To appear in Proc. IEEE 10th Int. Conf. Engineering of Complex Computer Systems (ICECCS 2005).

Cost-Benefit Trade-Off Analysis using BBN for Aspect-Oriented Risk-Driven Development

Siv Hilde Houmb
Department of Computer Science
Norwegian University of Science and Technology
Sem Selands Vei 7-9, NO-7034 Trondheim, Norway
sivhoumb@idi.ntnu.no

Geri Georg, Robert France, and James Bieman
Software Assurance Laboratory
Department of Computer Science, Colorado State University
601 S. Howes St., Fort Collins, CO 80523-1873
(georg/france/bieman)@CS.colostate.edu

Jan Jürjens
Systems Engineering, TU Munich
Boltzmannstr. 3, 85748 München/Garching, Germany
juerjens@in.tum.de

Abstract

Security-critical systems must perform at the required security level, make effective use of available resources, and meet end-users expectations. Balancing these needs, and at the same time fulfilling budget and time-to-market constraints, requires developers to design and evaluate alternative security treatment strategies. In this paper, we present a development framework that utilizes Bayesian Belief Networks (BBN) and Aspect-Oriented Modeling (AOM) for a cost-benefit trade-off analysis of treatment strategies. AOM allows developers to model pervasive security treatments separately from other system functionality. This eases the trade-off by making it possible to swap treatment strategies in and out when computing Return of Security Investments (RoSI). The trade-off analysis is implemented using BBN, and RoSI is computed by estimating a set of variables describing properties of a treatment strategy. RoSI for each treatment strategy is then used as input to choice of design.

Keywords:*Trade-off analysis, Bayesian Belief Network (BBN), Aspect-Oriented Modeling (AOM), and Risk-Driven Development (RDD).*

1 Introduction

In risk-driven development (RDD) security risks are identified, evaluated, and treated as an integrated part of development. The AORDD framework addresses the choice of security treatment strategies using a cost-benefit trade-off analysis computing Return of Security Investments (RoSI). RoSI is the value of loss reduction to money invested on security treatments. There are four ways to increase RoSI; minimize or eliminate losses, minimize investment, maximize positive returns, and accelerate the timing of returns.

The cost-benefit trade-off analysis is the main part of the AORDD framework and is implemented using BBN. The framework separates security concerns from core functionality using aspects. Each treatment strategy is modeled as an aspect model, and then composed with the primary model. Variables are used to estimate the properties of each treatment strategy, and are annotated in the composed model. Estimates are then fed into the BBN topology. The trade-off analysis provides decision-support for design choices, and follows a two step procedure; 1) evaluate security risks against the security risk acceptance criteria, and 2) trade-off designs by computing and comparing RoSI for each treatment strategy.

In the following we give a brief description of the AORDD framework and the basis of the BBN methodology.

We then present a part of the BBN topology followed by an example to demonstrate its use. The paper is organized as follows. Section 2 describes the AORDD framework, and Section 3 gives a brief introduction to the BBN methodology. In Section 4 we present the BBN topology and discuss how to manage security risks using the AORDD cost-benefit trade-off analysis. Section 5 gives a small example to demonstrate the approach, while Section 6 discusses future work.

2 AORDD Framework

The AORDD framework combines risk-driven development (RDD) [22] with aspect-oriented modeling (AOM) [9]. The framework consists of the AORDD process [11], an iterative development process, a security treatment aspect repository, an estimation repository, rules for how to annotate UML models with information used for estimation, rules for how to transfer information from the annotated UML models into the BBN topology, and a BBN-based cost-benefit trade-off analysis. Figure 1 gives an overview of the main components in the AORDD framework.

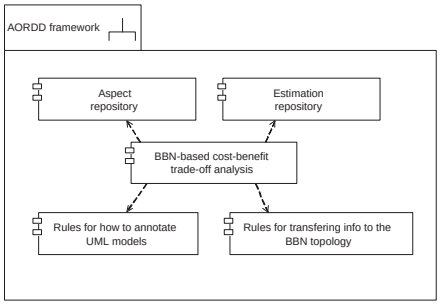

Figure 1. Overview of the components of the AORDD framework

Separation of concerns is important when making design trade-off decisions. We model each security treatment strategy as an aspect, which is added to the treatment aspect repository. We compose the aspect with the primary model, and perform functional and security verification [15]. Functional verification means that no functional requirement is affected by the treatment strategy. An example of security verification is provided in Section 6. This is done for all treatment strategies. We then chose an appropriate estimation set from the estimation repository. The estimation set depends on the variables used for trade-off analysis. In the example provided in Section 6, we describe treatment effect using the variables maintenance, cost, and security level. The estimation set is then applied on the composed model and given as input to the BBN topology.

