ISBN 978-83-60810-22-4
ISSN 1896-7094

Rapid Modeling and Analysis with QGeNIe

Marek J. Drużdżel
Decision Systems Laboratory, School of Information Sciences and Intelligent Systems Program,
University of Pittsburgh, Pittsburgh, PA 15260, USA,
Faculty of Computer Science, Białystok Technical University, Wiejska 45A, 15-351 Białystok, Poland
marek@sis.pitt.edu

***Abstract*—QGeNIe is a specialized interface to GeNIe, a decision modeling environment developed by the Decision Systems Laboratory, University of Pittsburgh. QGeNIe allows for rapid construction of graphical models in which all variables are propositional, almost no numerical probabilities are displayed at the user interface, and degrees of truth of propositions are displayed by means of node colors. All numerical parameters, such as prior probability distributions over variables and strengths of influences between variables, are entered by means of graphical sliders. While the underlying computations are all numerical and based on Bayesian updating, QGeNIe makes the impression of a qualitative, "order of magnitude" type system that aids rapid model building and an approximate analysis of systems.**

I. Introduction

One way of supporting business decisions is through creation of business models and subsequent exploration of such models by means of "what if" questions. This approach requires that the models are causal, i.e., when the "what if" question involves any kind of manipulation of the system, the model is able to predict the effects of this manipulation. For example, when ask the question "What if we introduce the third shift on the factory floor?," the model should should be aware that this will change the production capacity, it will increase the costs, but will not impact the customer orders or market price of raw materials or the products.

Ideally, a modeling environment for building such models should possess the following four characteristics: (1) it has to be capable of modeling causal asymmetries among variables, (2) it should allow for relatively fast model building, (3) it should offer an intuitive user interface that facilitates obtaining insight into the problem, and (3) it should be able to work in what is sometimes called "instant gratification" mode, i.e., be computationally efficient and give answers instantly.

Probabilistic graphical models, such as Bayesian networks (BNs) [1], offer two of these. They are acyclic directed graphs and are capable of modeling causal structure of system [2]. This structure, when displayed, is often intuitive and allows for analyzing interactions among variables. While inference in Bayesian networks has been shown to be NP-hard [3], existing commercial and academic systems allow for working with models consisting of hundreds or even thousands of variables.

This paper describes QGeNIe, a modeling environment developed at the Decision Systems Laboratory, University of Pittsburgh. QGeNIe is an attempt to build an ideal system, i.e., one that allows for rapid construction of causal models, that has an intuitive user interface, and that is fast enough to offer an "instant gratification" interface. Technically, QGeNIe is a specialized version of GeNIe, a decision modeling environment developed by the Decision Systems Laboratory, University of Pittsburgh. GeNIe has over 20,000 users worldwide and is slowly becoming a standard tool in research and in teaching such topics as probability and decision theory. It has also been used in several business and industrial applications. All variables in QGeNIe are propositional, almost no numerical probabilities are displayed at the user interface, and posterior probabilities (interpreted as degrees of truth of propositions) are displayed by means of node colors. All numerical parameters, such as prior probability distributions over variables and strengths of influences between variables, are entered by means of graphical sliders. While the underlying computations are all numerical and based on Bayesian updating, QGeNIe makes the impression of a qualitative, "order of magnitude" type system that aids rapid model building and an approximate analysis of systems.

In addition to introducing QGeNIe and describing the modeling techniques that we have made to make facilitate rapid model building, we describe typical environments where it has proven useful: modeling in "soft sciences," brainstorming sessions of the strategic planning type, involving multiple decision makers. Session participants may not be able to specify the exact numerical values of the interactions, but they will typically agree on the structure of the problem and the rough magnitude of the influences between variables. Instant feedback to "what if" questions based on the model allows for quick analysis of the model and reaching consensus and conclusions in group settings.

A few words about the notation. We will use uppercase letters to denote random variables (e.g., X) and lowercase letters to denote their states (e.g., x). Because all variables in this paper will be Boolean, a variable X will take only two states, x and $\overline{x}$. Bold uppercase letters will denote sets of random variables (e.g., $\mathbf{X}$) and bold lowercase letters (e.g., $\mathbf{x}$) will denote value assignments to sets of random variables. We will use $\Pr(X)$ to denote the probability distribution over a variable X.

II. Bayesian Networks

Bayesian networks (BNs) [1] are acyclic directed graphs in which nodes represent random variables and arcs represent direct probabilistic dependencies among them. A Bayesian network encodes the joint probability distribution over a set of variables $\{X_1, \ldots, X_n\}$ and decomposes it into a product of conditional probability distributions over each variable given its parents in the graph. When the parent set is empty, the corresponding factor reduces to prior probability over the variable in question. The joint probability distribution over $\{X_1, \ldots, X_n\}$ can be obtained by taking the product of all of these prior and conditional probability distributions:

$$\Pr(x_1, \ldots, x_n) = \prod_{i=1}^{n} \Pr(x_i|\text{pa}(x_i)) \; . \tag{1}$$

Figure 1 shows a highly simplified example Bayesian network modeling two causes of potential technical problems and a potential project failure faced by an airplane manufacturer. There are five variables in this model: *Increased Load* (L), *Higher Speed Requirements* (S), *Technical Problems* (P), *Project Failure* (F), and *High Fuel Cost* (H). For the sake of simplicity, we assume that each of these variables is binary. For example, P has two outcomes, denoted p and $\overline{p}$, representing “technical problems present” and “technical problems absent,” respectively.

