Probabilistic Risk Assessment for Security Requirements: A Preliminary Study

Seok-Won Lee
Department of Computer Science and Engineering
University of Nebraska, Lincoln, USA
slee@cse.unl.edu

Abstract — **Risk assessment is a critical decision making process during the Security Certification and Accreditation (C&A) process. However, existing infrastructure-wide C&A processes in real world are challenged by the ever increasing complexity of information systems and their diverse socio-technical operational environments. The lack of an explicit model and the associated uncertainties of software behavior are two main reasons that directly impact the effectiveness of risk assessment as well as the subjective decisions made based on the different level of domain expertise. In this paper, we propose a method for a probabilistic model-driven risk assessment on security requirements. The security requirements and their causal relationships are represented using MEBN (Multi-Entities Bayesian Networks) logic that constructs an explicit formal risk assessment model that supports evidence-driven arguments. The proposed approach is described by using real-world C&A scenarios to show not only its feasibility for security requirements risk assessment but also its effectiveness for the sensitivity analysis to identify critical influences among information entities in a complex and uncertain operational environment.**

Keywords – Probabilistic risk assessment, Security requirements, Bayesian network, Certification, Accreditation, Sensitivity analysis

I. INTRODUCTION

The government, defense, and private sectors spend billions of dollars every year in securing software systems that support their critical businesses/missions. According to a recent survey, among 1,300 global companies, and government and non-profit agencies in 55 nations, the compliance with regulations has taken the lead as the primary driver of security efforts in an organization, surpassing worms and viruses. However, various reports [8] [11] [31] indicate that the process of measuring compliance with security Certification & Accreditation (C&A) requirements is often irregular and unreliable. As a result, C&A processes lack consistent, complete and measurable outcomes and fail to provide adequate and timely information to understand security risks and make informed decisions.

Due to the ever increasing complexity of information systems and their diverse socio-technical operational environments, the associated risk assessment processes need to handle various types of uncertainties of the information. These uncertainties are inherent to the specific problem domain which is being modeled with many underlying assumptions based on a prior knowledge from the previously known phenomena. In order to achieve probabilistic risk assessment for effective decision making, the nature of uncertainty in risk assessment must be carefully identified, expressed, propagated, synthesized and understood. From the risk assessment perspective, uncertainties are originated from various sources:

- *Consistency and completeness of the information and its validity*: We often use different types of historic data, questionnaires results, etc. to support our claim, however, the coverage and effectiveness of the sources of the information are yet to be perfect to the given situation.
- *Assumptions behind the model*: Risk assessment process is based on the models of the relationships among different risk factors with assumptions that are very unique and specific to the situation which are hard to be treated in general.
- *Dynamics of the environment*: The constantly changing systems' operational environments are different to one another and the risks of the target system are always expected to have new factors to be considered based on the changes being made (or those that will be made).
- *Subjective nature of human expertise and judgment*: The level of human expertise on a certain subject, often becomes the main resource for risk assessment, for each individual is different and lack the ability to effectively communicate to converge into the final decision making points.

Risk assessment is a process of identifying relevant information resources (risk factors), discovering their relationships, and integrating them to form a risk assessment argument. In regards to the uncertainties, we must address four issues: 1) how the uncertainty of information should be understood and expressed (for problem understanding); 2) how different pieces of information are integrated (to build a causal model); 3) how new information can be incorporated (for model evolution); and 4) what if the operational situation is changed (for the changing risk factors and their adaptation).

During the process of risk assessment, experts can make not only judgments, but also meta-judgments [13], that is, judgments about the degree of certainty that they have in their judgments. Since the judgments can be precisely modeled through an appropriate probability distribution, we, therefore, use the probability theory to express uncertainties in the domain of risk assessment. In addition, we notice that the

978-0-7695-4453-3/11 $25.00 © 2011 IEEE
DOI 10.1109/SSIRI.2011.12

experts can do better in making judgments at a lower level, or at a more observable level (e.g. the configuration correctness of a certain countermeasure) but less effective at a higher level (e.g. the level of risk of the entire system). This is because, at a higher level, making a decision requires a more complex process to integrate or synthesize different information coming from lower levels, especially when there are hundreds of or even thousands of risk factors involved in the inspecting systems (or network of systems). Therefore, we use the Bayesian Belief Network (BBN) that provides a theoretical foundation to incorporate such accumulated evidences. Also, we propose a way to make our probabilistic risk assessment model parametric, so that the risk assessment model can be parameterized or instantiated automatically, instead of building a new model, whenever the operational situation is changed.