3 The AORDD cost-benefit trade-off analysis

The trade-off analysis consists of two phases: 1) evaluate security risks against the security risk acceptance criteria, and 2) trade-off design alternatives by computing and comparing RoSI for each treatment strategy. Figure 2 gives an overview of the inputs and outputs of the two-phase trade-off analysis. The first phase takes a set of identified misuses of the system and their associated risk levels as input, and evaluates them against a set of security risk acceptance criteria. Misuses can be intentional system attacks or simple erroneous system usage. The associated risk levels indicate the damage that can occur to the system as a result of the misuse. Risks can vary from a degradation of supplied system services to economic loss due to assets being compromised. Risk levels are a combination of the impact of the misuse and its frequency. Security risk acceptance criteria partition these security risk levels into those risks that must be treated, and those risks that can be discarded from further consideration.

The result of the first phase of trade-off analysis is a list of misuses in need of treatment. An example of security risk acceptance criteria used to partition risk levels is that all risks with levels greater than or equal to security risk level “HIGH must be treated. In this context *treated* means reducing the risk level to lower than “HIGH. Such criteria should be provided either by system decision-makers, or through the business security policy or similar information sources. Note that in this example, all security risks lower than ”HIGH” are disregarded.

The input to the second phase of trade-off analysis is the list of misuses in need of treatment and their associated alternative security treatment strategies. The evaluation is based on different sets of priorities, standards, laws and regulations, and in particular business strategies and policies. RoSI for a particular treatment strategy is derived by evaluating the effect and the cost of each treatment strategy against the impact (loss or gain) and frequency of the misuse.

4 Bayesian Belief Networks (BBN)

BBN have proven to be an powerful technique for reasoning under uncertainty, and have been successfully applied when assessing the safety of systems [4], [5], [7], [20], and [8]. The BBN methodology is based on Bayes rule, and was introduced in the 1980s by Pearl [19] and Lauritzen and Spiegelhalter [17]. HUGIN [13] is the leading tool supporting BBN.

Bayes rule calculates conditional probabilities. Given the two variables X and Y, the probability P for the

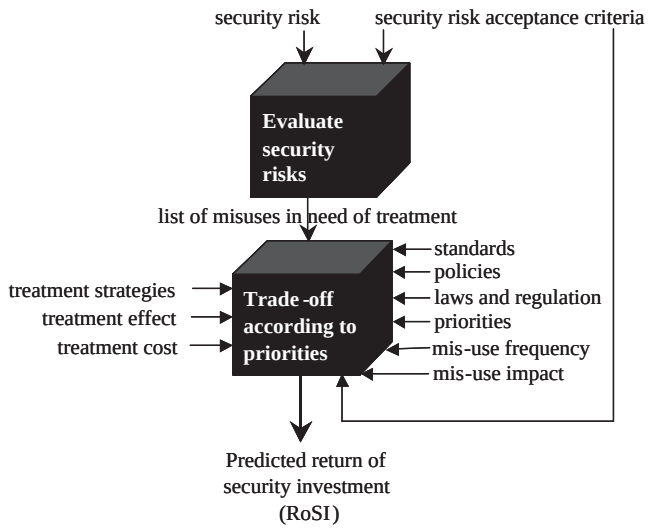

Figure 2. Overview of the trade-off procedure

variable X given the variable Y can be calculated from: P(X|Y)=P(Y|X)*P(X)/P(Y). By allowing X_i to be a complete set of mutually exclusive instances of X, Bayes formula can be extended to calculate the conditional probability of X_i given Y.

A BBN is a connected and directed graph consisting of a set of nodes, and a set of directed arcs (or links) describing the relations between the nodes. Nodes are defined as stochastic or decision variables, and multiple variables may be used to determine the state of a node. Each state of each node is expressed using probability density. The probability density expresses our confidence in the various outcomes of the set of variables connected to a node, and depends conditionally on the status of the parent nodes at the incoming edges. The nodes and associated variables can be classified into three groups:

- Target node(s) - the node(s) about which the objective of the network is to make an assessment. An example of such a node is "RoSI". In the example in Section 6 RoSI is defined as a decision variable with an associated utility function.
- Intermediate nodes - nodes for which we have limited information or beliefs. The associated variables are hidden variables. Typically hidden variables represent aspects that increase or decrease the belief in the target node, RoSI, such as "treatment level", "security level" etc.
- Observable nodes - nodes that can be directly observed or in other ways obtained. Examples of observable nodes for treatment level are nodes representing observable properties about the treatment and its environment; "treatment effect", "treatment cost" etc. In the example in Section 6 we use two stochastic variables to describe the treatment effect; security level and maintenance. Each of these variables has three associated states; low, medium, and high.

Application of the BBN method consist of three tasks:

- construction of BBN topology,
- elicitation of probabilities to nodes and edges, and
- making computations.

For further information on BBN, and in particular the application of BBN for software safety assessment see Gran [10].

