MaSO:A Tool for Aiding the Management of Schedule Overrun

Kawal Jeet[1], Renu Dhir[2] ,Vijay Kumar Mago[3] and Rajinder Singh Minhas[4]

[1&3]Department of Computer Science, DAV College, Jalandhar, India
[2]Department of Computer Science & Engineering, NIT, Jalandhar, India
[4]Department of Computer Science, MLUDAV College, Phagwara, India

[1]kawaljeet80@yahoo.com,[2]dhirr@nitj.ac.in,[3]vijay.mago@gmail.com,[4]minhas_rajinder@yahoo.com

Abstract*—Curbing schedule slippage is a daunting task in software industry. This problem evolves mainly because of poor analysis of risk factors and their management. This paper aims to handle this predicament with the help of Influence Diagram (ID). The three main risk factors that adversely affect the schedule are creeping user requirements, requirement instability and use of unnecessary features in the project. An integration of impacts of these with experts' opinion and with the existing databases (consisting of probability of occurrence of risk factors) helped us to create an ID based system that is capable to model the schedule slippage. This system can be used by the software manager at any stage of software development.

Keywords- Influence Diagram; Schedule Overrun; Software Developmen;, Schedule Management; Risk Management; Bayesian Network; K2 Algorithm.

I. INTRODUCTION

Software development projects (SDP) often exceed time and budgetary allocations and sometimes even fail to meet the user requirements. There are several factors that are responsible for this failure [1]. Some of these are: low productivity, creeping user requirements, and lack of client support, incompetent staff, staff training, and many more. These are called risk factors because they have potential to cause threat to the success of a project but have not yet happened [2].

One of the reasons for the failure of a SDP is delay in their delivery (also called schedule overrun). Some of the key factors responsible for this delay are [3-5] creeping user requirements; requirement instability and involvement of unnecessary features, staff training etc. According to [6], 60% of the projects fail due to this delay or schedule overrun. So it must be taken seriously.

Number of subjective techniques were developed and used in software industry to manage the schedule and thus timely delivery of projects. But they lack strong scientific theory and clarity. We have developed a tool named 'Management of Schedule Overrun' (MaSO) which is based on theory of Risk Management [8] and utilizes the Influence Diagram ID [7].

Risk Management, includes the process of assessing, identifying, analyzing, and eliminating the potential problems before it leads to failure of a project [8]. Risk management is the process of applying appropriate tools and methods to keep the risk within tolerable limits. It consists of several sub-activities such as:

- *Risk Identification*
- *Risk analysis*
- *Risk Planning*
- *Risk Tracking*
- *Risk Control*

The focus of this research is *risk prioritization* which is a sub activity of risk analysis phases. It helps in reducing the number of failed SDP by early identification of high risk elements and ID has been used for the same.

A. Influence Diagram

An ID also called a decision network is a compact graphical and mathematical method which is used to represent a decision situation as discussed in [9].The modeling is based on probabilistic theory. According to [10], the network contains chance nodes which are usually drawn as circles or ovals. There is a utility node usually drawn as a diamond and represents the desirability of different event combinations involved in the network. ID is actually an enhancement of Bayesian Networks BN which is a probabilistic network used for reasoning under uncertainty but can be constructed using chance nodes only.

Normally, an arc in an ID denotes an influence. For instance if there is an arc from A to C it means the node at the tail (A or B in Figure 1) of the arc influences the value (or the probability distribution over the possible values) of the node at the head (C in Figure 1) of the arc.

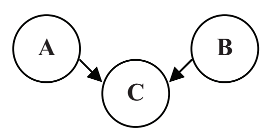

Figure 1 Cause-effect relationship

According to Baye's Theorem [11]

$$P(C \mid A) = \frac{P(A \mid C).P(C)}{P(A)} \qquad (1)$$

In this formula C is the hypothesis required to be tested. A is the evidence that confirms or disconfirms the hypothesis.

II. MOTIVATION

Since the early 80's, IDs have been used in a wide variety of applications like medical diagnosis [12], management of training of the staff involved in development of a software project [13], etc.

With respect to the present research area, many empirical and opaque studies have been done for the management of schedule of a SDP [5], [14]. The authors of [15-16] describe the way to compress the schedule of a SDP. Overall, these studies provided illuminating insights into schedule management but are weak in explaining its true impact and their management.