In this paper, as an application example, we focus on the Department of Defense Information Technology Security C&A Process (DITSCAP) that defines certification as a comprehensive evaluation of the technical and non-technical security features of an information system and other safeguards made in support of the accreditation process, to establish the extent to which a particular design and implementation meets a set of specified security requirements [2].

II. Related Work

Several risk assessment tools and approaches are proposed in order to determine threats/vulnerabilities in the early phase of software development life cycle. CORAS [5] and Risk-Management Framework [32] propose their own methodological steps, but lack specific guidelines to interoperate with C&A activities and appropriately utilize the evidences gathered for C&A requirements into the risk assessment process.

Several quantitative risk assessment methods exist. Butler [6] proposes a SAEM method which is a cost-benefit analysis process for analyzing security design decisions based on the comparison of a "threat index". However, it is based on some impractical assumptions. Ekelhart [9] uses security ontology to improve quantitative risk analysis and promote the common understanding of the involved risk factors. However, it is not sufficient to identify how these risk factors can contribute to the entire risk assessment process. Quantitative risk-based requirements reasoning in [12] uses PACT as a "filter" arranged in series to find out a proportion of likelihood or the impact of risk factor. However, it lacks the ability to represent the impacts among multiple risk factors. The SSRAM model in [28] provides a prioritization that aids in determining how the risks identified will be addressed in different phases of software development. However, it lacks a baseline for systematically identifying potential risks and reasoning about their relationships and interactions in a real operational environment.

In regards to the use of probability theory and modeling techniques in risk assessment, Littlewood uses Bayesian approach to assess the reliability of fault-tolerant software [26]. He also used Multi-legged arguments to increase the confidence in dependability arguments [27]. Gregoriades and Sutcliffe [19] developed a probabilistic model to reason about the system reliability that predicts human and machine reliabilities with given input variables representing the scenario and ranges of environmental conditions. Fenton and Maiden [15] and Neil et al. [30] have developed large BBN models to assess risk at the system level, such as the reliability of system engineering processes for developing ships, vehicles, or the operational reliability of air traffic control systems. Fenton [14] indicated that BBN is the most effective model in software quality management through comparison with six other types of methods. Hui, et al. [22] also introduced a method to build a BBN for software risk assessment. But these BBN approaches are limited since the number of nodes and structures of their models are fixed and therefore, can only be used in a certain specific situation. When the given situation changes, the model needs to be modified manually to fit the new situation, even these changes are simply to add some repeated nodes or substructures. This is partly because standard BBN lacks the sufficient expressive power.

To address this limitation, a number of languages have been developed that represent probabilistic knowledge as modular units with repeated substructures that can be composed into complex domain models [24]. These include pattern theory [20], hidden Markov models [10], the plates language implemented in BUGS [18], Object-Oriented Bayesian networks (OOBN) [23], probabilistic relational models (PRMs) [17], and MEBN [24]. In contrast, MEBN provides a more flexible, coherent way to facilitate representation of knowledge at a natural level of granularity. It combines the expressive power of first-order logic with a sound and logically consistent treatment of BBN. MEBN fragments (MFrags) are parametric causal fragments and can be instantiated and combined to form arbitrarily complex graphical probability model. A feature of MEBN not present in PRMs, plates or OOBNs is the use of context constraints to specify logical conditions that determine whether one random variable influences another.

III. BackGround

A. Modeling C&A Requirements and Risk Components

To systematically identify and reason about the risk components expressed in natural language C&A security requirements descriptions, we extend the Common Criteria security model [1]. The resulting model in Figure 1, explains the relationships between security requirements and risk components.

