

AFRL-IF-RS-TR-2003-25
In-House Final Technical Report
February 2003



USING BAYESIAN NETWORKS AND DECISION THEORY TO MODEL PHYSICAL SECURITY

Nancy A. Roberts

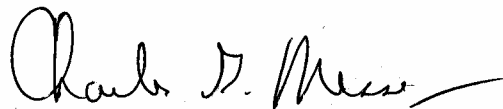
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

**AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK**

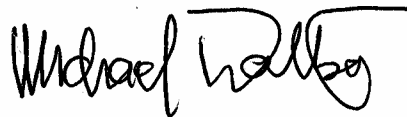
This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2003-25 has been reviewed and is approved for publication.

APPROVED:



CHARLES G. MESSENGER, Chief
Information Awareness & Understanding Branch
Information Technology Division



FOR THE DIRECTOR:

MICHAEL L. TALBERT, Maj., USAF
Technical Advisor
Information Technology Division

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE February 2003	3. REPORT TYPE AND DATES COVERED In-House Final, October 1995 – June 2002	
4. TITLE AND SUBTITLE USING BAYESIAN NETWORKS AND DECISION THEORY TO MODEL PHYSICAL SECURITY			5. FUNDING NUMBERS PE - 62702F PR - 5581 TA - 27 WU - 01	
6. AUTHOR(S) Nancy A. Roberts				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) AFRL/IFTB 525 BROOKS ROAD ROME, NY 13441-4505			8. PERFORMING ORGANIZATION REPORT NUMBER AFRL-IF-RS-TR-2003-25	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) AFRL/IFTB 525 BROOKS ROAD ROME, NY 13441-4505			10. SPONSORING / MONITORING AGENCY REPORT NUMBER AFRL-IF-RS-TR-2003-25	
11. SUPPLEMENTARY NOTES AFRL Project Engineer: Nancy Roberts/IFTB/315-330-3566				
12a. DISTRIBUTION / AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 Words) Home automation technologies allow a person to monitor and control various activities within a home or office setting. Cameras, sensors and other components used along with the simple rules in the home automation software provide an environment where the lights, security and other appliances can be monitored and controlled. These home automation technologies, however, lack the power to reason under uncertain conditions and thus the system can sometimes perform in ways the user didn't intend. The objective of this report is to apply Bayesian Networks and Decision Theory to this problem and thus, improve the accuracy.				
14. SUBJECT TERMS home automation software, Bayesian networks, decision theory, artificial intelligence			15. NUMBER OF PAGES 23	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL	

TABLE OF CONTENTS

Section 1: Introduction.....	1
Section 2: Scenario	1
Office Security	2
Section 3: Bayesian Modeling	2
Section 4: Decision Theory.....	6
Section 5: The Home Automation Environment	7
Section 6: Conclusions.....	9
Section 7: Future Work.....	10
Appendix A.....	11
Appendix B.....	14
Bibliography	18

LIST OF FIGURES

Figure 1: Singly Connected Graph 3
Figure 2: MSBNx Assessment..... 4
Figure 3: Office Security Model..... 4
Figure 4: MSBNx Evaluation 5
Figure 5: Multiply Connected Graph..... 6
Figure 6: Utility table..... 7
Figure 7: HomeSeer 8
Figure 8: MSAgent 8
Figure 9: Table of Selected Decisions 9
Figure 10: Prior probability of Day 11
Figure 11: Prior probability of Time..... 11
Figure 12: $P(\text{BackDoorBreakIn}|\text{Day})$ 11
Figure 13: $P(\text{FrontDoorBreakIn}|\text{Day, Time})$ 12
Figure 14: $P(\text{Janitor}|\text{Day, Time})$ 12
Figure 15: $P(\text{CeilingBreakIn}|\text{Time})$ 13
Figure 16: $P(\text{BackDoorSensor}|\text{BackDoorBreakIn})$ 13
Figure 17: $P(\text{FrontDoorSensor}|\text{Janitor, FrontDoorBreakIn})$ 13
Figure 18: $P(\text{CeilingSensor}|\text{CeilingBreakIn})$ 13

Section 1: Introduction

Advanced technology products have become so inexpensive that they are woven into all aspects of life today making tasks easier, and saving time and money. One of the hottest technology areas for the home is the home automation or smart home technology products such as the cameras, sensors, and other components used for monitoring and controlling various aspects of home life. Some of the current technologies allow a person to monitor and control the lights, security and other appliances (e.g., TV, stereo and coffee maker). These components can be used alone or grouped together to perform more complicated tasks.

Most of the home automation systems out there today allow a user to express rules to control their environment. One such use may be to turn on a light upon entering a room and turn off the light upon exiting. A person could set up a motion sensor at the doorway to the room. By triggering this sensor once, the light will turn on and then triggering it again will shut it off. This technology is not exact science though. What happens when a user triggers a motion sensor when they don't want to or when their cat triggers the sensor or when another person enters the room? Unless that is the effect that person was trying to achieve, it will just leave that person in the dark.

These home automation technologies lack the power to reason under uncertainty. The objective is to add intelligence to these technologies by applying Bayesian Networks and Decision Theory. These techniques will allow the user to model the problem not knowing every possible situation. The tools used in this project are

- MSBNx – Microsoft Belief Network Tool – used for Bayesian modeling,
- HomeSeer – home automation software,
- X10 sensors, transceiver and ActiveHome™ X10 computer interface - used to set up the sensor network, and
- Microsoft Visual Basic – used to access MSBNx and HomeSeer.

In the next section, there will be a description of the scenario used to illustrate the home automation domain. Section 3 will give an overview of Bayesian Networks and describe how to model the scenario. Section 4 will describe Decision Theory and how it is used on this problem. The home automation environment will be detailed in Section 5 and the conclusions will be covered in Section 6.

