

# Modeling Bayesian Networks for Autonomous Diagnosis of Web Services

Haiqin Wang, Guijun Wang, Alice Chen, Changzhou Wang  
Casey K Fung, Stephen A Uczekaj, Rodolfo A Santiago

Boeing Phantom Works  
P.O.Box 3707, MC 7L-66  
Seattle, WA 98124

{*haiqin.wang,guijun.wang,alice.chen,changzhou.wang,casey.k.fung,stephen.a.uczekaj,rodolfo.a.santiago*}@boeing.com

## Abstract

We took an innovative approach to service level management for network enterprise systems by using integrated monitoring, diagnostics, and adaptation services in a service-oriented architecture. The autonomous diagnosis for trouble-shooting of web service interruptions is based on Bayesian network models. In this paper, we present our methods for building the diagnostic models. We focus on two types of Bayesian network models of different structure complexity. Our result shows that the two-layer model outperforms the three-layer model in the applied domain. This challenges the common belief that adding unnecessary nodes in a Bayesian network and growing its structural complexity does not deteriorate performance. Hence such practice of building more complex models than necessary should be approached cautiously within the context of the applied domain.

## Introduction

Today as the Internet is used preferably as an information delivery vehicle, more and more companies provide digital documents electronically to their customers through web services. The reliability of web services is critical to the satisfaction of customers, since it directly relates to timely delivery and affects overall system performance. If the web service system is experiencing any problems and running slowly, it is highly desirable to quickly find out the root causes of the problem and take immediate actions to solve it and return the system to normal status.

Traditionally, monitoring, diagnostics, and adaptation services are isolated. Monitoring agents collect data on web service transactions. Monitored events are logged into databases for off-line analysis. If anything critical happens, corresponding alerts are fired instantly to duty managers in charge through pagers or emails. Human interaction is then engaged to solve the problem in a timely fashion.

We take an innovative approach to service level management using integrated monitoring, diagnostics, and adaptation services for networked enterprise systems (Wang *et al.* 2004; 2005b). We implemented on-line analysis of monitoring data for autonomous diagnosis. The diagnostics service and the adaptation service are an integral part of the QoS information management system. The diagnostics service

alerts QoS manager of a critical situation. The QoS manager then automatically makes corresponding adaptation specified by the system policy as a rescue mechanism (Wang *et al.* 2005a). Hence, system performance may downgrade gracefully in emergency. In addition, less human interaction is involved.

On the other hand, as one of the most widely used technologies for diagnostics, Bayesian networks (Pearl 1988) are a compact yet powerful framework for knowledge representation and reasoning. We applied Bayesian networks in autonomous, real-time diagnosis of web services using monitoring data. Two types of models are used to encode the dependence relationships between web service entities. The first type consists exclusively of nodes representing the absolutely necessary components such as web applications, monitoring entities, alert events and such. A node in this kind of model is either a diagnosis target or an observable variable. A diagnosis target is a web service component that can possibly run into problems and trigger alert events. An observable variable can be an indication or test result of the health state of web service, e.g., an alert event. The second type of model has some additional nodes between the diagnosis targets and the observable variables. These intermediate nodes aggregate the health status of possible causes. They link the causes to the alert event nodes. In this paper, we will refer to the first type of model as a two-layer model and the second type of model as a three-layer model, although the actual number of layers in the models may not be exactly two or three.

The experimental results showed that the two models do not have the same reasoning performance in terms of diagnosis. The two-layer model has better performance than its three-layer counterpart. This suggests that making the structure of a Bayesian network more complex may deteriorate its performance than using a simple structure. The additional layer of nodes does not serve as a faithful information channel as assumed. Rather, the evidence message is diluted or weakened somehow when passed through this layer from upper to lower and vice versa.

The reason may be many-fold. One is that the additional layer of nodes encode different independence relationships between the upper layer nodes and the lower layer nodes. And inappropriate independence assumptions result in lower diagnostic accuracy than smaller models that do not include

such assumptions (Fryback 1978). Another reason is the different set of probability parameters used with different model structures. Unless the two sets of probability distributions are equivalent, the reasoning results can be different. This is known as sensitivity of Bayesian networks (Coupé & van der Gaag 2002).

In the following sections, we first describe the web service system and its monitoring and diagnostics architecture. We then spend a large section on our modeling approach to building the diagnostic Bayesian networks for adaptive and reliable web services. We also present our experimental results showing the difference in diagnostics performance of the two kinds of models. In addition, we explain why the two-layer model outperforms the three-layer model. At the end of the paper, we draw a brief conclusion.