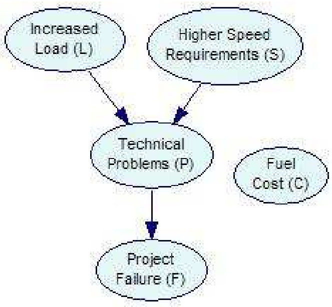


Fig. 1. An example Bayesian network

A direct arc between L and P denotes the fact that whether or not the customer demands a higher carrying load of the airplane will impact the likelihood of technical problems. Similarly, an arc from P to F denotes that technical problems increases the likelihood of project failure.

Lack of a direct arcs is also a way of expressing knowledge, notably assertions of (conditional) independence. For instance, lack of direct arcs $L \rightarrow F$ and $S \rightarrow F$ encodes the knowledge that demands for an increased load and for an increased speed are independent do not impact project failure directly but only indirectly, through technical problems. These causal assertions can be translated into statements of conditional independence: in this case, F is independent of L and S given P. In mathematical notation,

$$\Pr(f|p) = \Pr(f|pl) = \Pr(f|ps) = \Pr(f|ph) = \Pr(f|plsh) \; .$$

Structural independences, i.e., independences that are expressed by the structure of the network, are captured by so called Markov condition, which states that a node (here F) is independent of its non-descendants (here L, S, and H) given its parents (here P).

Similarly, the absence of arc $L \rightarrow S$ means that the demands for an increased load and for an increased speed are independent. The absence of any links between H and the remainder of the variables means that H is independent of the other variables. In fact, H would typically be considered irrelevant to the problem of project failure and we added it to the model only for the sake of presentation.

These independence properties imply that

$$\Pr(l, s, p, h, f) = \Pr(l) \; \Pr(s) \; \Pr(h) \; \Pr(p|l, s) \; \Pr(f|p) \; ,$$

i.e., that the joint probability distribution over the graph nodes can be factored into the product of the conditional probabilities of each node given its parents in the graph. Please note that this expression is just an instance of Equation 1.

The assignment of values to observed variables is usually called *evidence*. The most important type of reasoning in a probabilistic system based on Bayesian networks is known as *belief updating* or *evidence propagation*, which amounts to computing the probability distribution over the variables of interest given the evidence. In the example model of Figure 1, the variable of interest could be F and the focus of computation could be the posterior probability distribution over F given the observed values of L and S, i.e., $\Pr(f|l, s)$.

In addition to computing the effect of observing evidence, when Bayesian networks are causal, they can be used to predict the effect of external manipulation. One approach to manipulation is spelled out by so called *manipulation theorem* [4] that asserts that to compute the effect of manipulating a variable in a model, we need to invalidate its conditional probability distribution and then compute posterior belief over such mutilated network. In the simplest case of perfect manipulation, i.e., one that fixes the value of the variable, this amounts to cutting off all its incoming arcs. Cases where the “arc cutting” fails to model what happens in the system were studied in, for example, [5], [6].

The hardest problem with Bayesian networks, commonly perceived to be the bottleneck in their practical application, is model construction. While building the structure of a Bayesian network is perceived doable, obtaining large quantities of numbers to parameterize them is hard [7]. These, stored as conditional probability tables (CPTs) inside each of the nodes of the graph, grow exponentially in the number of parents of the node. Because nodes with more than 10 parents are not uncommon, this may mean huge quantities of numbers.

One way of handling this problem, directly relevant to the approach taken in this paper is introduction of so called “canonical models” and especially their subset, the Independence of Causal Influences (ICI) models (see [8] for a comprehensive review of the existing ICI models). ICI nodes provide a solution to the problem of quantification of Bayesian networks by assuming that parent variables cause the effect independently of each other. The benefit of this assumption is such that the number of required parameters is linear, rather than exponential, in the number of parent variables. The next section discusses the DeMorgan model, which forms the foundations for the basic building block of QGeNIe.

III. The DeMorgan Gate

Of interest in a system like QGeNIe are such ICI models that are capable of modeling real-world interactions among variables in models. Such interactions may be of a variety of types, although as a minimum one has to be able to model opposing influences, i.e., combinations of influences that increase, and decrease the posterior probability of the child variable. This is important because in practice we always deal with factors that have positive and negative influences. The same variable may have a positive influence on one variable and a negative influence on another. In implementing QGeNIe, we used two interaction models: the CAST gate [9] and the DeMorgan gate [10]. We believe that the CAST gate is semantically less clear than the DeMorgan gate and it causes problems with knowledge elicitation from experts. We have implemented it in QGeNIe for the sake of completeness and compatibility with existing models and software. We will focus in this paper on the DeMorgan gate and refer the interested readers to the paper by Chang [9]. In order to create a CAST gate in QGeNIe, please hold the SHIFT key when clicking on the graph window when creating a node.