Section 2: Scenario

This scenario is based on a fictitious example using the home automation components to provide security to an office setting. However, this could realistically be

applied to security in the home or to control other aspects of the home automation domain.

Office Security

In this case, the office of Don Paranoid, the owner of Secureitall, Inc., is located on the 13th floor of a high-rise office building. He has recently added a safe filled with confidential documents in which he is the only one that knows the combination to the safe. Due to this new addition, Don is looking to add some security to his workplace.

The office has two doors (and no windows) which he has placed motion sensors next to. He is also worried that someone may try the same move as Tom Cruise in the movie “Mission Impossible” and come in through the ceiling, so he has placed a sensor by the ceiling. Normal activity consists of his normal work hours that he has chosen as 8:30 a.m. - 5:00 p.m. Monday through Friday. He also has to consider the case of when the janitor cleans the office. That action occurs pretty routinely 3 days a week between 5:00 p.m. and 6:00 p.m. Monday-Friday, but most often on Mondays. Any activity through the ceiling would be rare and should be considered abnormal, except on the occasion that the janitor would fix a light bulb. The placement of the sensors is such that the firing of any two sensors at the same time would be considered abnormal. This might mean a highly sophisticated coordinated attack in which the attackers are coming after him and then the safe. If only one sensor is triggered, then Don Paranoid has a chance to escape out the other door and call the police.

Section 3: Bayesian Modeling

Bayesian Networks allow a user to model a problem when full knowledge of the domain is incomplete or uncertain. A model is captured in a Bayesian Network as a directed a-cyclic graph with the nodes representing variables and the arcs representing the dependencies between those variables. The diagram illustrates the relations between nodes of how a particular variable causes another variable. In Figure 1, a simplified version of the security example shows how day of the week and time of day could influence whether there is a door break-in and thus trigger the door sensor. This type of model is known as a singly-connected graph [1]. This means that there is only one path between any two nodes. An example of a multiply-connected graph [1] is shown in Figure 5. Based on the type of graph, the system could have to do different types of calculations for the probabilities.

The particular tool used here to model a Bayesian Network is Microsoft Belief Networks (MSBNx). More details on MSBNx can be found in this report [5]. A user creates a new model and then right mouse clicks to add new nodes. Once a new node is

added and named, discrete values must then be assigned to that node. The values for Day are “Monday”, “Tuesday”, ... , and “Sunday”. The values assigned to Time are “0830-

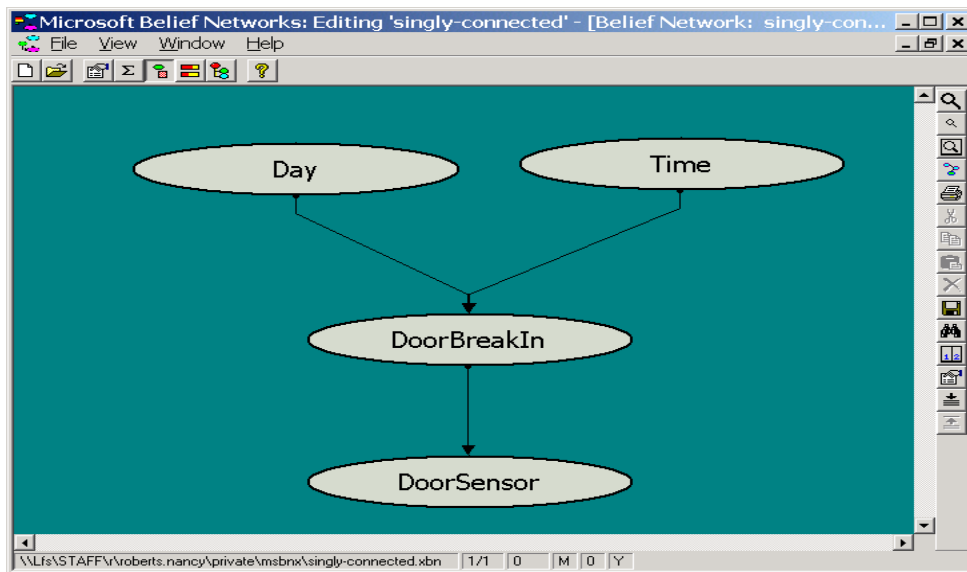


Figure 1: Singly Connected Graph

1700”, “1701-1800”, and “1801-0829”. Time was broken into convenient intervals instead of equal intervals. DoorBreakIn has values “Yes” and “No” and DoorSensor has values “On” and “Off”. The arcs are then added between these nodes showing the dependencies and usually flow from top to bottom.

The next step would be to add knowledge about what is known based on probabilities. These probabilities will then be used along with the model to infer things about the problem. Given the prior probabilities of the root nodes in a diagram and the conditional probabilities of the other nodes, MSBNx can then calculate all the other probabilities. To assign probabilities to a particular variable, click left on that node and select “Assess”. The probabilities assigned to $P(\text{DoorBreakIn}|\text{Day}, \text{Time})$ are shown in Figure 2. The probability assigned to $\text{DoorBreakIn}=\text{“Yes”}$ given $\text{Day}=\text{“Monday”}$ and $\text{Time}=\text{“0830-1700”}$ is equal to 0.05. This likelihood of break-in is low. However as Day changes to “Friday” and Time changes to “1801-1829”, the probability of break-in increases to 0.3.

Figure 3 shows the Bayesian Network model for the full Secureitall Inc. Office problem. The variables include Day, Time, FrontDoorBreakIn, BackDoorBreakIn, CeilingBreakIn, Janitor, FrontDoorSensor, BackDoorSensor, and CeilingSensor. In Appendix A, Figures 9 and 10 show the prior probabilities for Day and Time and Figures 11 through 17 show the conditional probabilities assigned to all the other nodes in the network.