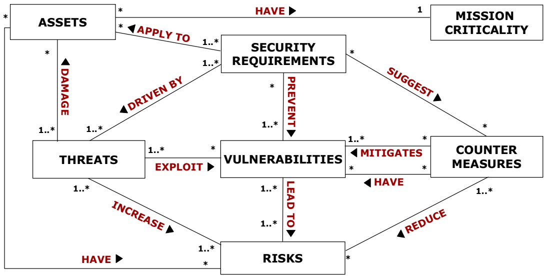
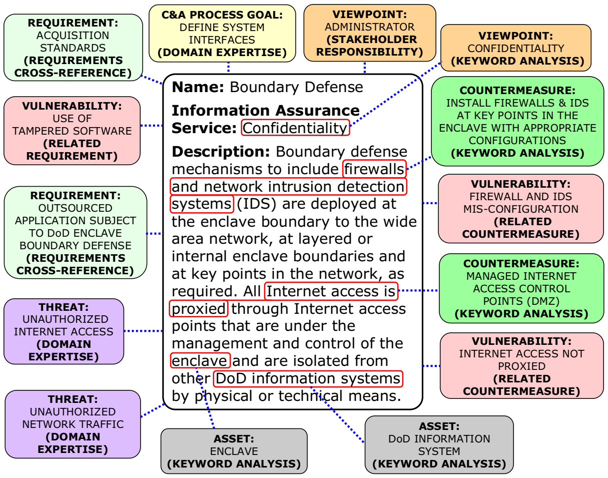


Figure 1. Requirements and Risk Model [16]

The risk components and their correlations in the Requirements and Risk Model (hereinafter called the Risk

Model) can be simply described as: the *assets* have *mission criticalities* and *risks*, *threats* can exploit the *vulnerabilities* to damage the assets thus increase their risks. The security requirements driven by threats suggest *countermeasures* to mitigate the vulnerabilities thus prevent the vulnerabilities and reduce the risks, whilst countermeasures may also introduce vulnerabilities which can lead to risks. The final goal is to evaluate the risk of assets in a given situation.

Based on the model in Figure 1, domain experts identify the relevant risk components and map them into the concepts in the domain-specific taxonomies of threats, assets, vulnerabilities, and countermeasures modeled in the Problem Domain Ontology (PDO) [25]. For example, Figure 2 shows the explication of multi-dimensional domain concepts for the DITSCAP "Boundary Defense" requirement [3].



Name: Boundary Defense

Information Assurance Service: Confidentiality

Description: Boundary defense mechanisms to include firewalls and network intrusion detection systems (IDS) are deployed at the enclave boundary to the wide area network, at layered or internal enclave boundaries and at key points in the network, as required. All Internet access is proxied through Internet access points that are under the management and control of the enclave and are isolated from other DoD information systems by physical or technical means.

REQUIREMENT: ACQUISITION STANDARDS (REQUIREMENTS CROSS-REFERENCE)
C&A PROCESS GOAL: DEFINE SYSTEM INTERFACES (DOMAIN EXPERTISE)
VIEWPOINT: ADMINISTRATOR (STAKEHOLDER RESPONSIBILITY)
VIEWPOINT: CONFIDENTIALITY (KEYWORD ANALYSIS)
VULNERABILITY: USE OF TAMPERED SOFTWARE (RELATED REQUIREMENT)
COUNTERMEASURE: INSTALL FIREWALLS & IDS AT KEY POINTS IN THE ENCLAVE WITH APPROPRIATE CONFIGURATIONS (KEYWORD ANALYSIS)
VULNERABILITY: FIREWALL AND IDS MIS-CONFIGURATION (RELATED COUNTERMEASURE)
REQUIREMENT: OUTSOURCED APPLICATION SUBJECT TO DoD ENCLAVE BOUNDARY DEFENSE (REQUIREMENTS CROSS-REFERENCE)
COUNTERMEASURE: MANAGED INTERNET ACCESS CONTROL POINTS (DMZ) (KEYWORD ANALYSIS)
THREAT: UNAUTHORIZED INTERNET ACCESS (DOMAIN EXPERTISE)
VULNERABILITY: INTERNET ACCESS NOT PROXIED (RELATED COUNTERMEASURE)
THREAT: UNAUTHORIZED NETWORK TRAFFIC (DOMAIN EXPERTISE)
ASSET: ENCLAVE (KEYWORD ANALYSIS)
ASSET: DoD INFORMATION SYSTEM (KEYWORD ANALYSIS)