The DeMorgan gate is used to model the interaction between a variable and its direct causes (parents of the variable in the directed graph). It allows to specify this interaction in terms of a single number per cause-effect link. There are four fundamental types of cause-effect interactions: *Cause*, *Barrier*, *Requirement*, and *Inhibitor*.

A. Cause

A cause is a parent that has a positive influence on the child. Please note that this influence does not need to be perfect. For example, smoking is generally believed to be a causal factor in lung cancer. Yet, incidence of lung cancer among smokers, while much larger than incidence of lung cancer among non-smokers, is still within a few percent. Hence, the conditional probability of lung cancer given that a person is a smoker is still fairly low.

B. Barrier

A barrier is a parent that decreases the probability of a child. For example, regular exercise decreases the probability of heart disease. While it is a well established factor with a negative influence on heart disease, it is unable by itself to prevent heart disease. One way of looking at a barrier is that it is dual to a cause: Absence of the barrier event is a causal factor for the child. One might go around the very existence of barriers by using negated versions of the variables that represent them. In the example above, one might define a variable "Lack of regular exercise," which would behave as a cause of the variable "Heart disease." This, however, might become cumbersome if "Regular exercise" participated in other interactions in a model. It might happen that it is a parent of both "Heart disease" and "Good physical shape." Because "Regular exercise" decreases the probability of one and increases the probability of the other, barrier, which is a negated cause, is a useful modeling construct.

C. Requirement

A requirement is a parent that is required for the child to be present. There are perfect requirements, such as being a female is a requirement for being pregnant but there are also requirements that are in practice not completely necessary. For example, a sexual intercourse is generally believed to be a requirement for pregnancy, but it is not a strict requirement, as pregnancy may be also caused by artificial insemination.

D. Inhibitor

An inhibitor is a parent that prevents the child from happening. For example, rain may inhibit wild land fire. Like in the other types of interactions, the parent may be imperfect in inhibiting the occurrence of the child. Fire may start even if there is rain. Similarly to the relationship between causes and barriers, inhibitors are dual to requirements: Absence of an inhibitor event is a requirement for the child. One might go around the very existence of inhibitors by using negated versions of the variables that represent them. In the example above, one might define a variable "No rain," which would behave as a requirement for the variable "Wild land fire." This, however, might become cumbersome if "Rain" participated in other interactions in a model. It might happen that it is a parent of both "Wild land fire" and "Good crop." Because "Rain" is an inhibitor for the former and a requirement for the latter, inhibitor, which is a negated requirement, is a useful modeling construct.

In deterministic terms, the four types of causes and an effect interact with their effect through the following logical formula:

$$y = (c_1 \vee c_2 \vee \overline{b_1} \vee \overline{b_2}) \wedge n_1 \wedge n_2 \wedge \overline{i_1} \wedge \overline{i_2} \ ,$$

where:

- cs stand for *Causes*
- bs stand for *Barriers*
- ns stand for necessary conditions (*Requirements*)
- is stand for *Inhibitors*

For the effect, y, to happen, all ns need to be present (one of them absent can bring down the effect) and all is have to be absent (one of them present can bring down the effect). Any c or b can cause y but bs work through their absent state, i.e., their absence can affect y. bs are just negated cs and is are just negated ns.

E. Parameter Elicitation

Of essence to model builders are the questions that are asked of an expert when eliciting the parameter for each link type. These questions have to be clear so as to obtain reliable parameterizations.

It is important to realize that the DeMorgan model is an ICI model. This means in practice that the parents influence the child independently of each other. Influence of each parent can be specified in separation from the influences of the remaining parents. It is assumed that each remaining parent is in its "distinguished state." The distinguished state is the state in which the parent has no effect on the child. Because the effects of the four types of parents are different, their distinguished states are also different.

1) Cause: The distinguished state of a cause is the state in which the cause has no effect on the child. For example, “not being a smoker” has no effect on “lung cancer” in the example introduced above. “Not being a smoker” is the distinguished state in this interaction. So is “having no cancer.”

2) Barrier: The distinguished state of a barrier is also the state in which the cause has no effect on the child. For example, “exercise” may be thought as not influencing the risk of “heart disease.” “Exercise” is the distinguished state in this interaction. So is “no heart disease” in the child node.

3) Requirement: The distinguished state of a requirement is the state that is necessary for the effect to take place at all. For example, “being a female” is a requirement for becoming pregnant and it is the distinguished state in this interaction. “No pregnancy” is the distinguished state of the child node.

4) Inhibitor: The distinguished state of an inhibitor is the state that has no effect on the child, i.e., the inhibiting factor being absent. For example, “rain” is an inhibitor of wild land fire. Its distinguished state is “rain,” in which case the fire may not happen.