Parent Node(s)		DoorBreakIn		
Day	Time	Yes	No	bar charts
Monday	0830-1700	0.05	0.95	
	1701-1800	0.08	0.92	
	1801-0829	0.1	0.9	
Tuesday	0830-1700	0.05	0.95	
	1701-1800	0.08	0.92	
	1801-0829	0.1	0.9	
Wednesday	0830-1700	0.05	0.95	
	1701-1800	0.08	0.92	
	1801-0829	0.1	0.9	
Thursday	0830-1700	0.05	0.95	
	1701-1800	0.08	0.92	
	1801-0829	0.1	0.9	
Friday	0830-1700	0.05	0.95	
	1701-1800	0.08	0.92	
	1801-0829	0.3	0.7	
Saturday	0830-1700	0.08	0.92	
	1701-1800	0.2	0.8	
	1801-0829	0.3	0.7	
Sunday	0830-1700	0.08	0.92	
	1701-1800	0.2	0.8	
	1801-0829	0.3	0.7	

Figure 2: MSBNx Assessment

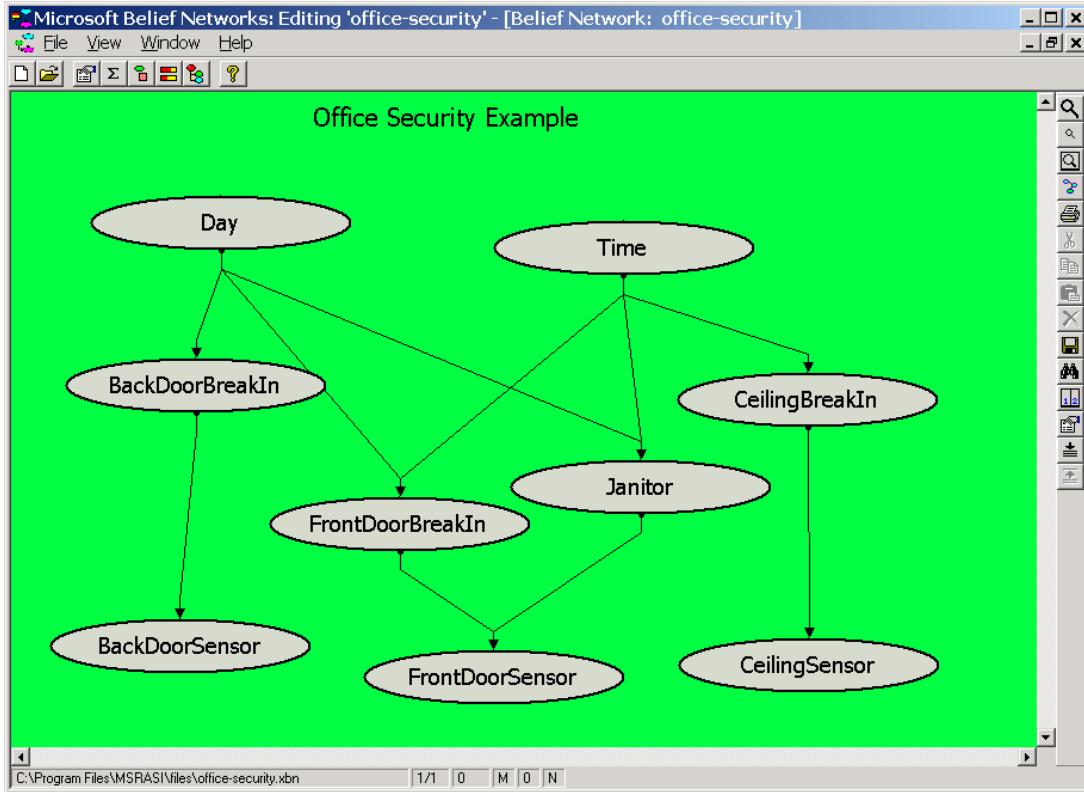


Figure 3: Office Security Model

MSBNx can now calculate probabilities given evidence that certain conditions hold. For example, what would be the probability of break-in given the Day happens to be “Wednesday”, the Time interval is “0830-1700” and the DoorSensor is “On”? MSBNx uses a formula based on Bayes Rule: $P(A|B) = P(B|A)/P(B)$ to calculate those probabilities for variables[5]. The results are shown in Figure 4. In this case, the calculation can be written out as follows:

$$1) P(\text{BreakIn}=\text{Yes}|\text{Day}=\text{Wednesday}, \text{Time}=\text{"0830-1700"}, \text{DoorSensor}=\text{"On"})$$

$$2) P(A|B) = P(A,B)/P(B):$$

$$3) P(BI|D,T,S) = P(D,T,S,BI)/P(D,T,S)$$

$$4) = \frac{P(D=\text{Wed})P(T=\text{0830-1700})P(BI=\text{yes}|D=\text{Wed},T=\text{0830-1700})P(S=\text{on}|BI=\text{yes})}{\sum_{i \in \{\text{yes,no}\}} P(D=\text{Wed})P(T=\text{0830-1700})P(BI_i|D=\text{Wed},T=\text{0830-1700})P(S=\text{on}|BI_i)}$$

$$5) = \frac{P(BI=\text{yes}|D=\text{Wed},T=\text{0830-1700})P(S=\text{on}|BI=\text{yes})}{\sum_{i \in \{\text{yes,no}\}} P(BI_i|D=\text{Wed},T=\text{0830-1700})P(S=\text{on}|BI_i)}$$

$$6) = 0.05(0.8)/(0.05*0.8 + 0.95*0.3) = 0.04/(0.04+0.285) = 0.04/0.325 = 0.1231$$