5 The BBN topology for computing RoSI

Figure 3 depicts the top level BBN for phase 2 of the AORDD trade-off analysis. The node "RoSI is the target node of the network. The nodes priorities (PRI), budget (BU), business goals (BG), law and regulations (LR), security risk acceptance criteria (SAC), and policies (POL) are observable nodes, nodes that represent information and evidence that can be directly observed or in other ways obtained. The nodes acceptance level (AL) and security level (SL) are intermediate nodes and requires inputs from observable nodes. The node SL receives information and evidence from the three input nodes; static security level (CC EAL), dynamic security level (OS DL), and treatment level (TL). Each of these nodes are decomposed into BBN sub-nets and receive information and evidence from their respective sub-nets. Figure **??** shows the topology of the sub-net treatment level (TL). This node receives information from the stochastic varialbe nodes treatment effect (TE) and treatment cost (TC).

Recall that the target node gives the objective of the assessment; computing RoSI for each security treatment strategy. In BBN there are two sets of variables; stochastic and decision variables [14]. The stochastic variables represent the set of information on which a decision is based. We use nine stochastic variables to compute RoSI; treatment cost (TC), misuse cost (MC), confidentiality (Conf), integrity (Integr), availability (Avail), non-repudiation (NonR), accountability (Accnt), authenticity (Auth), and reliability (Relia). Variables can be in a set of states. In the current version of the BBN topology all variables have three associated states; low, medium, and high. Table 1 gives an overview of the variables and states of the top-level BBN.

The node PRI determines the priorities for the trade-off given as an order set of the variables for BU, BG, LR, SAC,

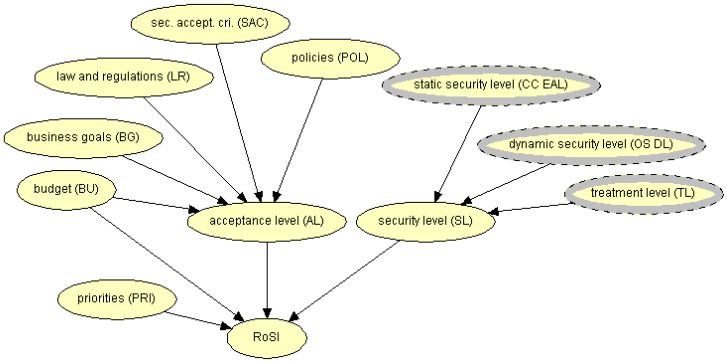

Figure 3. Top level BBN for phase 2 of the AORDD cost-benefit trade-off analysis

Node	Variables	States
RoSI	TC, MC, Conf, Integr, Avail, NonR, Accnt, Auth, and Relia	low, medium, and high
PRI	BU, BG, LR, SAC, and POL	low, medium, and high
BU	BUcostlimit	low, medium, and high
BG	BGcostlimit, Conf, Integr, Avail, NonR, Accnt, Auth, and Relia	low, medium, and high
LR, SAC, and POL	Conf, Integr, Avail, NonR, Accnt, Auth, and Relia	low, medium, and high
AL	ALcostlimit, Conf, Integr, Avail, NonR, Accnt, Auth, and Relia	low, medium, and high
SL	TC, MC, Conf, Integr, Avail, NonR, Accnt, Auth, and Relia	low, medium, and high

Table 1. Overview of variables and states of the top-level BBN

and POL. This set is then evaluated against the variables of the intermediate node security level (SL) as depict in Figure 3. The variables of SL are the same as for the RoSI target node.

Due to space restrictions we focus on the sub-net for the intermediate node TL. As shown in Figure 4, the sub-net targets treatment level and consists of the observable nodes effect of treatment strategy (TE) and cost of treatment strategy (TC). The node TE has a set of associated stochastic variables, while TC describes treatment cost. In the example we use two stochastic variables to describe treatment effect; treatment maintenance ($_M$) and treatment security level ($_S L$) (see Figure 8).

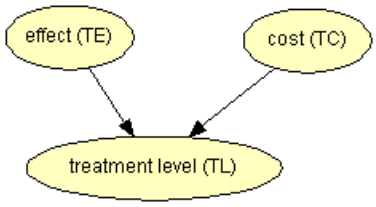

Figure 4. Sub-net for the intermediate node TL

6 Using the BBN topology to compute RoSI

The e-Commerce platform ACTIVE [1] was developed by the EU EP-27046-ACTIVE project. To access any of the services in ACTIVE users must either login as a registered user or a visitor. Logging into the system presents a security risk if the login actions are not properly protected. Different authentication mechanisms can be considered as cross-cutting aspects of the login sequence using AOM techniques. These mechanisms are modeled as aspects, and are considered separately from the main functionality of the ACTIVE platform. AOM allows the consideration of different authentication mechanisms to be added

(composed) with the system for trade-off analysis purposes. Figure 5 shows the original system login sequence, without any additional authentication mechanisms.