Now, for each type of interaction, the parameter associated with the causal link from a parent p_i corresponds to the probability of the effect e happening if all parents are in their distinguished states (i.e., not acting upon e and p_i is not in its distinguished state. There is also an additional parameter, called “leak” which expresses the probability of the effect given that all parents are in their distinguished states.

F. Example

Consider the following network based on the DEMORGAN gate with one cause (*Fire Spreads Quickly*), one barrier (*Fire Escapes Are Accessible*), one requirement (*People Are Still In the Building*), and one inhibitor (*Fire Is Quickly Controlled*):

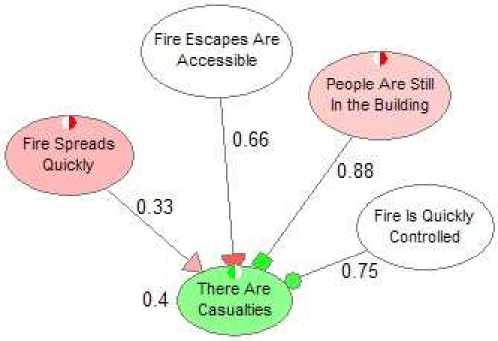

Fire Escapes Are Accessible
People Are Still In the Building
Fire Spreads Quickly
0.66
0.88
0.33
Fire Is Quickly Controlled
0.75
0.4
There Are Casualties

Fig. 2. An example of DEMORGAN gate

1) The “leak” parameter: In order to elicit the leak parameter (0.4 in Figure 2), the knowledge engineer has to ask the following question: *What is the probability of casualties if the fire does not spread quickly, fire escapes are not accessible, people are still in the building, and fire is not quickly controlled? Please note that this probability can be due to other, unmodeled causes.*

2) Cause: In order to elicit the strength of the influence of the node *Fire Spreads Quickly* on *There Are Casualties* (0.33 in Figure 2), the knowledge engineer has to ask the following question: *What is the probability of casualties if the fire spreads quickly, fire escapes are not accessible, people are still in the building, fire is not quickly controlled, and no other unmodeled causal factors are present?*

3) Barrier: In order to elicit the strength of the influence of the node *Fire Escapes Are Accessible* on *There Are Casualties* (0.66 in Figure 2), the knowledge engineer has to ask the following question: *What is the probability of casualties if the fire does not spread quickly, fire escapes are not accessible, people are still in the building, fire is not quickly controlled, and no other unmodeled causal factors are present?*

4) Requirement: In order to elicit the strength of the influence of the node *People Are Still In the Building* on *There Are Casualties* (0.88 in Figure 2), the knowledge engineer has to ask the following question: *What is the probability of casualties if the fire does not spread quickly, fire escapes are not accessible, there are no people in the building, fire is not quickly controlled, and no other unmodeled causal factors are present?*

5) Inhibitor: In order to elicit the strength of the influence of the node *Fire Is Quickly Controlled* on *There Are Casualties* (0.75 in Figure 2), the knowledge engineer has to ask the following question: *What is the probability of casualties if the fire does not spread quickly, fire escapes are accessible, there are people in the building, fire is quickly controlled, and no other unmodeled causal factors are present?*

Each of the above questions can be adjusted to the needs of particular context, i.e., things can be rephrased or omitted if they do not make sense.

IV. MODEL BUILDING WITH QGENIE

Model building with QGENIE consists of building the structure and subsequently estimating the model parameters.

All QGENIE variables are binary and should refer to some proposition. The truth of this proposition will be the subject of inference in QGENIE. It is a good heuristic to think of these propositions as desirable and undesirable. When working with the model, their posterior probability will be displayed by a color. Typically, one assigns green color to desirable propositions and red color to undesirable ones. Red has been found to draw user’s attention and we advise that it be used to undesirable propositions. QGENIE allows its user to define a node coloring scheme. When defining this scheme, it is a good idea to follow the meaning of the majority of nodes. Single nodes can be designated as having the reverse meaning and the coloring scheme will be reversed for them.

It is best to assign variable names that are meaningful and self-explanatory. Please note that QGENIE does not put any limitations on the length of the names. The program also allows for placing groups of nodes into submodels. We advise to make a generous use of submodels in case your models become larger than, say 50 variables. A large number of variables may not fit on the screen and may make the interaction with the model cumbersome.

Interactions among variables are defined by means of influence arcs. These are drawn by clicking the mouse on the parent node and dragging and dropping it onto the child node. Figure 3 show a screen shot of this process. Please note that QGENIE prompts the user to assign the arc one of the four types of arcs defined by the DEMORGAN gate. Clicking on one of the four types defines the arc as being of that type.