In this paper we identify the risk factors having an impact on schedule of a project and by using ID, exact delay in schedule is calculated.

III. DESIGN OF MASO

Schedule overruns 'always' occurred in 1%, 'usually' in 31%, 'sometimes' in 50%, 'rarely' in 15%, and, 'never' in 2% of the organizations [17]. According to [18], and based on extensive interviews that we have conducted with 34 software professionals, it has been identified that the following risk factors have more adverse effects on schedule of a SDP than others. A brief description of these factors are as follows:

- Creeping User Requirement: User keeps on posing the requirements about the project in fragments throughout the system development [14].
- Unnecessary features: Adding more functionality/ features than actual requirements [15].
- Requirement Instability: User is not sure about the requirements and keeps on changing the statements [16].

The aim of designing MaSO using ID is to calculate the delay in the delivery of the project due to the risk factors mentioned above. These risk factors are presented as chance nodes as shown in Figure 2. 'Schedule Overrun' is dependent on the probability of occurrence of these risk factors and is also represented as a chance node. Slippage in schedule is represented by a utility node and is driven by the parent node 'Schedule Overrun'. It gives delay (in months) that can happen in delivery of project due to the parent risk factors.

A. *Use of SMILE*

SMILE and java has been used to develop MaSO. SMILE is developed by Decision Systems Laboratory (DSL) [10]. It provides platform independent library built in C++ classes for reasoning in BNs and IDs. SMILE libraries can be accessed from within Java applications by using a Java Native Interface (JNI) library called jSMILE (used to make ID). The interface for MaSO is developed by using NETBEANS (an IDE for java) [19].

B. *Measurement scale for Influence*

For each node (of ID) of MaSO, a measurement scale i.e. a categorization (such as probable, possible etc.) of possible outcomes is required. This categorization is required to be as close as possible to the way the management conducts assessment in that organization, so that MaSO fits well into the organization. With the help of experts, following five categories have been identified for the nodes discussed in the beginning of section 3. These are called *possible outcomes*.

Frequent: If the risk factor is found to occur very often.
Probable: If it is found to occur less frequently.
Occasional: If it is found to occur at a normal frequency.
Remote: If it is found to occur less.
Improbable: If the risk rarely occurs in the present SDP.

Since all the risk factors do not have the same impact on schedule overrun, so severity level of each risk factor has to be identified. These impact values are to be developed based on experiences and past historic data to suit the given organization, risk factor, people and environment.

According to extensive interviews from 34 software professionals and with reference to [2], the default scale for the above is compiled to be:

Catastrophic: If the risk factor has very severe consequence.
Critical: If consequence is lesser severe.
Serious: If consequence or loss due to the risk factor is normal.
Minor: If loss is less.
Negligible: If loss is least.

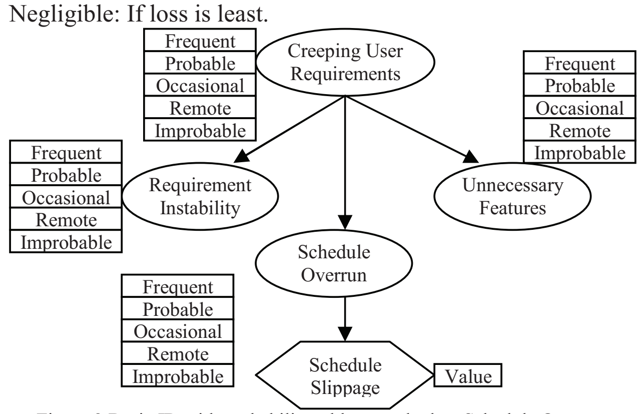

Figure 2 Basic ID with probability tables to calculate Schedule Overrun

C. *MaSO Automation*

By using NETBEANS (for java), jSMILE (to construct ID) and SMILE wrappers (for dealing with databases), MaSO model has been developed with the following functionalities.

Project managers (users) can enter the impacts of all three risk factors responsible for delaying the project deliveries, as shown in Figure 3. After accepting inputs from the user, MaSO asks for another set of inputs as shown in Figure 4. Using this interface, the user can set the probability/likelihood of occurrence of the risk factors in the project.

In the background, MaSO generates the ID (.xdsl file) and schedule overrun is displayed as shown in Figure 2.