A user wishing to login to the e-commerce system uses a web browser on their local machine. The browser communicates with a web server over the internet. The web server has several related classes; only those associated with login and beginning a user session are shown in Figure 5. An account manager and associated database authenticate users. A profile manager and associated database keep track of personalized shopping information. A session manager creates a unique session identifier and keeps track of profile personalization that occurs during the session. The session manager is also responsible for session timeouts and storing information once the session is complete.

The IST EU-project CORAS [3] performed three risk assessments of ACTIVE inthe period 2000-2003. One of the misuses identified for the authentication mechanism was a man-in-the-middle attack [6]. During this kind of attack, user names and passwords can be intercepted by an attacker, and used later to impersonate a valid user.

The security attributes integrity and confidentiality are both compromised in this type of attack, so mechanisms that address integrity and confidentiality are potential security risk treatment strategies. We demonstrate the use of two such mechanisms, a variant of transport layer security (TLS) [15], and secure remote password (SRP) [21] to mitigate the risk. We model these two treatment strategies using aspect models in order to analyze their effect as input to cost-benefit trade-off analysis in AORDD. By using aspect models we can easily swap strategies in and out and feed results into the BBN topology.

An aspect model is composed with the primary model using composition rules before doing trade-off. Figure 6 shows the composed model of TLS with the ACTIVE platform login sequence.

Login still starts with the user's web browser requesting a login page from the e-commerce web server. The server responds with a login page. Now the TLS sequence is inserted; instead of the web browser sending a login message with a user name and password, an init message is sent, with a nonce (a non-repeating sequence value), the user's public key, and a self-signed certificate containing the user name and user's public key. The logic for the TLS handshake continues as described by Jürjens [15]. Since the TLS mechanism includes authentication, the classes for user account management and its associated database can be removed.

Figure 7 shows the composed model of the SRP aspect sequence with the e-commerce login sequence. The basic idea behind SRP is that prior to beginning the intial handshake, the client and server have received portions of a shared secret, usually based on a password. Our assumption is that this transaction takes place outside of the system. (This is similar to the assumption that acquiring secure certificates takes place outside the system in the TLS example.) The requirement is that a password verifier must have the client password, and from it generate a verification string, which is given to the server. The client and server must also agree upon a generator function and large prime number. Calculations of information and keys rely on these agreements. The client and server then exchange information based on the password and the verification string (respectively), and each generates a key independently. Upon completion of the handshake both client and server have keys that can be used to encrypt information that needs to be sent between them. The server has also authenticated the client since the verification string it has is associated with a particular client.

Similar to the TLS example, the SRP handshake is inserted into the login sequence after the initial request for a login page. Also similar to the TLS example, only the WebBrowser and WebServer classes are affected. And again, since the handshake includes authentication as well as generation of secret keys for encryption, the user account manager and database are not necessary. However, the SRP mechanism does require that the server have access to all the verification strings created for each user password. Thus, a verification string repository is needed for a large system like the e-commerce system. We have therefore eliminated the user account manager class from the e-commerce system, and changed the contents and name of its associated database to a user verification string database.

Security verification of the composed models is used as part of the input to the treatment level sub-net shown in Figure 4. To measure treatment effect TE, we use two stochastic variables; treatment maintenance ($_M$) and treatment security level ($_SL$). The probability distribution for the three security level states; low, medium, and high is determined by verification of the security treatment using an automated theorem prover (see Jürjens [15]). (We do not discuss maintenance metrics further in this paper, since the main aim is to demonstrate that feeding difference values into the BBN topology gives different outputs.)

We can establish that a security protocol such as the TLS variant here in fact satisfies its security requirements by making use of automated tool support which analyzes UML diagrams using automated theorem provers [16]. More specifically, we use the automated theorem prover e-SETHEO for verifying security protocols as a "black box": A TPTP input file is presented to the theorem prover and an output is observed. No internal properties of or information from e-SETHEO is used. This means that e-SETHEO can be used interchangingly with any other ATP accepting TPTP as an input format (such as SPASS, Vampire and Waldmeister) when it may seem fit.