Figure 2. Analyzing a DITSCAP Requirement

IV. Probabilistic Risk Assessment for Security Requirements

In the Requirements and Risk Model shown in Figure 1, we can identify two types of links, the *Information Discovery Link*, and the *Causal Relationship Link*. The former has been used to discover and understand multi-dimensional correlations among C&A requirements [16] as shown in Figure 2. The latter, we believe, is helpful to extract the causal relationships and reason about their impacts within the process of risk assessment.

The domain of risk assessment is full of uncertainties and effective decisions must be made to answer questions, for example, "How likely a certain threat will occur?" or "How effective a countermeasure can mitigate certain vulnerability?" etc. As mentioned in section II, BBNs have been widely applied as a probabilistic reasoning technique in software engineering and other domains. BBNs allow the construction of probabilistic models involving large numbers of interrelated uncertain hypotheses and have the ability to combine qualitative expert knowledge with quantitative measures of plausibility and statistical data. In Figure 1, the risk is causally related to a certain set of security requirements, threats, vulnerabilities, countermeasures and other risk factors (here risk factors represent not only the risk components in the Risk Model, but also their properties related to risk assessment, e.g., the maintenance of countermeasure etc.), therefore, we propose to model the risk assessment by using Bayesian probability to perform the probabilistic risk assessment.

A. Motivations and Benefits

In a security C&A process, our probabilistic risk assessment is motivated by the following needs to: 1) Make the process of risk assessment more explicit and systematic to support better decisions under uncertainty; 2) Make risk assessment techniques be based on probabilistic metrics that can be evaluated systematically with little involvement of subjective measures from domain experts [34]; 3) Classify the relevant risk components and map their causal relationships to concepts in the domain-specific taxonomies of threats, assets, vulnerabilities, and countermeasures modeled in the PDO; 4) Use BBN to represent and model uncertainty among dependability requirements in C&A process then, analyze causal relationships and impacts qualitatively and quantitatively among different risk components; and 5) Automatically generate BBN according to the given risk assessment scenarios.

Also, our probabilistic risk assessment approach has the following benefits: 1) Different stakeholders can have an explicit and common understanding of the risk assessment process and the decision rationale; 2) Every phase of risk assessment process is arguable and can be re-evaluated based on the common understanding; and 3) Simulation is possible through sensitivity analysis to show how different risk components are related to and impacted by each other.

B. Bayesian Network for Risk Assessment

In risk assessment, BBN can provide stakeholders a causal relationship graph and inference capabilities among different risk components. Before building the BBN for risk assessment, it is worth giving a brief definition for the risk components discussed in our domain (adapted from [4]).

- *C&A requirement* is the specification that describes the security conditions and constraints under which the system must be operated.
- *Mission criticality* is related to asset, representing relative importance of the asset to the mission success.
- *Asset* is the resource of the system that needs to be protected to achieve its goal
- *Threat* is a potential danger to the system such as a person, system component or event that might result in a compromise of the secure operation of the system.
- *Vulnerability* is a weakness in the system or a point where a system is susceptible to attack. This weakness could be exploited to violate the system security.
- *Countermeasure* is an action, device, procedure or technique for protecting the system against threats to its secure operation.
- *Risk* is the asset-based risk, it presents the risk extent of an asset can impact on the whole system.

We extract the causal relationship and construct the BBN based on the Risk Model in Figure 1. Building a BBN to model the risk assessment process requires a thorough enumeration of all the relevant risk factors and knowing correctly which risk factor causally influences other factors. For a complex socio-technical system, it is very difficult to produce precise network structure and Node Probability Tables (NPTs) for hundreds of or even thousands of risk factors. In addition, different people have different experiences and understanding; the building process may be very subjective. In order to make our BBN reliable and justifiable, we adopt an objective and formal way to capture the process knowledge of experts' risk assessment.