## Architecture for Web Service Monitoring and Diagnostics

Web services allow customers around the world to access various e-documents over the Internet anywhere anytime. Hence, web service components are required to be highly reliable. For example, the web applications should run without interruptions; the hosts of the web applications should be robust; the proxy servers and routers should be fairly fast. High reliability of a web service system also requires that the system is constantly aware of its performance status, and is able to diagnose on the fly and adapt whenever necessary.

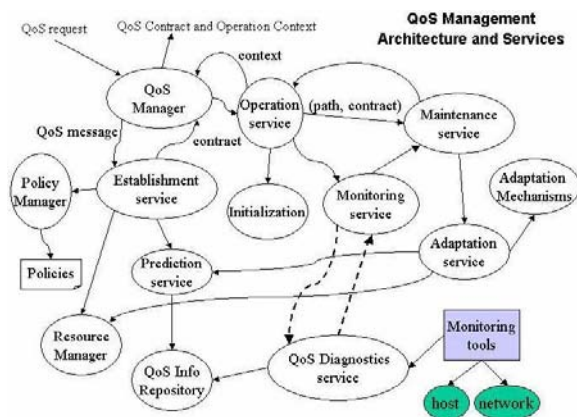


Figure 1: An Integrated QoS Management Architecture for QoS Service Providers.

Traditionally, a monitoring tool is an isolated process and works merely as a data collector. In an integrated QoS management architecture for a web service framework, various component services interact with external services such as real-time host and network condition monitoring. Key component services include *QoS manager*, *establishment service*, *policy manager*, *resource manager*, *prediction service*, *operation service*, *maintenance service*, *monitoring service*, *adaptation service*, and *diagnostics service*. Figure 1 shows the interactions of these services. The interaction between the monitoring service and the QoS diagnostics service follows a registration and notification style, while the interac-

tions among other services in the architecture are based on a request and reply style.

Monitoring, diagnostics and adaptation services are an integral part of end-to-end QoS management. The role of the monitoring service is to sample and aggregate QoS parameter values. It registers condition predicates with the diagnostics service, which returns with notifications when the predicates become true due to changes in system conditions. The diagnostics service is a vital service that uses formal reasoning models like Bayesian networks to aggregate low-level system signals into attributes on system conditions. It takes real-time inputs from monitoring tools, aggregates data on the fly, and stores the data in a repository. It may also evaluate any predicates on the attributes upon value changes and trigger notifications to interested parties such as the monitoring service. When the monitoring service receives the notifications of the conditions of interest, it updates the corresponding data in maintenance service, which in turn activates some adaptation mechanisms, defined in the policy, to take care of the situation through the adaptation service. Figure 2 shows a typical interface between the monitoring and diagnostics services in a web service architecture.

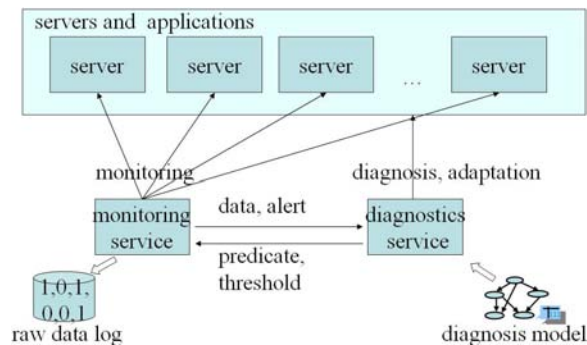


Figure 2: An Architecture of The Monitoring and Diagnosis in Web Service system.

The monitoring service is comprised of many distributed monitoring agents which periodically send simulated web transaction requests to the servers. The transaction delays are compared by the monitoring agents against the pre-configured thresholds. Corresponding alerts are generated if there is any threshold crossed over. These alerts are logged into a relational database for further analysis. There is rich knowledge embedded in the relational database tables that can be extracted for intelligent diagnostics.

## Modeling Diagnostic Bayesian Networks for Web Services

We applied the Bayesian network technology for autonomous, real-time diagnosis for web services based on the monitoring data. The availability of domain experts is often very limited for knowledge elicitation. So our major knowledge resource for building Bayesian networks for diagnostics of web services is the monitored database. Fortunately, in our practice, there are many dependence relationships encoded in the database, which allows for automatic construc-

tion of Bayesian network structures. Probability parameters can be roughly estimated based on domain knowledge and common sense. For instance, it is not hard to estimate how likely the intranet is ok or slow.