D. Generating conditional probability table

In order to generate ID, conditional probability tables (CPT) are generated for the node 'Schedule Overrun' on the basis of impacts of each risk factor as shown in Figure.3. These impacts are normalized before being used for populating the CPT of 'Schedule Overrun'. This means the relative strengths of the influences on the child node is calculated [20]. This can be done using function normalized as discussed below.

Function normalize is
Input: Impact of each risk factor involved i1,i2,…,in
Output: Normalized weights of given risk factors w1,w2,…..wn
For each of the risks involved, ti is assigned value on the basis of the severity of its impact.
t_i=5 if impact is catastrophic
t_i =4 if impact is critical
t_i =3 if impact is severe
t_i =2 if impact is minor
t_i =1 if impact is negligible
where i=1,2,3
Relative weight of risk factor (i) $w_i = t_i / \sum t_i$
end function

For all the risk factors i=1, 2,…n, the relative weights wi comes out to be in the range [0,1]. Sum of relative weights of all the risk factors comes out to be 1.

$$0 \le w_i \le 1$$
$$w_1 + w_2 + ...+ w_n = 1 \qquad (2)$$

For instance if the impact of various risk factors on delay in schedule is as entered in Figure 3 then relative or normalize weights are calculated as discussed below. For 'Creeping User Requirements' the impact is entered to be 'Catastrophic'. So as discussed in function normalize t1 = 5. Similarly, for 'Unnecessary features' the impact is entered to be 'Critical'. So t2=4. For 'Requirement Instability' the impact is entered to be 'Minor'. So t3=2. Corresponding relative weights calculated by using function normalize is

w1=t1/(t1+t2+t3)=5/(5+4+2)=5/11
w2=t2/(t1+t2+t3)=4/(5+4+2)=4/11 (3)
w3=t3/(t1+t2+t3)=2/(5+4+2)=2/11

After getting normalized weights we can countercheck the weights by calculating $\sum_{i=1}^{3} w_i$ which comes out to be w1+w2+w3 (calculated above) = 5/11+4/11+2/11=11/11=1 as discussed in (2).

CPT for 'Schedule Overrun' can be generated by using K2 algorithmic technique [21]. K2 is an algorithm for constructing a BN from a database of records. For the current ID, this database (shown in table 1) is built by expert interviews and past historic data. Each row of this table indicated the probability of occurrence of all the risk factors and 'Seclude Overrun' in a particular case.

In order to calculate the joint CPT for occurrence of 'schedule Overrun' with respect to all the risk factors occurring simultaneously [22], temporarily ID's and hence CPT's are to be constructed for each individual relationship (indicated in Figure 2) by using K2 technique.

CPT is developed for: 'Creeping User Requirements'---'Schedule Overrun', 'Unnecessary Features----'Schedule Overrun', and 'Requirement Instability'---'Schedule Overrun' as shown in Table 2-4. Let *c1, c2, c3* denotes matrices holding the CPT (as shown in Table 2, 3 and 4) of child nodes from each of the temporary ID's built as mentioned above. CPT for 'Schedule Overrun' can be calculated by using following formula. According to [28],

$$p(x^q \mid y1^{q1}, y2^{q2}, y3^{q3}) = \sum_{j=1}^{3} w^j p(x^q \mid y_j^{qj}) \qquad (4)$$

where q,q_j = {Frequent, Probable, Occasional, Remote and Improbable}.
j=1,2,3
x=Schedule Overrun
y1=Creeping User Requirement
y2=Requirement Instability
y3=Unnecessary Features

w_j is normalized impact of risk factor j as obtained by function normalize discussed in section 3.3.

It means, joint probability of occurrence of a possible outcome for x, provided y1,y2,y3 has possible outcomes to be q1,q2,q3 respectively is equal to sum of normalized impact of each risk factor(calculated above) multiplied by conditional probability of occurrence of with respect to its corresponding risk factor.