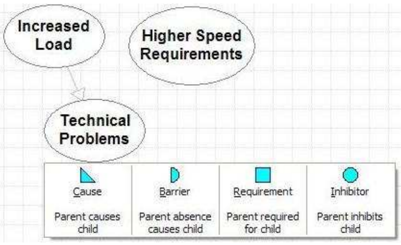


Fig. 3. Drawing an arc in a QGENIE model

Model parameters in QGENIE are defined by double-clicking on a node or an influence arc. This pops up a slider that can be used to enter the numerical parameter, as defined in Section III. Figure 4 shows a screen shot of this process. The numerical value in the slider can be chosen by clicking a corresponding section of the slider or by moving the slider by means of arrow keys. Slider invoked by double-clicking a node corresponds to the "leak value" (or the prior probability of the proposition in case the node has no parents) while a slider associated with an arc corresponds to the influence of the parent on the child through that arc. The meaning of each of the parameters was explained in Section III.

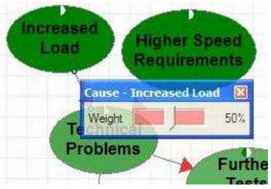


Fig. 4. Entering a numerical parameter value in QGENIE by means of a slider

V. STATIC MODELS

The most basic type of models created in QGENIE are static models. Static models can be viewed as qualitative abstractions of Bayesian networks. Figure 5 shows a simple static network modeling issues related to a project under consideration by an airplane manufacturer. Colors of nodes express the posterior probabilities of the propositions that they represent. "What if" questions are asked by double-clicking a node or an arc and modifying the value of the parameter. After a short delay (introduced on purpose), QGENIE calculates the new posterior probabilities and the colors that they correspond to.

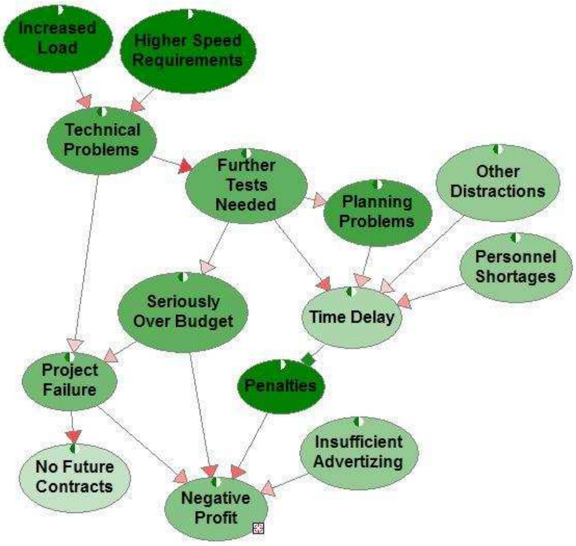
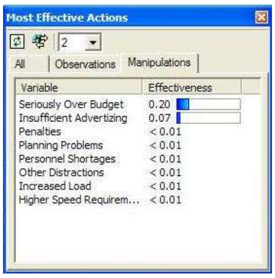


Fig. 5. An example static model

VI. COMPUTATIONAL SUPPORT FORT MOST EFFECTIVE ACTIONS

It is of interest to a user which propositions can be best manipulated in order to affect a specified node most. To this effect, QGENIE allows for specifying a *Focus node*, denoted by a small target icon on the graph. There is a special dialog window (Figure 6) in which QGENIE lists all possible manipulations (every node can be designated to be observable or manipulable) along with the rough value of this manipulation in terms of the magnitude of the expected change in the focus variable. This value is essentially the cross-entropy between the focus variable and the node in question.



Variable	Effectiveness
Seriously Over Budget	0.20
Insufficient Advertizing	0.07
Penalties	< 0.01
Planning Problems	< 0.01
Personnel Shortages	< 0.01
Other Distractions	< 0.01
Increased Load	< 0.01
Higher Speed Requirem...	< 0.01

Fig. 6. Value of manipulating various pressure points

VII. TEMPORAL MODELS

QGENIE extends static models into temporal domains. A temporal model allows for tracing the development of a system over time. Figure 7 shows a simple model for a manufacturer, who is brainstorming whether much attempt should be made to increase the quality of a newly launched product. The model looks 10 time steps ahead. These can be viewed as months or years. An additional window allows for viewing a plot of the numerical posterior probabilities. Figures 8 and 9 show this plot for high quality of product and low quality

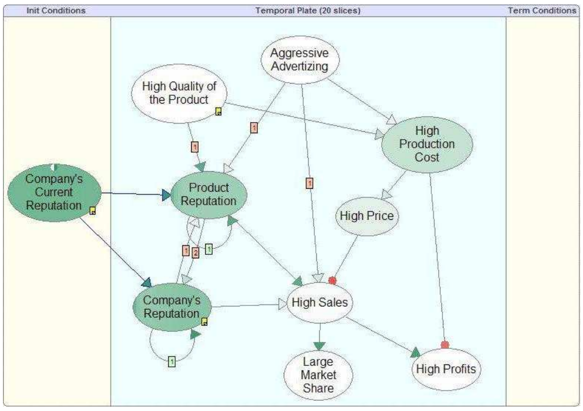


Fig. 7. A dynamic model analyzing the influence of the quality of new product on business

of product respectively. Please note the development of the posterior probabilities in each case.