Initially all the related entities recorded in the database are added as nodes in the Bayesian network. As a result, we built a three-layer Bayesian network model. However, evaluation of this model showed that, in our application, the three-layer model does not provide a good diagnostic results. So we built a simple two-layer Bayesian network model. We found that the two-layer model outperforms the three-layer model in our evaluation. In this section, we introduce our modeling process, and present evaluation of the two different modeling patterns.

### Three-layer Model and Evaluation

In a Bayesian network, nodes (variables) can be classified into three categories based upon their roles in diagnostics: *target*, *observable*, and *intermediate*.

The first type is *target variables* in diagnostics. These are hypothesis variables that represent uncertain events. They are unobservable but their certainties are of interest to get an estimate. For example, a hypothesis can be the possible failure components in an airplane system (e.g., Line Replacable Units) or diseases in a medical diagnostics system. In diagnostics of a web service, the web service component such as web applications, their hosts, and the Internet or intranet are the target variables. This kind of variable also includes monitoring entities used to constantly monitor web services. Failures (e.g., a slow response) of these components may need immediate attention and corresponding correction.

The second type is *observable variables*. The *observable variables* provide evidence and information that may reveal some clue about the hypothesis events and help to identify the certainties of the hypothesis events. This type of variables includes observable symptoms and achievable tests. In web service diagnostics, event alerts, which describe the status of web service transactions, belong to this category. The alerts are generated by the monitoring entities when monitoring web transactions. They describe in detail what kind of failures happened at what time on which web service component.

The third type is *auxiliary variables*. *Auxiliary variables* help to establish the information channel from causes to effects. They are the intermediate nodes between the hypothesis nodes and the information nodes in a Bayesian network. The auxiliary variables are usually not of interest in reasoning or diagnosis. Except as a mediating function when building a Bayesian network model, the existence of auxiliary variables does not increase the accuracy of reasoning results (Provan 1995; Wang 2005). Therefore, it is not surprising to see some Bayesian network models are bi-partite graphs and only consist of target nodes and observable nodes.

In web service diagnostics, *auxiliary variables* can be component health variables, which describe the aggregated health status of web service components. The component health nodes are intermediate nodes between the service-related components and the alert nodes. They serve as an information transferer. They are designed to have three states,

i.e., good, warning, or critical, to reflect the possible status of web services, i.e., ok, slow, and very slow. The component health nodes usually have multiple parents including web applications, monitoring entities, and intranet.

A full quantification of the conditional probability table (CPT) for a node usually requires many entries. Since a large number of probability values is difficult to obtain, the leaky noisy-max model is applied and greatly reduces the quantification parameters. For instance, a node with three parents requiring 243 probability entries for its CPT now only needs 30 probability entries using the leaky noisy-max model.

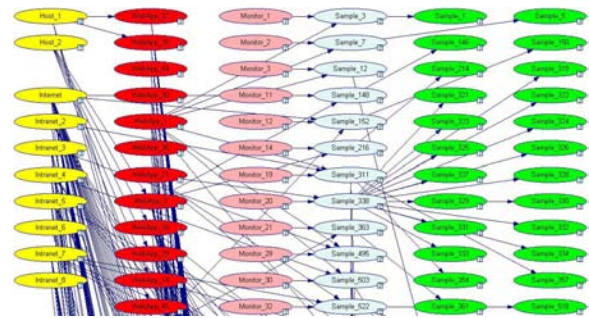


Figure 3: A Fragment of A Three-layer Bayesian Network Model For A Web Service Diagnosis

Figure 3 shows a fragment of a three-layer network model built for our web service diagnostics. In the figure, the green nodes are alert nodes, the light blue nodes are component health nodes, and the rest of the nodes are web service related components.

Usually, use of intermediate nodes helps make the structure of a Bayesian network model easier to manage. Sometimes, it can also help make the elicitation of probabilities less expensive because it may reduce the number of parents of a single node if used properly. The major reason that we have these additional nodes in the network models is that they exist in the monitoring database as a monitoring entities.