With respect to the security verification, the results of the

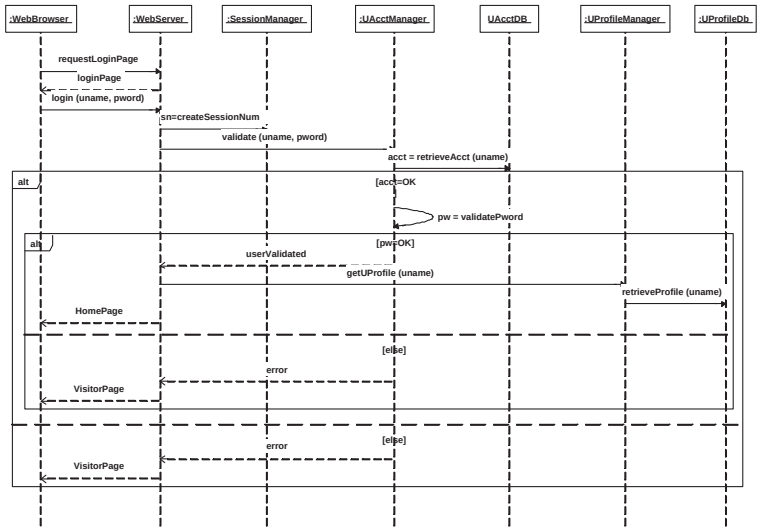

Figure 5. Login sequence

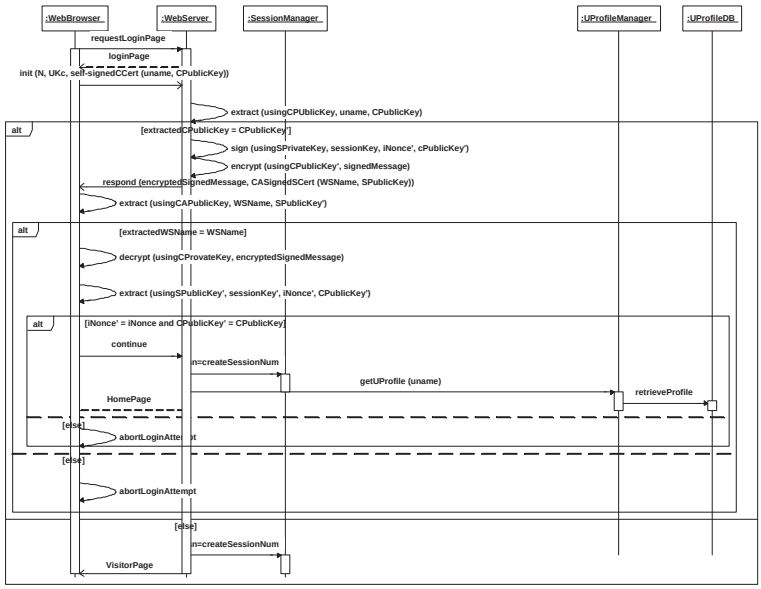

Figure 6. e-Commerce login sequence composed with TLS aspect

theorem prover have to be interpreted as follows: If the conjecture stating that an adversary may get to know the secret can be derived from the axioms which formalize the adversary model and the protocol specification, this means that there may be an attack against the protocol. We then use an attack generation machine programmed in Prolog to construct the attack. If the conjecture cannot be derived from the axioms, this constitutes a proof that the protocol is secure with respect to the security requirement formalized as the negation of the conjecture, because the logical derivation is sound and complete with respect to semantic validity for first-order logic. Note that since first-order logic in general is undecidable, it can happen that the ATP is not able to decide whether a given conjecture can be derived from a given set of axioms.

With respect to the TLS variant, e-SETHEO gives back the result that the conjecture $\mathsf{knows}(\mathsf{secret})$ cannot be derived from the axioms formalizing the protocol. Note that this result, which was delivered within 5 seconds, means that there actually exists no such derivation, not just that the

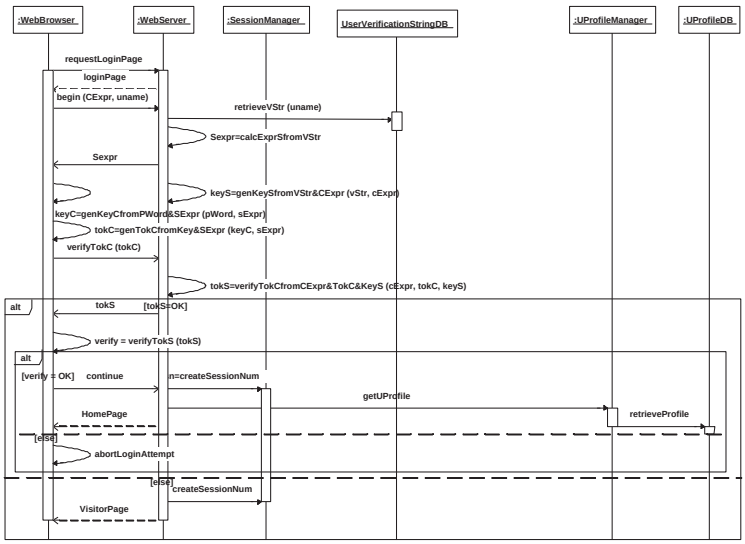

Figure 7. e-Commerce system login sequence composed with SRP aspect

theorem prover is not able to find it. This means in particular that an attacker cannot gain the secret knowledge anymore.

6.1 Cost-benefit trade-off analysis

For each treatment strategy we need to estimate treatment effect (TE) and cost (TC). This is done using a selection of estimation sets from the estimation repository in the AORDD framework. The set used depends on the type of system and its current development phase.