For instance, from table 2, it is obtained that if probability of occurrence of 'Creeping user requirement' is 'Probable' then probability of occurrence of 'Schedule Overrun' to be occasional is 0.25. From table 3, it is obtained that if probability of occurrence of 'Unnecessary Features' is 'Improbable' then probability of occurrence of 'Schedule

Overrun' to be occasional is 0.25. From table 4, it is obtained that if probability of occurrence of 'Requirement Instability' is ''Remote' then probability of occurrence of 'Schedule Overrun' to be occasional is 0.125. As mentioned in section 3.4 the normalized weights w1, w2 and w3 will be 5/11, 4/11, 2/11.So by using (4)

$$p(x^{Occasional} \mid y1^{\Pr obable}, y2^{\mathrm{Im}\, probable}, y3^{remote}) =$$
$$5/11*0.25+4/11*0.25+2/11*0.125=$$

0.113636363636363636363636363636+
0.0909090909090909090909090909090901+
0.0227272727272727272727272727272727 = 0.227273 as shown in Table 5

E. Evaluation of Result

MaSO uses the above built ID and calculates the actual slippage in schedule due to already mentioned risk factors. From Figure 2 it is obtained that 'Schedule Slippage'- a utility node (represents the desirability of different event combinations involved in the network.) is driven by 'Schedule Overrun', as discussed in section 3. For instance, if probability of occurrence of risk factors is as shown in Figure 4 then value of 'Schedule Overrun' for each possible outcome as obtained from its CPT populated in section 3.4 and are shown in Table 5 is

Probability of occurrence	Values
Frequent	0.193182
Probable	0.204545
Occasional	0.227273
Improbable	0.227273
Remote	0.147727

TABLE 1 DATABASE MAINTAINING PROBABILITY OF OCCURRENCE OF RISK FACTORS

Schedule Overrun	Requirement Instability	Unnecessary Features	Creeping User Req.
Frequent	Remote	Occasional	Remote
Probable	Remote	Remote	Probable
Occasional	Improbable	Improbable	Occasional
Remote	Improbable	Improbable	Improbable
Improbable	Remote	Remote	Remote
Remote	Occasional	Probable	Probable
Improbable	Probable	Frequent	Remote
Occasional	Probable	Probable	Probable

From Figure 2 it is obtained that 'Schedule Slippage' is driven by 'Schedule Overrun', as discussed in section 3. There is a utility table associated with this node which holds the schedule slippage in months for each possible outcome of 'Schedule Overrun.