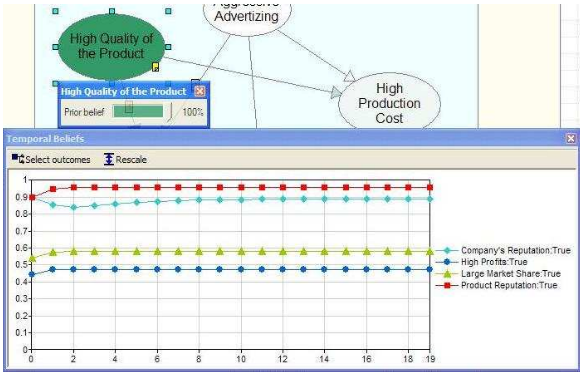


Fig. 8. A dynamic model analyzing the influence of the quality of new product on business: Development of key economic indicators when the quality of the new product is high

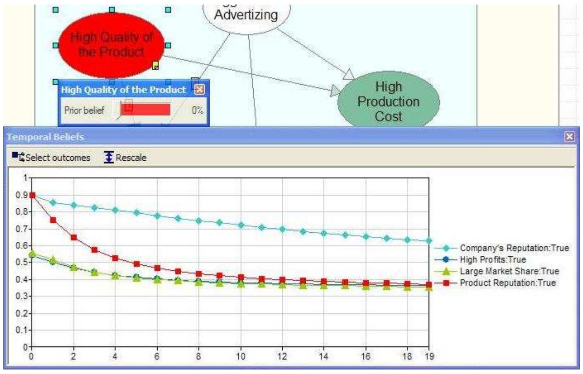


Fig. 9. A dynamic model analyzing the influence of the quality of new product on business: Development of key economic indicators when the quality of the new product is low

VIII. Applications of QGeNIe

A. *Rapid Prototyping*

Even if the goal is to build a fully quantified Bayesian network, QGeNIe is extremely valuable as an environment for rapid prototyping. Models developed withinQGeNIe can be exported to GeNIe for further elaboration and parameter refinement.

B. *Modeling in "Soft Sciences"*

As late Herb Simon convincingly argued [11], "soft sciences" are really hard sciences. Models of social sciences systems are really hard to build, because very often little is known about them. Still, there is ample evidence from behavioral decision theory (e.g., [12]) that even simplest mathematical models typically perform better than unaided human intuition.

QGeNIe has been applied to projects that are truly hard to tackle with formal methods. One example that we are aware of is an application in modeling security in Costa Rican cities [18]. Figure 10 shows a simple QGeNIe model used in the study.

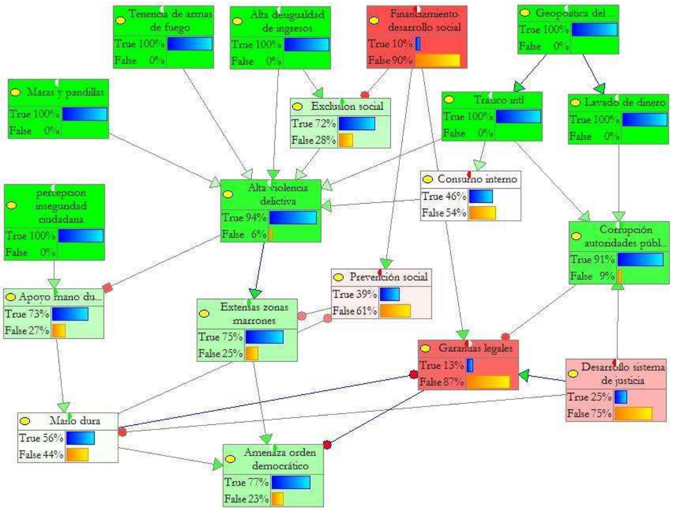


Fig. 10. A QGeNIe model for city security in Costa Rica

C. *Group Decision Making*

The original inspiration for QGeNIe was provided by our collaboration with policy analysts at the United States Naval War College, Bradd Hayes and Theo Gemelas. A model developed there (see Figure 11, the foundations for this model were developed in [13], Figure 5-1, page 101) consisted of 99 variables organized into 12 submodels. The goal of this model was to bring together experts from a variety of areas relevant to stability of a region (in this case, the Black Sea region). The experts know some aspect of the problem (e.g., economy, culture, or energy) but not everything. They may help with building various parts of the model. Asking "what if" questions of the complete model allows each of the individual experts to verify their intuitions but also see how manipulations propagate through those submodels that they did not know much about. A model of this complexity cannot typically be understood completely by a single expert. Presence of multiple experts in a room, each of whom is familiar with a parts of the model, usually helps with verifying model assumptions and obtaining insight into the problem and consequences of the resulting decisions.