Recall from Section 4 that the BBN methodology consist of construction of the BBN topology, elicitation of probabilities to nodes and edges, and making computations. In section 5 we described the BBN topology, which is a general topology for computing RoSI in the cost-benefit trade-off analysis. The elicitation of probabilities and computations is, however, target of evaluation-specific and needs to be assigned in each assessment. Probability distribution functions (pdf) may be continuous functions or discrete values. In this example we use discrete values since this makes it conceptually easier for experts to assess, as well as making the computations much simpler.

6.2 Elicitation of probabilities

As an example of elicitation of probabilities we use the TL sub-net directly connected to the target node RoSI as depicted in Figure 8. We use three discrete stochastic variables to describe each treatment strategy; maintenance ($_M$), security level ($_S L$), and cost ($_C$). Since we are only evaluating two treatment strategies we include variables for both mechanism in the same network. In Figure 8 we have six stochastic variables; SRP_M, SRP_C, and $SRP_S L$ representing maintenance, cost, and security level for SRP and TLS_M, TLS_C, and $TLS_S L$ representing maintenance, cost, and security level for TLS.

To perform trade-off analysis we also need decision variables. Figure 8 includes three decision variables, the $ROSI_S RP$, $ROSI_T LS$, and RoSI. The decision variables are shown as rectangles. Their values are calculated using the observed states of the stochastic variables. The stochastic variables are shown as ovals. The diamonds in Figure 8 are utilities, and describe the interrelationships between the stochastic variables (in the cases of U2 and U3), or the interrelationships between the decision variables (in the case of U1). Utilities describe the resulting value of a decision variable given any combination of state values for the variables above it. Thus, the utility U2 specifies the value $ROSI_S RP$ should have, given any combination of states of the variables SRP_M, SRP_C, and $SRP_S L$. Similarly, the utility U1 specifies the value of RoSI given any combination of values of the intermediate decision variables $ROSI_S RP$ and $ROSI_T LS$. In our example, the utilities are simple lookup tables, but they can be defined using more sophisticated decision logic if desired (e.g. if one variable is to be given more weight than another). Figure 8 shows each of the stochastic variables, and the preliminary probability distributions for their associated states. These probability distribution functions are called prior distributions.

During elicitation of probabilities we feed the prior distributions into the BBN topology. In our example we as-

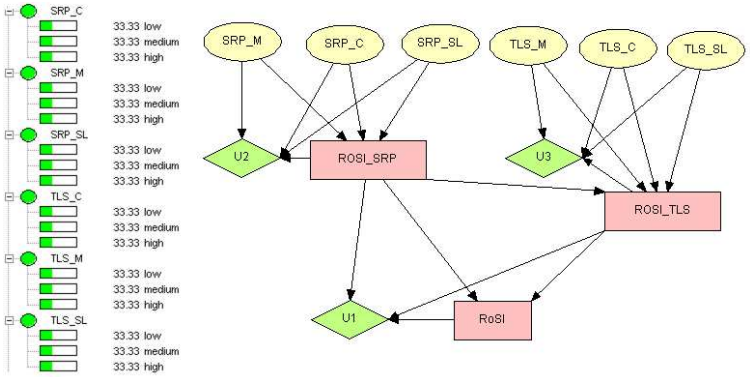

Figure 8. Variables and states for the example TL sub net

sume that expert judgment is collected and aggregated with empirical data. Several authors have discussed aggregation and expert judgment collection strategies [18], [2], and [12]. For simplicity we set the prior distribution for all observable variables to 0.33 (meaning that all states of all variables are assigned the same prior distribution and have the same influence on the outcome). This gives for all variables, $\{P(X = low) = 0.33\}$, $\{P(X = medium) = 0.33\}$, and $\{P(X = high) = 0.33\}$. The function $\{P(Y|X)\}$ (see Section 4) expresses the belief one has in, for example, the maintainability level of SRP if one knew the cost of SRP (the variable SRP_C). This information is expressed in a dependence matrix as given in Table 2.

6.3 Computation with the BBNs

The BBN computation first inserts observations in the observable nodes, and then uses the rules for probability calculation backward and forward along the edges, from the observable nodes, through the intermediate nodes to the target node. Forward calculation is straight forward, while backward computation is more complicated. Backward calculation is solved using Bayes methodology (see Jensen [14] for details). Manual computation on large BBN topologies is not tractable, so we make use of the BBN tool HUGIN [13]. Note that the amount of information collected before a decision is made depends on the type of decision, the resources available, time frame, and budget (meaning that one should not spend more money on collecting information than the value of the decision).

Figure 9 shows the result of the computation after observations are given as input to the BBN topology. Actual states of each of the stochastic variables are shown in the figure, on the left side. So, for example, the SRP_C variable is in the high state, the SRP_M variable is in the medium state, and the SRP_SL variable is in the low state. Utilities U2 and U3 are used to determine the states of the decision variables $ROSI_SRP$ and $ROSI_TLS$. In this example, the combination of states of the SRP variables means that the $ROSI_SRP$ decision variable is twice as likely to be in the medium state as the high state, and it is one and one-half times as likely to be in the medium state as the low state. The $ROSI_TLS$ decision variable is twice as likely to be in the high state as the medium state, and one and one-half times as likely to be in the low state as the medium state. Utility U1 takes these distributions and calculates the state of the RoSI decision variable. The result shows that the TLS treatment is one and one-half times more effective than the SRP treatment.