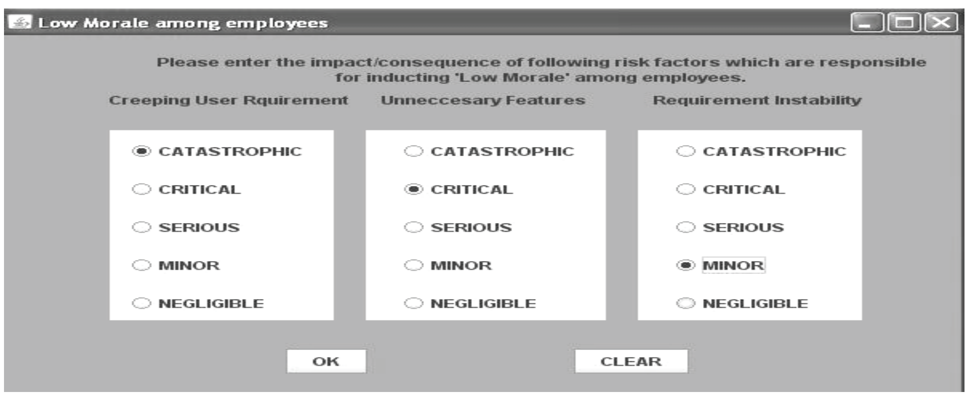

Figure 3. Interface to input impact of risk factors on Delay in Schedule

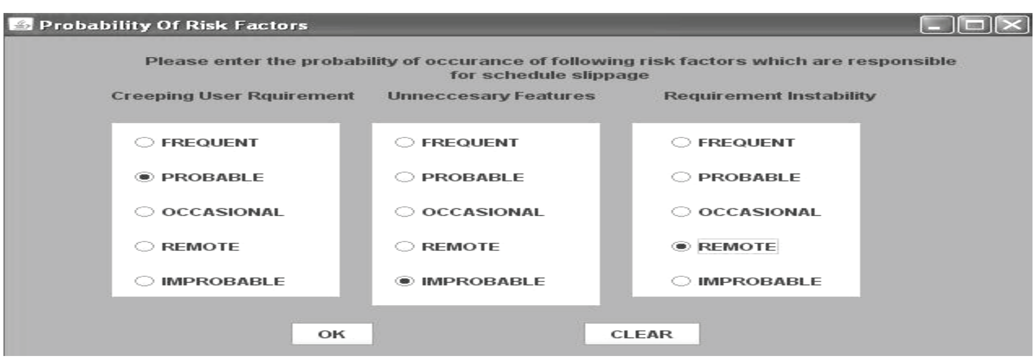

Figure 4. Interface to set the evidences/probabilities for risk factors

TABLE 2 CPT for Schedule Overrun depending on Creeping User Requirement

Creeping Requirement	Frequent	Probable	Occasional	Remote	Improbable
Frequent	0.166667	0.125	0.166667	0.25	0.285714
Probable	0.166667	0.25	0.166667	0.125	0.142857
Occasional	0.166667	**0.25**	0.333333	0.125	0.142857
Remote	0.333333	0.25	0.166667	0.125	0.285714
Improbable	0.166667	0.125	0.166667	0.375	0.142857

TABLE 3 CPT for Schedule Overrun depending on Unnecessary Features

Unnecessary Features	Frequent	Probable	Occasional	Remote	Improbable
Frequent	0.166667	0.142857	0.285714	0.142857	0.25
Probable	0.166667	0.142857	0.142857	0.285714	0.125
Occasional	0.166667	0.285714	0.142857	0.142857	**0.25**
Remote	0.166667	0.285714	0.285714	0.142857	0.25
Improbable	0.333333	0.142857	0.142857	0.285714	0.125

TABLE 4 CPT Schedule Overrun depending on Requirement Instability

Requirement Instability	Frequent	Probable	Occasional	Remote	Improbable
Frequent	0.166667	0.142857	0.285714	0.25	0.142857
Probable	0.166667	0.142857	0.142857	0.25	0.142857
Occasional	0.166667	0.285714	0.142857	***0.125***	0.285714
Remote	0.333333	0.142857	0.285714	0.125	0.285714
Improbable	0.166667	0.285714	0.142857	0.25	0.142857

TABLE 5 Partial CPT for Schedule Overrun

Creeping User Requirements	Probable									
Unnecessary Features	Remote					Improbable				
Requirement Instability	Frequent	Probable	Occasional	Remote	Improbable	Frequent	Probable	Occasional	Remote	Improbable
Frequent	0.139069	0.13474	0.160714	0.154221	0.13474	0.17803	0.173701	0.199675	0.193182	0.173701
Probable	0.247835	0.243506	0.243506	0.262987	0.243506	0.189394	0.185065	0.185065	0.204545	0.185065
Occasional	0.195887	0.217532	0.191558	0.188312	0.217532	0.234848	0.256494	0.230519	**0.227273**	0.256494
Remote	0.22619	0.191558	0.217532	0.188312	0.217532	0.265152	0.230519	0.256494	0.227273	0.256494
Improbable	0.191017	0.212662	0.186688	0.206169	0.186688	0.132576	0.154221	0.128247	0.147727	0.128247

TABLE 6 CALCULATION OF SINGLE CONDITIONAL PROBABILITY FOR SCHEDULE OVERRUN WITH OUTCOME TO BE OCCASIONAL AND IMPACTS AND PROBABILITIES AS ENTERED IN FIGURE 3 AND FIGURE 4

Risk Factor	Probability	Outcome (Value)	Normalized Weights	Contribution of Risk Factor (Value* Impact)
Creeping Requirement	Probable	0.25 (Table. 2)	5/11	0.11363636363636363636363636363636
Unnecessary Features	Improbable	0.25 (Table. 3)	4/11	0.0909090909090909090909090909090901
Requirement Instability	Remote	0.125(Table. 4)	2/11	0.0227272727272727272727272727272727

TABLE 7 UTILITY TABLE FOR SCHEDULE SLIPPAGE

Schedule Overrun	Frequent	Probable	Occasional	Remote	Improbable
Value	12	9	6	3	1

This table could be populated with values obtained by expert interviews. For instance if the utility table is a s shown in Table 7 then it means if schedule overrun happens to be ‘Frequent’ then ‘Schedule Slippage' will be by 12 months. For the case discussed above, the slippage in schedule or delay in project schedule will be 0.193182*12+0.204545*9+0.227273*6+0.227273*3+0.147727*1 = 6.35227 months (shown in Figure 6).

