology model has the dual function of knowledge repository and automation facilitator and the generated BN serves to monitor effects of management activity and forms part of a feedback look for self-configuration decisions and tasks.

This approach opens up several avenues for future work, the first of which is an evaluation of the current system. However, the evaluation of BN structures is a non-trivial task and estimation of the success of this ontology-based approach would require both a subjective and an objective evaluation. The subjective evaluation must compare how the task is perceived by the ontology or BN builders to assess whether there has been any saving in the time and effort of domain experts. The objective evaluation should assess the quality of the generated structure by performing a comparison of the ontology-built structure and other data-learnt models on the basis of a selected metric, such as predictive accuracy for an expert–annotated test data set.

Other technical extensions are also planned. To date, the implemented algorithm does not specify any values for the BN conditional probability tables. In future implementations, we aim to specify CPT priors on the basis of properties of the ontology model. For example, the service workflows which are composed of events imply that the service is active if at least one of its events is present, this could be encoded in the event CPT. Similarly, the triggering of services by KPI violations can be encoded in the service CPT as a deterministic relationship $p_{service} = 1$ when $KPI \geq threshold$. Another more complex direction for future research involves modification of the ontology-built structure by supplementing additional arcs or removing superfluous ones on the basis of learnt data. This is an area ripe for research as existing methodologies entail learning an entirely new structure from data using the original structure as a prior in the learning process. This research direction should also provide interesting insights into the primacy of expert knowledge, in the form of ontologies, over information learnt from data as the degree and kinds of modification required are an indicator of the (in)accuracies of the expert model.

References

[1] Javier Baliosian, Huw Oliver, Ann Devitt, Francoise Sailhan, Epifanio Salamanca, Boris Danev, and Gerard Parr. Self-configuration for radio access networks. In *Proceedings of 7th IEEE Workshop on Policies for Distributed Systems and Networks (Policy 2006)*, London, Canada, June 5-7 2006.

[2] Kevin Korb and Ann E. Nicholson. *Bayesian Artificial Intelligence*. Chapman & Hall/CRC, 2004.

[3] Colin Harris and Vinny Cahill. Power management for stationary machines in a pervasive computing environment. In *Proceedings of the Hawaii International Conference on System Sciences*, 2005.

[4] Gustavo Arroyo-Figueroa and Luis Sucar. A temporal bayesian network for diagnosis and prediction. In *Proceedings of the 15th Annual Conference on Uncertainty in Artificial Intelligence (UAI-99)*, pages 13–20, San Francisco, CA, 1999. Morgan Kaufmann Publishers.

[5] D. J. Spiegelhalter and S. L. Lauritzen. Sequential updating of conditional probabilities on directed graphical structures. *Networks*, 20:579–605, 1990.

[6] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[7] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistics Society*, B 39:1–38, 1977.

[8] K. B. Laskey and S. M. Mahoney. Network engineering for agile belief network models. *IEEE Transactions on Knowledge and Data Engineering*, 12:487–498, 2000.

[9] M. Neil, M. Fenton, and L. Nielsen. Building large-scale bayesian networks. *The Knowledge Engineering Review*, 15:257–284, 2000.

[10] G. Cooper and E. Herskovits. A bayesian method for constructing bayesian belief networks from databases. In *Proceedings of Uncertainty in Artificial Intelligence*, San Fransisco, 1991.

[11] W. Lam and F. Bacchus. Learning bayesian belief networks: An approach based on the mdl principle. *Computational Intelligence*, 10:269–293, 1993.

[12] C. S. Wallace, K. Korb, and H. Dai. Causal discovery via mml. In *Proceedings of the 13th International Conference on Machine Learning*, pages 516–524, San Mateo, CA, 1996. Morgan Kaufman.

[13] E. Helsper and L. C. Van der Gaag. Building bayesian networks through ontologies. In *Proceedings of the 15th European Conference on Artificial Intelligence*, pages 680–684, Amsterdam, the Netherlands, 2002. IOS Press.

[14] Ann E. Nicholson. *Monitoring Discrete Environments using Dynamic Belief Networks*. PhD thesis, Department of Engineering, Oxford, 1992.

[15] Javier Baliosian and Ann Devitt. Forecasting unstable policy enforcement. In *Proceedings of the IEEE Internation Conference on Systems and Networks Communications*, Tahiti, French Polynesia, November 2006. to appear.

[16] Jorge E. López de Vergara, Víctor A. Villagrá, Juan I. Asensio, and Julio Berrocal. Ontologies: Giving semantics to network management models. *IEEE Network, special issue on Network Management*, 17(3), May/June 2003.

[17] David Cleary and Boris Danev. Using ontologies to simplify wireless network configuration. In *Proceedings of the 1st International Workshop Formal Ontologies Meet Industry, FOMI 2005*, 2005.