Figure 10 shows how the BBN computation changes when the observations entered for the stochastic variables are changed. As in Figure 9, the values for the states of each of the stochastic variables are shown in the figure on the left side. The same utilities are used to calculate the states of the decision variables. In this figure, the states of SRP_M, SRP_SL, and TLS_SL have been changed from the values given in Figure 9. The result is that the decision variable $ROSI_SRP$ is one and one-half times as likely to be in the high state as either the medium or low states. The $ROSI_TLS$ variable is one and one-half times as likely to be in the low state as the medium state and a half time as likely to be in the low state as the high state. As for the previous example the U1 utility is used to compute the state of the decision variable RoSI. In this case the result shows that the SRP treatment is a half time more effective than the TLS treatment.

7 Conclusion and further work

This paper has briefly described the AORDD framework and focused on the cost-benefit trade-off analysis of AORDD. The cost-benefit trade-off analysis is implemented using BBN. The BBN topology covers the security level

SRP_C $ROSI_S RP$	Low	Medium	High
Low	1.0	0.0	0.0
Medium	0.0	1.0	0.0
High	0.0	0.0	1.0

Table 2. Dependency matrix on the belief one has in the maintainability level of SRP given the cost of SRP

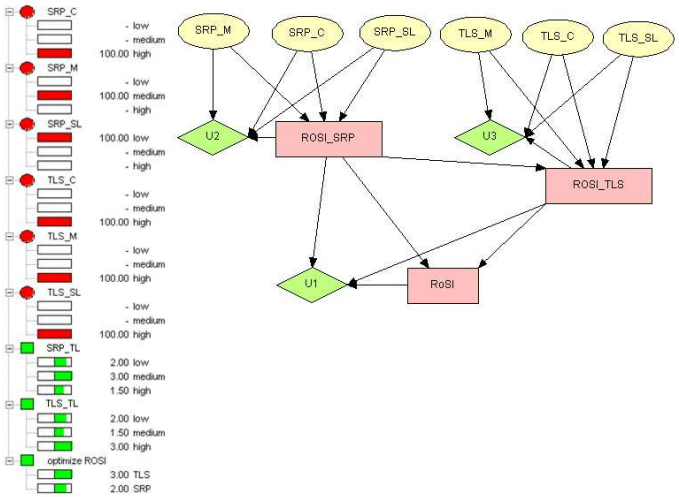

Figure 9. Example of observations in favor of TLS

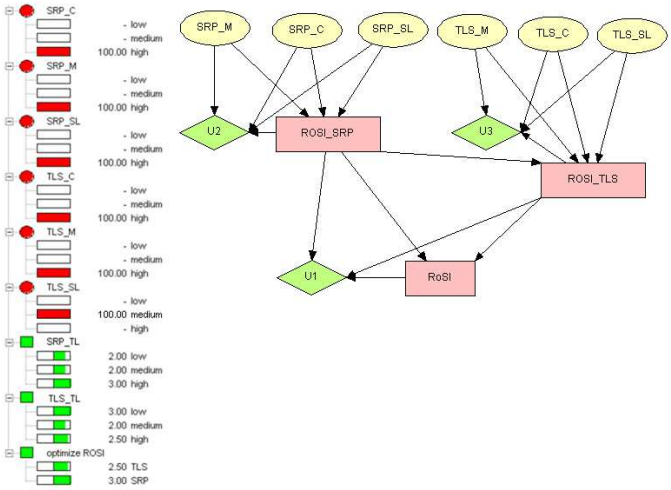

Figure 10. Example of observations in favor of SRP

of a system described as the combination of its static security level, its dynamic security level (covering operation security), and the treatment level of a specific security treatment. The security level is evaluated against an acceptance level comprised of the budget, security acceptance criteria, law and regulations, business goals, and policies. The main

goal of the trade-off analysis is to compute RoSI of each treatment strategy to be used as input to design decisions.

The BBN methodology consists of three steps; (1) construction of the BBN topology, (2) elicitation of probabilities to nodes and edges, and (3) making computations. Elicitation of probabilities is done using available empirical or observable information sources combined with subjective expert judgment, while computations are done using the algorithm provided by HUGIN for conditional probabilities. However, to demonstrate the approach we used two sets of fictive observations for the effect and cost variables of two different treatment strategies. This is done in order to demonstrate how different observable variable sets influence the decision calculation. During development of systems, these values are obtained from experience within the company, and general experience factories, in addition to using the stakeholders and participants in the development project as experts.