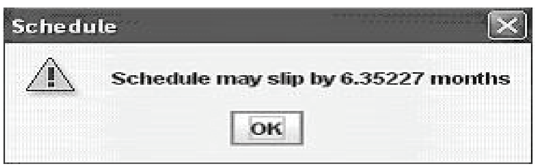

Figure 6. Schedule Slippage in months

IV. CONCLUSION

This paper outlines the system which is helpful to project managers for estimating the slippage in schedule that might occur due to various risk factors identified above, On the basis of impacts and the probability of occurrences of risk factors involved, risk managers will now be able to estimate the delay that could occur in the delivery of a project.

Although the response for a live schedule overrun management is encouraging but depending on the organization, some more factors that may have an impact on the schedule of a software development project. As a result, this work is scalable.

REFERENCES

[1] H. Barki, S. Rivard, J. Talbot, “Towards an assessment of risk”, *Journal of Management Information System*, vol. 10, pp. 203 – 225. 1993.

[2] John McManus ,*Risk Management in Software Development Projects*, MA:Butterworth-Heinemann, 2004, pp. 3-30.

[3] J. Jiang, G. Klein, “Software development risks to project effectiveness”, *Journal of Systems and Software*, vol. 52, pp. 3-10, May 2000.

[4] B.Boehma, C. Abts, S. Chulani “Software development cost estimation approaches – A survey”, *Annals of Software Engineering*, Vol. 10, pp. 177–205, Oct. 2004.

[5] B.Ramesh, M,Jarke, “Toward Reference Models for Requirements Traceability”,IEEE *Transactions on Software Engineering*, pp. 58-93, Jan 2001.

[6] J.S. Osmundsona, J.B. Michaelb, Martin ,J. Machniakc, M.A. Grossmand, “Quality management metrics for software development”, *Information and Management* ,vol 40, pp. 799 – 812, Sept. 2003.

[7] K. B. Uffe, M.L. Anders, Bayesian Networks and Influence Diagrams: A Guide to Construction and Analysis, Springer, Germany.

[8] Dale Walter Karolak, *Software Engineering Risk Management*, Wiley-IEEE Computer Society, 1995.

[9] K S Park and S H Kim, “Artificial intelligence approaches to determination of CNC machining parameters in manufacturing: a review” *Artificial Intelligence in Engineering*, vol. 12, pp. 127-134, January-April 1998.

[10] http://genie.sis.pitt.edu Accessed on 27th Feb., 2008.

[11] D.R. Jeffery and G. Low, “Calibrating estimation tools for software development”, *Software Engineering Journal*, IEEE, vol. 5, pp. 215-221, July 1990.

[12] Vijay Kumar Mago, M.Syamala Devi, “Decision making system: Sgent diagnosing child care diseases”, *Lecture Notes in Computer Science: Multi-Agent Systems and Applications V,Ivol.4696/207,* Springer Berlin / Heidelberg,2007, pp. 316-318.

[13] Kawal Jeet, Vijay Kumar Mago, Bhanu Prasad, Rajinder Singh Minhas, “*MaST: A Tool for Aiding the Staff Training Management by Using Influence Diagrams*”, SETP, 2008, pp. 156-164.

[14] Dan X. Houstan, Gerald T. Mackulak, James S. Collofello, “Stoichastic simulation of risk factor potential effects for software development risk management”, Journal of Systems and Software, vol. 59, pp. 247-257, Dec. 2001.

[15] T.K. Abdel-Hamid, “Investigating the cost/schedule trade-off in software development”, Software, IEEE, vol. 7, pp. 97-105,Jan. 1990.

[16] T.R. Browning, S.D. Eppinger, “Modeling impacts of process architecture on cost and schedule risk in product development”, IEEE Transactions on Engineering Management, pp. 428- 442,2002.

[17] M. Van Genuchten , “Why is software late? An empirical study of reasons for delay insoftware development”, *IEEE Transactions on Software Engineering*, vol. 17,pp. 582-590, June 1991.

[18] John S. Reel, “Critical Success Factors In Software Projects”, IEEE Software, vol. 18, pp. 18-23, May/Jun 1999.

[19] www.sun.java.com Accessed on 27th Feb, 2008.

[20] Balaram Das, “Generating Conditional Probabilities for Bayesian Networks: Easing the Knowledge Acquisition Problem”,Nov. 2004.

[21] Gregory f. Cooper, “A Bayesian Method for the Induction of Probabilistic Networks from Data”, Machine Learning, vol.9, pp. 309-347, Oct. 1999.

[22] Dimitri P. Bertsekas, John N. Tsitsiklis, “Introduction to Probability”, Athena Scientific, June 2002