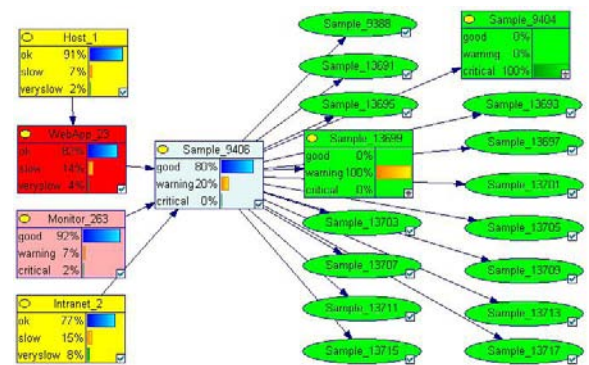


Figure 4: A Diagnosis Example Based on a Fragment of the Three-layer Model

But when we test the three-layer model, the diagnosis results are not very encouraging. See Figure 4 for an example

of the diagnosis results with a small fragment of the three-layer Bayesian network model. Note that the structural layout is rearranged manually for a better visualization of the Bayesian network structure. In this setting, the evidence contains two alert events, i.e., *Sample\_9404* is *critical* and *Sample\_13699* is *warning*. But the posterior probabilities of the corresponding web application, host, monitor and the intranet running ok are still close to or above 80%, which indicate a decent working mode of the whole system.

## Two-layer Model and Evaluation

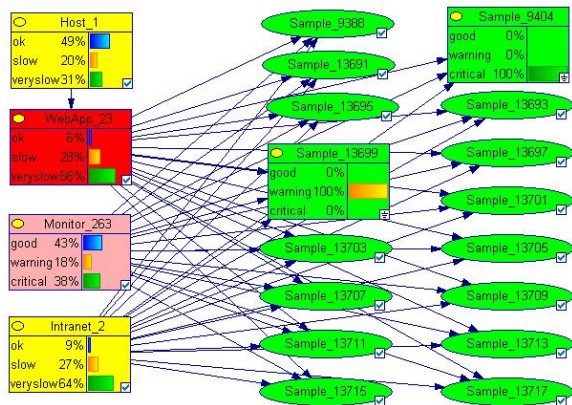


Figure 5: A Diagnosis Example Based on a Fragment of the Two-layer Model

Simply tuning the probability parameters did not work very well to improve the reasoning performance of this network model, so we built a two-layer model. The two-layer model is basically the same as the three-layer model, with the difference that there are no intermediate nodes representing the aggregated component health, i.e., the light blue nodes in Figure 3. Changing the three-layer model to the two-layer model can be done by linking the parents of a component health node directly to its children. Also, the alert event nodes were changed from CPT model to noisy-max model, because their number of parents grows from, typically, 1 (the component health node) in the three-layer network, to at least 3 in the two-layer network.

Figure 5 shows an example of the diagnosis result with a small fragment of the two-layer Bayesian network model. This network fragment is the counterpart of the three-layer network fragment in Figure 4. Again, the structural layout is rearranged manually for a better visualization of the Bayesian network structure. With the same evidence setting, the posterior probabilities of the corresponding web application and the related part of the intranet running ok are very low (6% and 9% respectively). Other related web service components also have a low probability (lower than 50%) of running ok given the evidence. This diagnosis result is much more reasonable and the scenario explanation is much more easily accepted. And it differentiates the possible failures more clearly, with the posterior probabilities of the top two failures being higher than 40% versus the posterior probabilities of other suspects being lower than 10%.

We tested with many scenarios from the monitoring data, and found similar performance difference in the two models. Overall, compared with the three-layer Bayesian network model, the two-layer model is more powerful in diagnosing the probable failures, in differentiating the possible suspects, and in explaining the most likely scenarios. In the diagnosis application for web services, the two-layer Bayesian network is clearly the winner over the three-layer model.

## Discussion

As we see in the previous section, the two-layer model and the three-layer model do not have the same diagnosis results given the same evidence settings. The simpler model can outperform the more complex one in our application for web service diagnosis. This is not consistent with the common belief that a more complex Bayesian network tends to produce better reasoning results than a simpler counterpart. It seems that the additional layer of the nodes do not serve as a faithful information channel as assumed. Rather, the evidence message is diluted or weakened somehow when passed through this layer from upper layer to lower layer and vice versa.

**Independence Relationship** One possible reason is that the additional layer of nodes actually changes the conditional dependence relationships between the upper layer nodes and the lower layer nodes. In the three-layer network, the upper layer nodes and the lower layer nodes are d-separated given the state of the component health nodes that connect them. But in the two-layer network, they are always dependent. The jointree of the three-layer network in Figure 4 has only one 4-node clique, but the jointree of the two-layer network in Figure 5 has almost all the cliques of size 4.