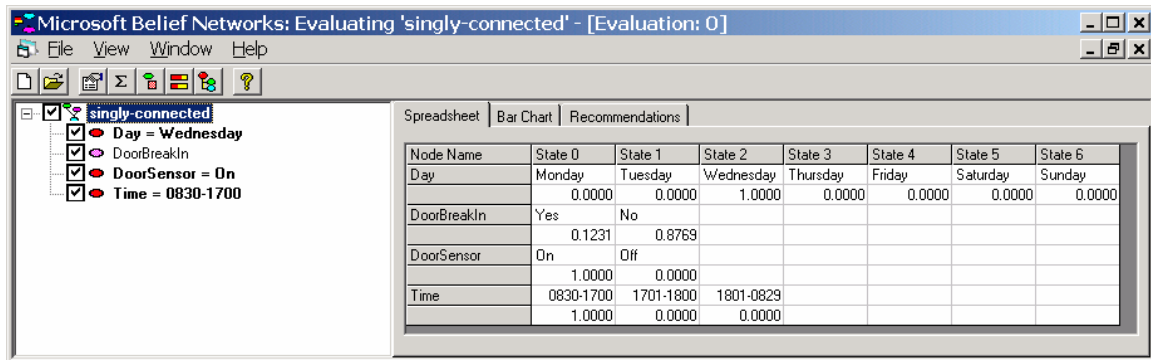


Figure 4: MSBNx Evaluation

All calculations are not calculated the same. Diagrams of different shapes use different algorithms to calculate the probabilities. Figure 5 is an example of a network that is multiply-connected and thus will use a “noisy-or” calculation [1] for its probability calculation. It is multiply-connected because there are two paths from Day to WindwSensor. Fortunately, MSBNx can do these different calculations.

Because there could possibly be three types of break-in’s in the Secureitall Inc. example, the calculation for an overall break-in includes all three and is given below.

$$\begin{aligned}
 P(\text{Break In}) &= P(\text{FDBI}) \vee P(\text{BDBI}) \vee P(\text{CBI}) \\
 &= P(\text{FDBI}) + P(\text{BDBI}) + P(\text{CBI}) - P(\text{FDBI} \wedge \text{BDBI}) - P(\text{FDBI} \wedge \text{CBI}) - P(\text{BDBI} \wedge \text{CBI}) \\
 &\quad + P(\text{FDBI} \wedge \text{BDBI} \wedge \text{CBI}) \\
 &\text{where FDBI is FrontDoorBreakIn, BDBI is BackDoorBreakIn, and CBI is CeilingBreakIn}
 \end{aligned}$$

But because the values of the combinations of probabilities such as $P(FDBI \wedge BDBI)$, $P(FDBI \wedge CBI)$, $P(BDBI \wedge CBI)$ and $P(FDBI \wedge BDBI \wedge CBI)$, are not known, the overall probability will be approximated and expressed as an interval. The maximum value for break-in will be when all of the unknown probabilities: $P(FDBI \wedge BDBI)$, $P(FDBI \wedge CBI)$, $P(BDBI \wedge CBI)$ and $P(FDBI \wedge BDBI \wedge CBI)$ are equal to zero. But that number could possibly be greater than one, so the maximum value is the minimum of 1 and that number. The minimum value for break-in would be the maximum of $P(FDBI)$, $P(BDBI)$ and $P(CBI)$. The intervals for break-in and no break-in are shown below:

$$1) P(\text{Break In}) = [\max(P(FDBI), P(BDBI), P(CBI)), \min(1, P(FDBI) + P(BDBI) + P(CBI))]$$

$$2) P(\text{No Break In}) = [1 - \min(1, P(FDBI) + P(BDBI) + P(CBI)), 1 - \max(P(FDBI), P(BDBI), P(CBI))]$$

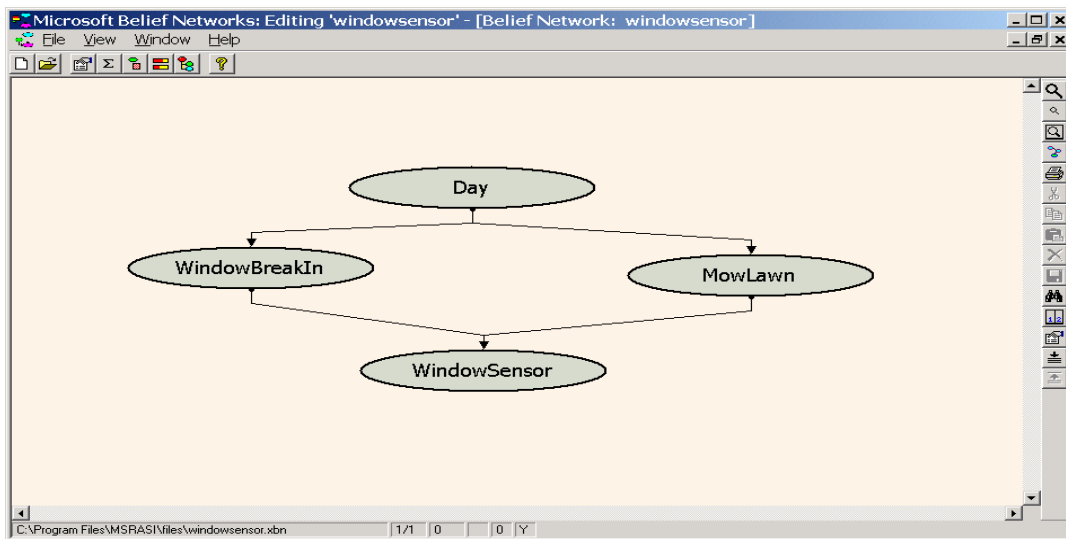


Figure 5: Multiply Connected Graph

Section 4: Decision Theory

The output of the MSBNx tool of whether there is a break-in or not from the MSBNx tool will be a probability. Based on that, how does one decide what to do? The answer is that one can make an educated decision based on decision theory about the security of the office from that probability.