The result of the cost-benefit trade-off analysis is highly dependent on the observation and evidence entered, as well as the variables and the relation between them. This means that different sets of measures will give different results. We have not addressed estimation of variables in this paper due to space restrictions, but will address this issue in further work. Estimation sets are domain-, abstraction level-, viewpoint-, and development phase-specific.

References

[1] EP-27046-ACTIVE, Final Prototype and User Manual, D4.2.2, Ver. 2.0, 2001-02-22., 2001.

[2] R. M. Cooke. *Experts in Uncertainty: Opinion and Subjective Probability in Science*. Oxford University Press, 1991.

[3] CORAS (2000–2003). A platform for risk analysis of security critical systems. IST-2000-25031, http://www.sourceforge.net/coras/, 29. November 2004.

[4] P.-J. Courtois, N. E. Fenton, B. Littlewood, M. Neil, L. Strigini, and D. R. Wright. Bayesian belief network model for the safety assessment of nuclear computer-based systems. Second year report part 2, Esprit Long Term Research Project 20072-DeVa, 1998.

[5] K. Delic, M. Mazzanti, and L. Stringini. Formilizing engineering judgment on software dependability via belied netowrks. In *DCCA-6, Sixth IFIP International Working Conference on Dependable Computing for Critical Applications, "Can We Rely on Computers?"*, Garmisch-Partenkirchen, Germany, 1997.

[6] T. Dimitrakos, B. Ritchie, D. Raptis, J. O. Aagedal, F. den Braber, K. Stølen, and S. Houmb. Intgegrating model-based security risk management into ebusiness systems development: The coras approach. In J. Monteiro, P. Swatman, and L. Tavares, editors, *Second IFIP Conference on E-Commerce, E-Business, E-Government (I3E 2002)*, volume 233 of *IFIP Conference Proceedings*, pages 159–175. Kluwer, 2002.

[7] N. Fenton, B. Littlewood, M. Neil, L. Strigini, A. Sutcliffe, and D. Wright. Assessing dependability of safety critical systems using diverse evidence. *IEEE Proceedings Software Engineering*, 145(1), 1998.

[8] N. Fenton and M. Neil. A critique of software defect prediction models. *IEEE Transaction of Software Engineering*, 25(5):675–689, 1999.

[9] G. Georg, R. France, and I. Ray. An aspect-based approach to modeling security concerns. In *Workshop on Critical Systems Development with UML (CSDUML'02)*. Dresden, Germany, October 2002.

[10] B. A. Gran. *The use of Bayesian Belief Networks for combining dispparate sources of information in the safety assessment of software based systems*. Doxtoral of engineering thesis 2002:35, Department of Mathematical Science, Norwegian University if Science and Technology, 2002. 2002:35.

[11] S. H. Houmb, G. Georg, R. France, and D. Matheson. Using aspects to manage security risks in risk-driven development. In *3rd International Workshop on Critical Systems Development with UML*, number TUM-I0415, pages 71–84. TUM, 2004.

[12] S. H. Houmb, O. A. Johnsen, and T. Stalhane. Combining Disparate Information Sources when Quantifying Security Risks. In *Proceeding of SCI 2004, RMCI 2004, Orlando, July 2004*, 2004.

[13] HUGIN: Tool made by Hugin Expert a/s, Alborg, Denmark, 2004. http://www.hugin.dk.

[14] F. Jensen. *An introduction to Bayesian Network*. UCL Press, University College London, 1996.

[15] J. Jürjens. *Secure Systems Development with UML*. Springer-Verlag, Berlin Heidelberg New York, 2004.

[16] J. Jürjens and P. Shabalin. Tools for Critical Systems Development with UML. In N. Jardin Nunes, B. Selic, A. Silva, and A. Toval, editors, *UML Modeling Languages and Applications. UML 2004 Satellite Activities, Lisbon, Portugal, October 11–15, 2004, Revised Selected Papers*, volume 3297 of *LNCS*. Springer, 2004.

[17] S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems (with discussion). *Journal of the Royal Statistical Society*, Series B 50(2):157–224, 1988.

[18] K. Øien and P. R. Hokstad. Handbook for performing expert judgment. Technical report, SINTEF, 1998.

[19] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Network forPlausible Inference*. Morgan Kaufmann, 1988.

[20] SERENE: Safety and Risk Evaluation using Bayesian Nets. ESPIRIT Framework IV nr. 22187, 1999. http://www.hugin.dk/serene/.

[21] Thomas Wu, "The SRP authentication and key exchange system", RFC 2945, Network Working Group, 2000.

[22] K. Stølen, F. den Braber, T. Dimitrakos, R. Fredriksen, B. A. Gran, S. H. Houmb, Y. C. Stamatiou, and J. Ø. Aagedal. Model-based risk assessment in a component-based software engineering process: The CORAS approach to identify security risks. In F. Barbier, editor, *Business Component-Based Software Engineering*, pages 189–207. Kluwer, 2002. ISBN: 1-4020-7207-4.