This coincides with Fryback's findings in performance comparisons between complex Bayesian network models and their simple counterparts. With a Bayesian network framework for medical diagnosis, Fryback(Fryback 1978) showed empirically that large models with many inappropriate independence assumptions can have lower diagnostic accuracy than smaller models that do not include such inappropriate independence assumptions. Unfortunately, building complex Bayesian networks usually makes more independence assumptions than building simple networks, and therefore is liable to make more mistakes.

**Probability Parameter** Another possible reason for the performance difference is due to the use of different sets of probability parameters in the two types of models. In the three-layer network, the childless nodes are modeled as CPT nodes and need to have their full CPTs specified. But in the two-layer network, the same variables are modeled as noisy-max nodes, and inherit the probability distributions of the corresponding component health nodes in the three-layer network. Maybe there is a set of CPTs for the childless nodes in the three-layer network that can make the two networks equivalent with regard to their reasoning performance. However, it is not yet discovered in our practice of tuning and validating the networks built. And if the hypothe-

sis holds true, the three-layer network must be very sensitive to its probability parameters. This is not a good property for a Bayesian network which is often desired to have high robustness, reliability, and tolerance of noises.

But having the intermediate nodes in a three-layer network makes the model vulnerable to wrong probability estimates. As the intermediate nodes are inherently implicit and hidden, estimating their probability distributions is more difficult and error-prone, especially when the estimation is based on human being's judgement. Therefore, it is not surprising to see that a three-layer network performs worse than its two-layer network counterpart.

Other research work also investigated the influence of intermediate nodes on the performance of Bayesian networks. Provan(Provan 1995) conducted a series of experiments on some simplified Bayesian networks generated by converting the CPCS network (Parker & Miller 1988) into two-layer models. The original CPCS Bayesian network is built for medical diagnosis in liver and bile disease and consists of 448 nodes and over 900 arcs. In his experiments, Provan used the reduced CPCS networks which consist of, respectively, 42 nodes, 143 nodes and 245 nodes. The comparison criterion is the average of the posterior probabilities of the true positives over the test sets. The results indicate that the intermediate nodes do not make a statistically significant difference in the domain studied.

**Other Related Work** Effort has been made to make the modeling of simple Bayesian networks easier with graphical user interface support. GENIERATE(Kraaijeveld & Druzdzel 2005) is such a tool that allows users to quickly build Bayesian network models under constraint where fault or disease nodes are linked directly to finding or observation nodes without intermediate nodes. Optionally, context nodes, which represent variables of the faults' context properties that may influence the risk of causing the faults, can be added as parents of the fault nodes. Examples of the context variables are the age of a device, the gender of a patient, and the smoking history of a patient. Overall, the network models built using the GENIERATE do not exceed three layers: one layer for fault nodes, one layer for observation nodes, and one layer for context nodes. Note that in our application for web service diagnosis, there are no context nodes. So using GENIERATE to build our models will generate two-layer models, which are preferred by our empirical validation results.

In summary, adding intermediate nodes in the three-layer network model did not encode more domain knowledge in our case. So it is not possible for this complex network model to exceed the simpler model in reasoning power. And the fact that the simpler Bayesian network model outperform its complex counterpart is consistent with a general principle of model selection called Occam's Razor (Russell & Norvig 1995): *The most likely hypothesis is the simplest one that is consistent with all observations.* Or in original Occam's statement: *Entities are not to be multiplied without necessity.* In short, a simple model that is consistent with the domain knowledge is more likely to be correct than a com-

plex one.

## Diagnosis

After Bayesian network models are built as knowledge representations, diagnostic reasoning for web service troubleshooting can be performed by using available inference algorithms provided by many Bayesian network software packages. We developed our diagnostics system based on the SMILE inference engine, which has a jointree (also referred as clustering) algorithm implemented for exact inference in Bayesian networks.

The diagnostics procedure of our web service works as below. First the relevant event data is read from the monitoring database (in offline working mode) or the event data is obtained directly at the time of alert notification (in online working mode). Then this event data is fed into the Bayesian network model and the corresponding evidence is set. To set a piece of evidence is to set the corresponding alert node in the network to its state that represents the observed alert event, i.e., good, warning or critical. After the evidence is set in the Bayesian network model, the jointree inference algorithm is called for exact belief update(Pearl 1988). Finally the updated posterior beliefs are output for the possible fault causes.