There are three components of a decision: acts, states and outcomes [9]. In this case, the acts are “Ring Alarm” or “Don’t Ring Alarm” and the states are “Break-In” and “No Break-In”. In this case, there are also two possible outcomes based upon whether it is believed that there is a break-in or not: 1) Ring the alarm, or 2) Don’t ring the alarm.

A decision will be based on maximizing expected utility where $EU(a) = \sum_{s \in \text{states}} u(a,s)p(s)$, where a is an act [8]. The utility table for this example is shown in Figure 6. The figures in the table are based upon how desirable or tolerable a person would be towards the alarm ringing. In this case, Don Paranoid is very tolerable towards an alarm ringing just so there isn't a break-in.

	BreakIn	No BreakIn
Ring Alarm	1.0	0.3
Don't Ring Alarm	0	0.8

Figure 6: Utility table

In this case, the $P(\text{BreakIn})$ is calculated as an interval so the calculation of EU gets a little more complex and is shown below:

To calculate $EU(A)$:

If $u(A, BI) > u(A, NBI)$

$$\min EU(A) = u(A, BI) * \min P(BI) + u(A, NBI) * (1 - \min P(BI))$$

$$\max EU(A) = u(A, BI) * \max P(BI) + u(A, NBI) * (1 - \max P(BI))$$

else

$$\min EU(A) = u(A, NBI) * \min P(NBI) + u(A, BI) * (1 - \min P(NBI))$$

$$\max EU(A) = u(A, NBI) * \max P(NBI) + u(A, BI) * (1 - \max P(NBI))$$

Utility Interval for Ring Alarm(A) = [min EU(A), max EU(A)]

Utility Interval for Don't Ring Alarm(NA) = [min EU(NA), max EU(NA)]

The decision is based on max of (min EU(A), min EU(NA))

Section 5: The Home Automation Environment

The environment is made up of the HomeSeer Version 1.5 software, Microsoft Visual Basic(VB), X10 sensors and transceivers, and the ActiveHome™ X10 computer interface. HomeSeer Version 1.5 is the home automation tool used to help with the monitoring and controlling of the environment. This tool, shown in Figure 7, allows a user to specify devices used and events or rules applied to those devices. The sensors will send signals back to the transceivers and then on to the computer. The HomeSeer tool can perform many more tasks than what will be described here. For more details, see their users manual [6].

In this environment, there are three devices to monitor: sensor1 (FrontDoorSensor), sensor3 (BackDoorSensor) and sensor5 (CeilingSensor). The rule or event defined is to check and report back the status of the sensors every minute. This event then triggers a VBscript that basically calls the Visual Basic program. This program is detailed in Appendix B. This program has as input the status of the three

sensors. The program then accesses the MSBNx probabilities and comes up with individual values for break-in. The VB program then calculates the probability interval. Then based on this, calculates the decision.



Figure 7: HomeSeer

Depending on the result of the calculation, the system will perform the appropriate action. In this example, HomeSeer calls upon a Microsoft Agent that speaks an action of whether to take precaution against an attack or not. An example of a MSAgent is shown in Figure 8. By triggering one of the sensors during the daytime on a weekday, the agent will speak the phrase, "All is normal." However, if all three sensors are triggered at 3:00 a.m. on a weekend, the agent will say, "Ring Alarm. Check for Break-In."

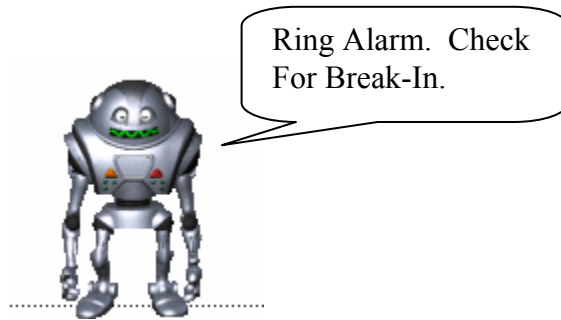


Figure 8: MSAgent

Section 6: Conclusions

A table of selected decisions based on certain conditions is shown in Figure 9. Line #1 shows the simple case when all the sensors are off during a normal workday so the system does nothing and Line #5 is when all the sensors are on during after hours on the weekend so the system rings an alarm. This behavior seems very predictable and could easily be modeled with the simple production rules that come with HomeSeer. Line #1 could be mapped to If Sensor1=Off and Sensor2=Off and Sensor3=Off then “Do Nothing”. Line #5 could be expressed as If Day=Saturday or Day=Sunday and Time=1801-0829 then “Ring Alarm”. But now look at Lines #2, #3, and #4, they start to get more complex. In Line #2, #3, and #4, the back door sensor is the only sensor off. The decision for Line #2 is to “Do Nothing” where the other two are to “Ring Alarm”. In this case, it is not as simple as weekday and time. It is based on the knowledge that the janitor is more likely to be working and possibly triggering the ceiling sensor on Monday than any other day of the week. This can be better represented as probabilities.

	Sensor1 (Front)	Sensor3 (Back)	Sensor5 (Ceiling)	Time	Day	Decisions
1)	Off	Off	Off	0829-1700	Monday	Do Nothing
2)	On	Off	On	1701-1800	Monday	Do Nothing
3)	On	Off	On	1801-0829	Monday	Ring Alarm
4)	On	Off	On	1701-1800	Friday	Ring Alarm
5)	On	On	On	1801-0829	Sunday	Ring Alarm

Figure 9: Table of Selected Decisions

One can see the usefulness of a home automation system. Alone they add value to the environment that they are trying to improve. The fact is that having sensors turn on lights only when needed can help save people money. However, the components with the simple rules are inadequate to model every state that this environment could possibly be in resulting in a frustrated user or possibly more money spent because the light switch needed to be turned on multiple times. The use of Bayesian Networks and Decision Theory are the right tools to enhance that environment to make it more accurate. The results were demonstrated in the simple example of office security.