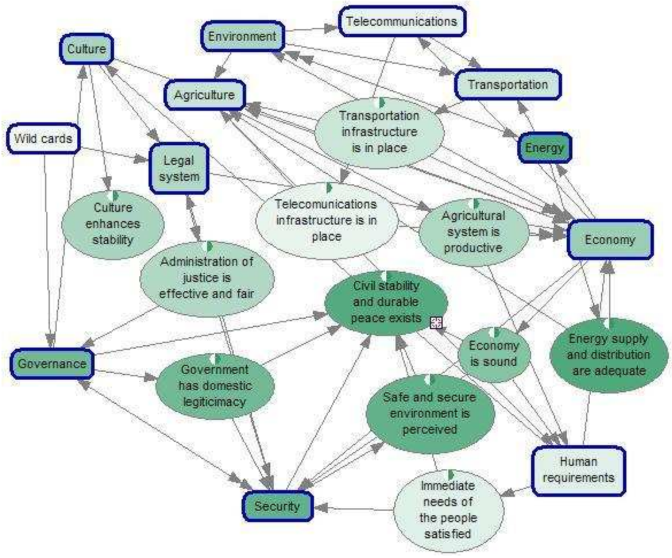

Culture
Environment
Telecommunications
Agriculture
Transportation
Wild cards
Transportation infrastructure is in place
Energy
Legal system
Culture enhances stability
Telecomunications infrastructure is in place
Agricultural system is productive
Economy
Administration of justice is effective and fair
Civil stability and durable peace exists
Economy is sound
Energy supply and distribution are adequate
Governance
Government has domestic legiticimacy
Safe and secure environment is perceived
Human requirements
Security
Immediate needs of the people satisfied

Fig. 11. A model for stability of Black Sea region

IX. Related Work

A modeling environment similar to that of QGeNIe was developed by SAIC. The program, available at SAIC's web site (`http://www.inet.saic.com/`), is based on the CAusal STrength (CAST) model [9], which is an extension of BNs that is able to model simultaneous opposing influences. Although very popular, particularly in government and military applications, a major weakness of the CAST model is its unclear parametrization. Parents can influence a child variable in both of their states and do not have a distinguished state, hence, are not amechanistic. This, we believe is a major weakness of the CAST model. We have encountered puzzled looks on the part of our experts when we tried to elicit CAST parameters, especially the "baseline" probability, which may be viewed as "distinguished belief." A detailed analysis of the CAST parameterization has shown that it can be neither interpreted as the prior probability, nor it is the leak probability.

X. Conclusions

A question that one might ask is whether models created by means of QGeNIe are expressive enough to model real systems. We believe that as long as building a model on propositional logic is sufficient, the answer should be affirmative. The DeMorgan model is able to model any logical interaction between inputs, when their influences on the output are independent, i.e., when they are ICI. In particular, DeMorgan gate can handle a combination of positive and negative influences, while preserving both probabilistic soundness and the amechanistic property, critical in probability elicitation. Each parameter for the DeMorgan model can be elicited by asking a simple question in the form $\Pr(Y|\mathbf{X})$. Probabilistic soundness ensures that it is mathematically correct, and propositional logic, that lies at its foundations, ensures that our model is meaningful and intuitive for humans.

An important question that arises from this work is whether an "order of magnitude" system will be reliable enough in terms of its precision. While this is a good question that will have to be probed in real applications, there are good reasons for believing that precision of numerical parameters in a Bayesian networks is not critical. Two series of studies, introducing random noise with and without bias, and progressive rounding of parameters led to minimal degradation in diagnostic quality of real diagnostic models [14], [15], [16], [17].

We have embedded the DeMorgan model in SMILE and QGeNIe, a qualitative interface to SMILE, our probabilistic reasoning engine, and made it available to the community (`http://genie.sis.pitt.edu/`). QGeNIe is useful in rapid modeling of problems involving propositional variables. We are currently working on extending the DeMorgan model to multi-valued variables along the lines of the Noisy-MAX and Noisy-MIN gates.

XI. Conclusion

QGeNIe is a specialized interface to GeNIe, a decision modeling environment developed by the Decision Systems Laboratory, University of Pittsburgh. QGeNIe allows for rapid construction of graphical models in which all variables are propositional, almost no numerical probabilities are displayed at the user interface, and degrees of truth of propositions are displayed by means of node colors. Its underlying DeMorgan gate formalism allows for quick quantification of the models. All numerical parameters, such as prior probability distributions over variables and strengths of influences between variables, are entered by means of graphical sliders. While the underlying computations are all numerical and based on Bayesian updating, QGeNIe makes the impression of a qualitative, "order of magnitude" type system that aids rapid model building and an approximate analysis of systems. There is ample anecdotal and empirical evidence that precision of numbers is typically not that important.

There are several applications of QGeNIe: rapid prototyping and later elaboration on the model, modeling of vague and difficult to quantify domains, and group decision making sessions. Interested readers can download and explore QGeNIe at `http://genie.sis.pitt.edu/`.

Acknowledgments

This work has been supported by the Air Force Office of Scientific Research grant FA9550-06-1-0243 and by Intel Labs. QGeNIe is based on SMILE, a Bayesian inference engine developed at the Decision Systems Laboratory and available at `http://genie.sis.pitt.edu/`. First steps in developing QGeNIe were made during my interactions with Brad Hayes and Theo Gemelas at US Naval War College. I am indebted to both Bradd and Theo for introducing me to qualitative modeling and for offering many suggestions improved the quality of QGeNIe. Almost all of the implementation work was done by Tomek Sowinski.