As we described in the previous section, not all of the nodes in the Bayesian network model are of interest for diagnosis purposes. Only the web service components, such as certain web applications and their respective hosts, local intranet zones, *et al.*, are considered possible fault causes. These trouble-shooting targets are preset as diagnosis targets in the Bayesian network model for efficient reasoning.

Although in our application, the monitoring service is running constantly without interruption, the monitoring data is collected periodically. Therefore, the diagnostics system only considers the data relevant when its time stamp is within the time window. There may be a delay in data records that can possibly affect the time stamp of the alert event data. However, it is likely that the delay is uniform for all the alert data. Therefore, the delay can essentially be ignored for the fixed time window under the assumption of the uniform distribution.

Figure 6 shows a snippet for a diagnosis result. In the figure, the red box on the top of the figure highlights the event alerts; the green boxes highlight the top suspects of the trouble causes. In this example, some critical failures for accessing some web applications from Internet are detected by the monitoring service, and the diagnostics service found out that the Internet is the most probable cause. Note that in our diagnostics models, the internet node represents any network outside the enterprise security perimeter. It includes the problems of the security perimeter and firewall as well. In the test scenario shown in Figure 6, the corresponding web applications and the host are also likely to be the suspects.

## Conclusion

We took an innovative approach to service level management for distributed enterprise systems by using monitoring,

```

----- Introducing Evidence -----
2004-05-10 03:45:37.0 : Sample 8373 is critical
2004-05-10 03:50:07.0 : Sample 6000 is critical
2004-05-10 03:52:39.0 : Sample 9689 is warning
2004-05-10 03:54:47.0 : Sample 6568 is critical
----- Updating Posterior Beliefs Given the Evidence-----
P(Host_1 is slow) = 0.09      P(Host_1 is veryslow) = 0.90
P(Host_2 is slow) = 0.12      P(Host_2 is veryslow) = 0.20
P(Internet is slow) = 0.02    P(Internet is veryslow) = 0.98
P(Intranet_2 is slow) = 0.10  P(Intranet_2 is veryslow) = 0.05
P(Intranet_3 is slow) = 0.10  P(Intranet_3 is veryslow) = 0.05
P(Intranet_4 is slow) = 0.10  P(Intranet_4 is veryslow) = 0.05
P(Intranet_5 is slow) = 0.36  P(Intranet_5 is veryslow) = 0.18
P(Intranet_6 is slow) = 0.10  P(Intranet_6 is veryslow) = 0.05
P(Intranet_7 is slow) = 0.10  P(Intranet_7 is veryslow) = 0.05
P(Intranet_8 is slow) = 0.10  P(Intranet_8 is veryslow) = 0.05
P(WebApp_32 is slow) = 0.06    P(WebApp_32 is veryslow) = 0.93
P(WebApp_39 is slow) = 0.09    P(WebApp_39 is veryslow) = 0.90
P(WebApp_44 is slow) = 0.08    P(WebApp_44 is veryslow) = 0.02
P(WebApp_30 is slow) = 0.08    P(WebApp_30 is veryslow) = 0.02
P(WebApp_1 is slow) = 0.08     P(WebApp_1 is veryslow) = 0.02
P(WebApp_36 is slow) = 0.08    P(WebApp_36 is veryslow) = 0.02
P(WebApp_21 is slow) = 0.08    P(WebApp_21 is veryslow) = 0.02
P(WebApp_3 is slow) = 0.17     P(WebApp_3 is veryslow) = 0.43
P(WebApp_28 is slow) = 0.15    P(WebApp_28 is veryslow) = 0.21
P(WebApp_29 is slow) = 0.15    P(WebApp_29 is veryslow) = 0.21
P(WebApp_14 is slow) = 0.15    P(WebApp_14 is veryslow) = 0.21
P(WebApp_45 is slow) = 0.15    P(WebApp_45 is veryslow) = 0.21
P(WebApp_5 is slow) = 0.06     P(WebApp_5 is veryslow) = 0.93
P(WebApp_23 is slow) = 0.08    P(WebApp_23 is veryslow) = 0.90
P(WebApp_22 is slow) = 0.08    P(WebApp_22 is veryslow) = 0.90
P(WebApp_43 is slow) = 0.08    P(WebApp_43 is veryslow) = 0.02
P(WebApp_37 is slow) = 0.08    P(WebApp_37 is veryslow) = 0.02

```

Figure 6: A Snippet of a Diagnosis Result

diagnostics, and adaptation services in an integrated architecture. Compared with the traditional approach that uses monitoring as an isolated tool, our approach makes a web service system more reliable, highly adaptive, and quickly responsive.