Section 7: Future Work

The example given above demonstrates the application of Bayesian Networks and Decision Theory to COTS (commercial off the shelf) products to provide additional security to an office setting. One of the next steps would be to apply these techniques to a more robust application domain. Here is one possible area where it could be further applied: Interactive Data Wall application.

The AFRL Interactive Data Wall is an ultra high-resolution wall display that allows for the presentation and interactive control of an endless stream of information arriving from a diverse collection of sensors deployed on a variety of platforms. This environment allows for the interaction of multiple users with the display. However, for example, two people wanting to use a red laser pointer at the same time could cause problems. The system has an inability to differentiate between multiple red laser pointers[2]. Bayesian Networks and Decision Theory could be applied to this area providing improved accuracy in various aspects of multiple simultaneous users such as speech recognition, multiple cursors and multiple laser pointer tracking.

Another area the data wall could benefit has to do with intelligent interactive agents based on Bayesian Networks and Decision Theory. Much the same way as Microsoft Agents provide assistance with Microsoft software, these agents could provide assistance by displaying contextually relevant information from multiple applications. These assistants with the help of motion sensors, and cameras could also help to distinguish the multiple users of the data wall providing intelligent support for the commander.

Appendix A

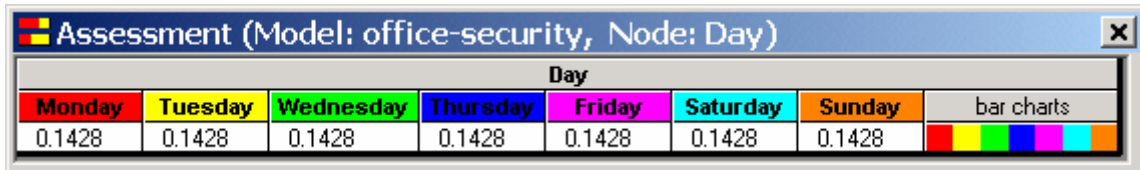


Figure 10: Prior probability of Day

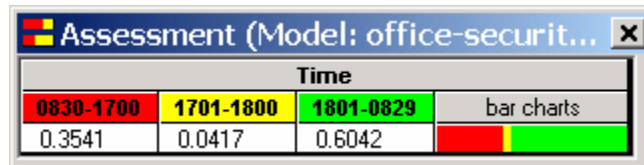


Figure 11: Prior probability of Time

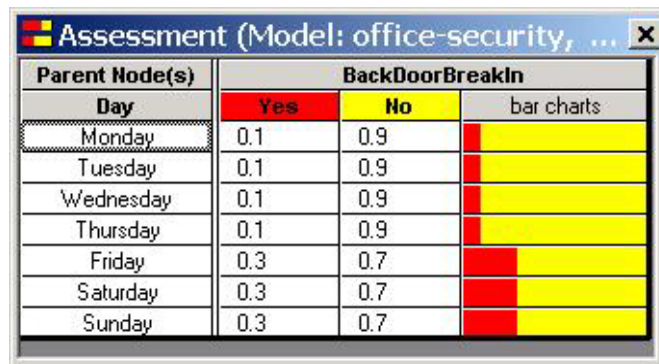


Figure 12: $P(\text{BackDoorBreakin}|\text{Day})$

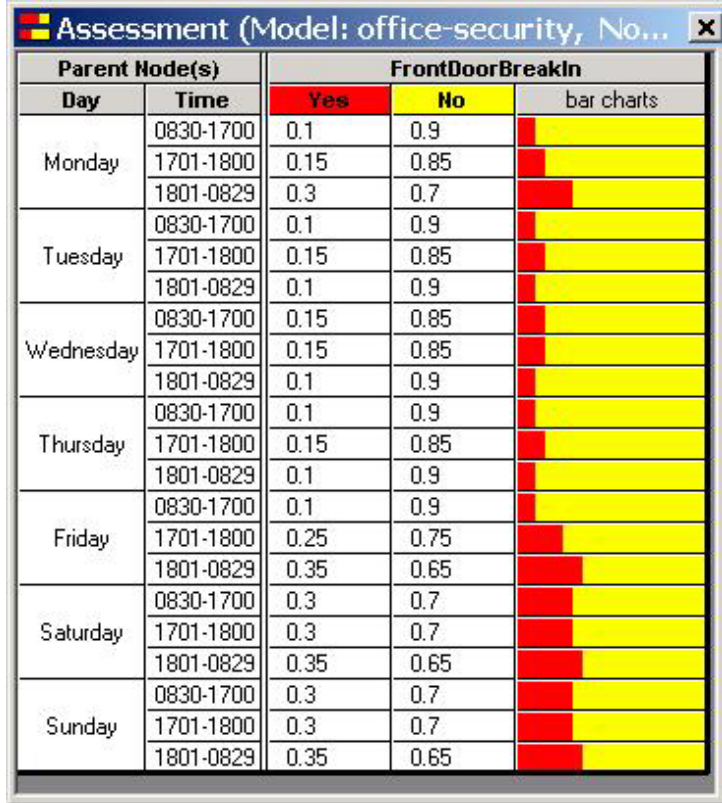


Figure 13: $P(\text{FrontDoorBreakIn}|\text{Day, Time})$

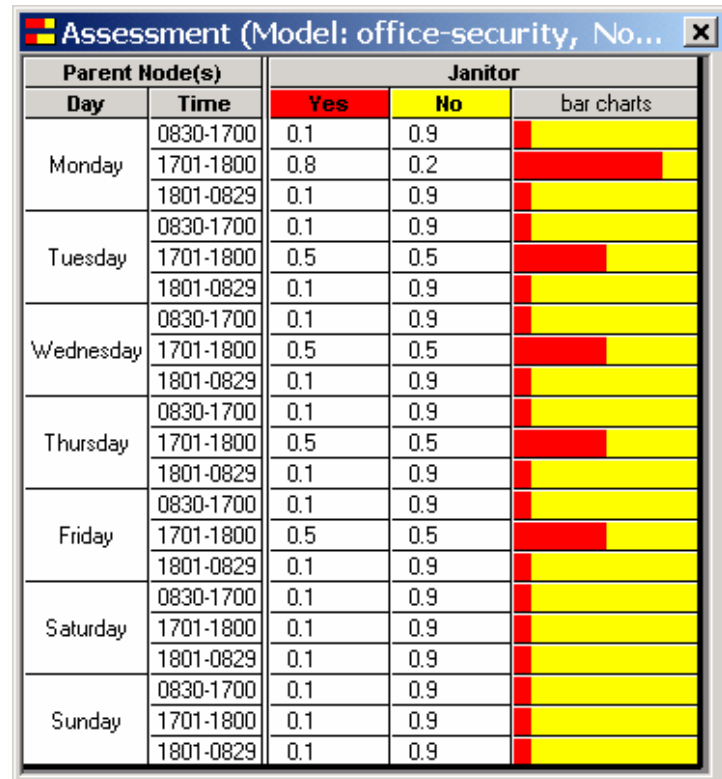


Figure 14: $P(\text{Janitor}|\text{Day, Time})$

Assessment (Model: office-security, ...)			
Parent Node(s)		CeilingBreakIn	
Time	Yes	No	bar charts
0830-1700	0.01	0.99	
1701-1800	0.05	0.95	
1801-0829	0.1	0.9	

Figure 15: $P(\text{CeilingBreakIn}|\text{Time})$

Assessment (Model: office-security, N...)			
Parent Node(s)		BackDoorSensor	
BackDoorBreakIn	On	Off	bar charts
Yes	0.9	0.1	
No	0.2	0.8	

Figure 16: $P(\text{BackDoorSensor}|\text{BackDoorBreakIn})$

Assessment (Model: office-security, Node: F...)				
Parent Node(s)		FrontDoorSensor		
Janitor	FrontDoorBreakIn	On	Off	bar charts
Yes	Yes	0.9	0.1	
	No	0.4	0.6	
No	Yes	0.8	0.2	
	No	0.1	0.9	

Figure 17: $P(\text{FrontDoorSensor}|\text{Janitor}, \text{FrontDoorBreakIn})$

Assessment (Model: office-security, N...)			
Parent Node(s)		CeilingSensor	
CeilingBreakIn	On	Off	bar charts
Yes	0.95	0.05	
No	0.2	0.8	