References

[1] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufmann Publishers, Inc., 1988.

[2] M. J. Druzdzel and H. A. Simon, "Causality in Bayesian belief networks," in *Proceedings of the Ninth Annual Conference on Uncertainty in Artificial Intelligence (UAI–93)*. San Francisco, CA: Morgan Kaufmann Publishers, Inc., 1993, pp. 3–11.

[3] G. F. Cooper and E. Herskovits, "A Bayesian method for the induction of probabilistic networks from data," *Machine Learning*, vol. 9, no. 4, pp. 309–347, 1992.

[4] P. Spirtes, C. Glymour, and R. Scheines, *Causation, Prediction, and Search*. New York: Springer Verlag, 1993.

[5] M. J. Druzdzel, "Probabilistic reasoning in decision support systems: From computation to common sense," Ph.D. dissertation, Department of Engineering and Public Policy, Carnegie Mellon University, Pittsburgh, PA, Dec. 1992.

[6] D. H. Dash and M. J. Druzdzel, "A fundamental inconsistency between equilibrium causal discovery and causal reasoning formalisms," in *Working Notes of the Workshop on Conditional Independence Structures and Graphical Models*. Toronto, Canada: Fields Institute, 27 September – 1 October 1999, pp. 17–18.

[7] M. J. Druzdzel and L. C. van der Gaag, "Building probabilistic networks: "Where do the numbers come from?" guest editors' introduction," *IEEE Transactions on Knowledge and Data Engineering*, vol. 12, no. 4, pp. 481–486, July–August 2000.

[8] F. J. Díez and M. J. Druzdzel, "Canonical probabilistic models for knowledge engineering," Unpublished manuscript, available at http://www.ia.uned.es/∼fjdiez/papers/canonical.html, 2008.

[9] K. Chang, P. Lehner, A. Levis, A. K. Zaidi, and X. Zhao, "On causal influence logic," Technical Report for Subcontract no. 26-940079-80, George Mason University, Center of Excellence for C3I, Tech. Rep., 1994.

[10] P. P. Maaskant and M. J. Druzdzel, "An ICI model for opposing influences," in *Proceedings of the Fourth European Workshop on Probabilistic Graphical Models (PGM–08)*, M. Jaeger and T. D. Nielsen, Eds., Hirtshals, Denmark, September 17-19 2008, pp. 185–192.

[11] H. A. Simon, *The Sciences of the Artificial*. Cambridge, MA: The MIT Press, 1969.

[12] R. M. Dawes, *Rational Choice in an Uncertain World*. Hartcourt Brace Jovanovich, Publishers, 1988.

[13] B. C. Hayes and J. I. Sands, "Doing windows: Non traditional military responses to complex emergencies," Decision Support Department, Center for Naval Warfare Studies, U.S. Naval War College, Newport, RI, USA, DSD Research Report 97–1, 1997.

[14] M. Pradhan, M. Henrion, G. Provan, B. del Favero, and K. Huang, "The sensitivity of belief networks to imprecise probabilities: An experimental investigation," *Artificial Intelligence*, vol. 85, no. 1–2, pp. 363–397, Aug. 1996.

[15] A. Onisko and M. J. Druzdzel, "Effect of imprecision in probabilities on the quality of results in Bayesian networks: An empirical study," in *Working Notes of the European Conference on Artificial Intelligence in Medicine (AIME–03) Workshop on Qualitative and Model-based Reasoning in Biomedicine*, S. Wierzchon, M. Klopotek, and M. Michalewicz, Eds. Protaras, Cyprus: Academic Publishing House EXIT, 19 October 2003, pp. 45–49.

[16] M. J. Druzdzel and A. Onisko, "The impact of overconfidence bias on practical accuracy of Bayesian network models: An empirical study," in *Working Notes of the 2008 Bayesian Modelling Applications Workshop, Special Theme: How Biased Are Our Numbers?, Part of the Annual Conference on Uncertainty in Artificial Intelligence (UAI–2008)*, Helsinki, Finland, 9 July 2008.

[17] ——, "Are Bayesian networks sensitive to precision of their parameters?" in *Intelligent Information Systems XVI, Proceedings of the International IIS'08 Conference*, S. Wierzchon, M. Klopotek, and M. Michalewicz, Eds. Warsaw, Poland: Academic Publishing House EXIT, Jun. 2008, pp. 35–44.

[18] Aníbal Pérez-Liñán, "*El dilema estratǵico de la seguridad ciudadana y el Estado democráico de derecho.*" In Alexandra Steinmetz (ed.), Informe Estado de la Región. State of the Nation—Region Program, Pavas, Costa Rica, http://www.estadonacion.or.cr/ Chapter 12, pages 460–524 Available at: http://www.estadonacion.or.cr/estadoregion2008/regional2008/capitulosPDF/Cap12.pdf