In this paper, we presented our application of Bayesian network technology as a knowledge representation and reasoning engine in autonomous diagnostics of web services. We used two modeling patterns. One is a two-layer network model. The other is a three-layer network model. Compared with the three-layer Bayesian network model, the two-layer model is more powerful in diagnosing the probable failures, in differentiating the possible suspects, and in explaining the most likely scenarios. In our application in diagnostics of web services, the two-layer Bayesian network is clearly the winner over the three-layer one.

This challenges the common belief that adding unnecessary nodes in a Bayesian network and growing its structure complexity does not deteriorate its performance. A complex network structure makes more independence assumptions that may introduce more errors if the assumptions are not appropriate. Besides, building a network with complex structure is often harder because estimating the probabilities of the intermediate nodes is more difficult and error-prone since these events are implicit and hidden.

In short, the use of additional nodes in a Bayesian network may result in worse performance. Adding the extra nodes is not always valid. Hence such practice of building more complex models than necessary should be approached very cautiously.

## Acknowledgement

We are very appreciative to Michael Kerstetter for his valuable feedback on the draft of this paper. We also thank

Robert Rencher, Robert Ruger, Colleen Jacobson, John Hurley, Grace Bahandari, and Catherine Kitto for their support. All experimental data has been obtained using SMILE, a Bayesian inference engine developed at the Decision Systems Laboratory in University of Pittsburgh. SMILE is available at <http://www.sis.pitt.edu/~genie>.

## References

- Coupé, V. M., and van der Gaag, L. C. 2002. Properties of sensitivity analysis of Bayesian belief networks. *Annals of Mathematics and Artificial Intelligence* 36:323–356.
- Fryback, D. 1978. Bayes’ theorem and conditional nonindependence of data in medical diagnosis. *Computers and Biomedical Research* 11:423–434.
- Kraaijeveld, P., and Druzdzel, M. J. 2005. GeNIeRate: An interactive generator of diagnostic Bayesian network models. In *Working Notes of the 16th International Workshop on Principles of Diagnosis (DX-05)*. To appear.
- Parker, R., and Miller, R. A. 1988. Using causal knowledge to create simulated patient cases: the CPCS project as an extension of internist-1. In PL, M., ed., *Topics in Medical Artificial Intelligence*, Computers and Medicine Series. New York: Springer-Verlag.
- Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufmann Publishers, Inc.
- Provan, G. 1995. Abstraction in belief networks: The role of intermediate states in diagnostic reasoning. In *Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence (UAI-95)*, 464–471. San Francisco, CA: Morgan Kaufmann Publishers.
- Russell, S. J., and Norvig, P. 1995. *Artificial Intelligence: A Modern Approach*. Englewood Cliffs, NJ: Prentice Hall.
- Wang, G.; Chen, A.; Wang, C.; Fung, C.; and Uczekaj, S. 2004. Integrated quality of service (QoS) management in service-oriented enterprise architectures. In *Proceedings of the Eighth IEEE International Enterprise Distributed Object Computing Conference*, 21–32. Monterey, CA: IEEE Press.
- Wang, C.; Wang, G.; Chen, A.; Wang, H.; Pierce, Y.; Fung, C.; and Uczekaj, S. 2005a. A policy-based approach for QoS specification and enforcement in distributed service-oriented architecture. In *Proceedings of the 2005 IEEE International Conference on Services Computing*. Orlando, Florida: IEEE Press.
- Wang, G.; Wang, C.; Chen, A.; Wang, H.; Fung, C.; Uczekaj, S.; Chen, Y.-L.; Guthmiller, W.; and Lee, J. 2005b. Service level management using QoS monitoring, diagnostics, and adaptation for networked enterprise systems. In *Proceeding of the Ninth IEEE International Enterprise Distributed Object Computing Conference*, 239–248. Enschede, The Netherlands: IEEE Press.
- Wang, H. 2005. The more, the merrier? In *Proceedings of the Third Bayesian Modeling Applications Workshop During UAI-2005*.