Figure 18: $P(\text{CeilingSensor}|\text{CeilingBreakIn})$

Appendix B

Option Explicit

'Office Security Scenario - December 2001

Sub TestSecurity()

' Access HomeSeer Functionality

Dim hs As HomeSeer.Application

Set hs = New HomeSeer.Application

'Create a Log File

Dim fso As New FileSystemObject, a As TextStream

Set a = fso.OpenTextFile("c:\program files\msrasi\files\office-security.txt",
ForAppending)

a.WriteLine ("-----")

Dim statusA1, statusA3, statusA5

Dim pstatus, pstatus3, pstatus5

'Get Status of Devices

statusA1 = hs.DeviceStatus("A1")

statusA3 = hs.DeviceStatus("A3")

statusA5 = hs.DeviceStatus("A5")

If statusA1 = 2 Then pstatus = "On" Else If statusA1 = 3 Then pstatus = "Off" Else If
statusA1 = 4 Then pstatus = "Dim" Else If statusA1 = 17 Then pstatus = "Unknown"

If statusA3 = 2 Then pstatus3 = "On" Else If statusA3 = 3 Then pstatus3 = "Off" Else If
statusA3 = 4 Then pstatus3 = "Dim" Else If statusA3 = 17 Then pstatus3 = "Unknown"

If statusA5 = 2 Then pstatus5 = "On" Else If statusA5 = 3 Then pstatus5 = "Off" Else If
statusA5 = 4 Then pstatus5 = "Dim" Else If statusA5 = 17 Then pstatus5 = "Unknown"

hs.Speak "This is the status of the sensors."

hs.Speak "A1 is " + pstatus

hs.Speak "A3 is " + pstatus3

hs.Speak "A5 is " + pstatus5

a.WriteLine "A1 is " + pstatus

a.WriteLine "A3 is " + pstatus3

a.WriteLine "A5 is " + pstatus5

```

Dim aMSBN As New MSBN3Lib.MSBN
'Load the Sensor model
Dim modelSensor As MSBN3Lib.Model

'Access MSBNx file
Set modelSensor = aMSBN.Models.Add(FileName:="c:\Program
Files\MSRasi\files\office-security.xbn", ErrorFilename:="errorfile.log")

Dim nodeFrontDoorSensor As MSBN3Lib.Node
Set nodeFrontDoorSensor = modelSensor.ModelNodes("FrontDoorSensor")

Dim nodeFrontDoorBreakIn As MSBN3Lib.Node
Set nodeFrontDoorBreakIn = modelSensor.ModelNodes("FrontDoorBreakIn")

Dim nodeBackDoorSensor As MSBN3Lib.Node
Set nodeBackDoorSensor = modelSensor.ModelNodes("BackDoorSensor")

Dim nodeBackDoorBreakIn As MSBN3Lib.Node
Set nodeBackDoorBreakIn = modelSensor.ModelNodes("BackDoorBreakIn")

Dim nodeCeilingSensor As MSBN3Lib.Node
Set nodeCeilingSensor = modelSensor.ModelNodes("CeilingSensor")

Dim nodeCeilingBreakIn As MSBN3Lib.Node
Set nodeCeilingBreakIn = modelSensor.ModelNodes("CeilingBreakIn")

'Call the model's inference engine "inferSensor"
Dim inferSensor As MSBN3Lib.Engine
Set inferSensor = modelSensor.Engine

Dim wd, mon, t, mth, t1

wd = WeekdayName(Weekday(Now))
mon = Month(Now)
t1 = Format(Now, "hhmm")

Debug.Print "Weekday:", wd, "Time: ", t1

If (t1 >= 830 And t1 <= 1700) Then t = "0830-1700" Else If (t1 >= 1701 And t1 <= 1800)
Then t = "1701-1800" Else t = "1801-0829"
Debug.Print t

'The sensors actually give 4 outputs: On, Off, Dim and Unknown.
If (pstatus = "Dim" Or pstatus = "Unknown") Or (pstatus3 = "Dim" Or pstatus3 =
"Unknown") Or (pstatus5 = "Dim" Or pstatus5 = "Unknown") Then
hs.Speak "Weird reading from sensor. Check it out"

```

```

Else
inferSensor.Evidence.Add "Day", wd
inferSensor.Evidence.Add "Time", t
inferSensor.Evidence.Add "FrontDoorSensor", pstatus
inferSensor.Evidence.Add "BackDoorSensor", pstatus3
inferSensor.Evidence.Add "CeilingSensor", pstatus5

a.WriteLine "Weekday: " + wd + " at time " + t1

Dim Pfdbi, Pbdbi, Pcbi, Alarm, minEUAlarm, maxEUAlarm, NoAlarm,
minEUNoAlarm, maxEUNoAlarm
Dim ABI, ANBI, NABI, NANBI, BI, lb, ub, n

Pfdbi = inferSensor.Belief("FrontDoorBreakIn", "Yes")
Pbdbi = inferSensor.Belief("BackDoorBreakIn", "Yes")
Pcbi = inferSensor.Belief("CeilingBreakIn", "Yes")

'Calculate the Expected Utility Interval
' Calculate the Pbi = Pfdbi or Pbdbi or Pcbi = Pfdbi + Pbdbi + Pcbi - P(fdbi and bdbi and
Pcbi)
' Which leads to the interval: [lb= max(Pfdbi, Pbdbi, Pcbi), ub= min(1, Pfdbi + Pbdbi +
Pcbi)]

If Pcbi <= Pfdbi >= Pbdbi Then lb = Pfdbi Else If Pcbi <= Pbdbi >= Pfdbi Then lb =
Pbdbi Else lb = Pcbi
If (Pfdbi + Pbdbi + Pcbi) > 1 Then ub = 1 Else ub = (Pfdbi + Pbdbi + Pcbi)

'Check a probability
Debug.Print Pfdbi, Pbdbi, Pcbi, lb, ub

'Desirabilities
ABI = 1.0
ANBI = 0.3
NABI = 0
NANBI = 0.8

'Calculate utility interval for Alarm and No Alarm

If ABI > ANBI Then
minEUAlarm = ABI * lb + ANBI * (1 - lb)
maxEUAlarm = ABI * ub + ANBI * (1 - ub)
Else
minEUAlarm = ANBI * (1 - ub) + ABI * ub

```

```
maxEUIAlarm = ANBI * (1 - lb) + ABI * lb
End If
```

```
If NABI > NANBI Then
minEUNoAlarm = NABI * lb + NANBI * (1 - lb)
maxEUNoAlarm = NABI * ub + NANBI * (1 - ub)
Else
minEUNoAlarm = NANBI * (1 - ub) + NABI * ub
maxEUNoAlarm = NANBI * (1 - lb) + NABI * lb
End If
```

```
a.Write "EU(Alarm) = "
a.Write "["
a.Write minEUIAlarm
a.Write ", "
a.Write maxEUIAlarm
a.WriteLine "]"
a.Write "EU(NoAlarm) = "
a.Write "["
a.Write minEUNoAlarm
a.Write ", "
a.Write maxEUNoAlarm
a.WriteLine "]"
```

'Speak action and write to log file.

```
If (minEUIAlarm > minEUNoAlarm) Then
hs.Speak "Ring Alarm. Check for breakin."
a.WriteLine "Ring Alarm. Check for breakin"
Else
hs.Speak "All is normal."
a.WriteLine "All is normal"
End If
```

End If

a.Close

End Sub

Bibliography

[1] Charniak, Eugene. “*Bayesian Networks without Tears*”, AI Magazine, Winter 1991, pp-50-63.

[2] Jedrysik, Peter, Jason Moore, 1 Lt Mark Brykowsytch and Richard Sweed. “*The Interactive Data Wall*”, Command and Control Research and Technology Symposium US Naval War College, Rhode Island, Proceedings June 29- July 11999.

[3] Jeffrey, Richard C. *The Logic of Decision*, The University of Chicago Press 1983.

[4] Jensen, Finn V. *An Introduction to Bayesian Networks*, Springer-Verlag, New York 1996.

[5] Kadie, Carl M., David Hovel and Eric Horvitz. “*MSBNx: A Component-Centric Toolkit for Modeling and Inference with Bayesian Networks*”, Technical Report – MSR-TR-2001-67, 28 July 2001.

[6] *HomeSeer User’s Guide Version 1.5., 2001 HomeSeer Technologies LLC.*

[7] Pearl, Judea. *Probabilistic Reasoning in Intelligent Systems*, Morgan Kaufmann Publishers, 1988.

[8] Pittarelli, Michael. “*Decisions with Probabilities over Finite Product Spaces*”, IEEE Transactions on Systems, Man and Cybernetics, Vol. 21, No.5, September/October 1991.

[9] Resnik, Michael D. *Choices: An Introduction to Decision Theory*, The University of Minnesota Press 1987.