Normally, there are two methods to build a BBN: 1) learning the network structure based on the given data set; and 2) constructing a network structure according to expert's experience [21]. In this paper, we adopt the second approach due to two reasons: 1) Constructing network structure by learning requires a lot of sample data and such data is hard to be collected effectively and comprehensively; and 2) We believe that the experts of many years' experience can judge the dependent/independent and direct/indirect relationships between risk factors more effectively. In addition, a formal way of building the BBN makes the process more explicit and arguable, and improve the communication and understanding among different experts while helping to reach to an agreement.

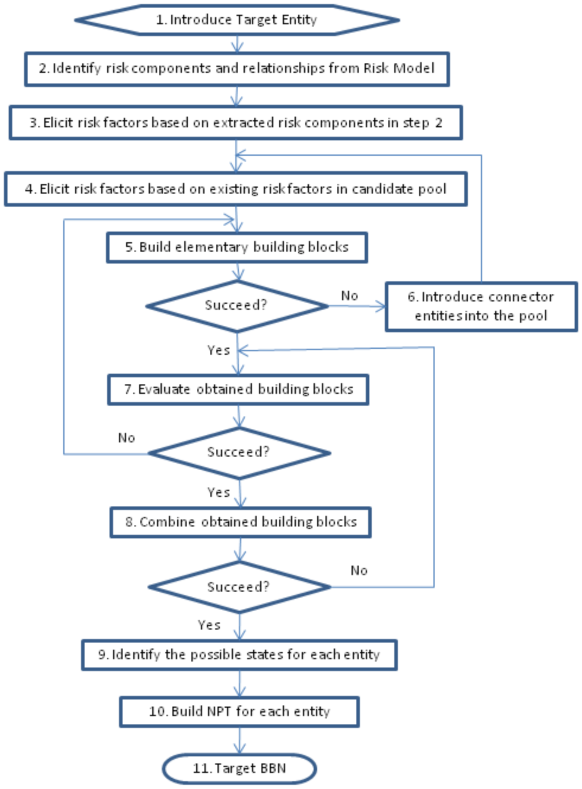

1. Introduce Target Entity
2. Identify risk components and relationships from Risk Model
3. Elicit risk factors based on extracted risk components in step 2
4. Elicit risk factors based on existing risk factors in candidate pool
5. Build elementary building blocks
Succeed?
No
6. Introduce connector entities into the pool
Yes
7. Evaluate obtained building blocks
No
Succeed?
Yes
8. Combine obtained building blocks
Succeed?
No
Yes
9. Identify the possible states for each entity
10. Build NPT for each entity
11. Target BBN

Figure 3. The process of developing causal model

For the causal relationships extraction, we use the method suggested by Neil [29], proposed "building blocks" concepts, using "idioms" to build causal dependency diagrams. Five idioms are summarized as follows:

- *Definitional/synthesis idiom*: This idiom can be used to represent definitional relationships such as X=Y/Z. Synthetic nodes can ease calculation or understanding, and create hierarchies of sub-attributes to define complex super attributes.
- *Cause-effect idiom*: This cause connection can be determined based on the following facts: Chronology of events in time; Events occur in sequence; Productive, physical or intentional relationship between cause and consequence; Rules for determining causality couched in common sense.
- *Measurement idiom*: This idiom is used to model the uncertainty of our own ability to observe accurately. For example, the uncertainty of the testing accuracy is impacted by actual defects and detected defects.
- *Induction idiom*: This idiom models the uncertainty related to inductive reasoning (including historical experiences) based on populations of similar or exchangeable members.
- *Reconciliation idiom*: This idiom reconciles independent sources of evidence about a single attribute of a single entity, where these sources of evidence have been produced by different methods. Also, it combines uncertain definition model with causal inference model and combines information from different causal models.

For the NPTs definition, we use the historic statistical data and the estimation of experts. For example, analysts can identify the possible types of threats and calculate or estimate their frequencies under the given environment. We firstly identify all the concerned risk factors and add them into a candidate pool, then identify their causal relationships according to the five idioms along with the Risk Model, and finally combine these causal fragments and build their NPTs to finish the construction of the BBN. Figure 3 depicts the process and the brief descriptions about each step are as follows:

Step 1. The target entity should be defined. The target node presents our main purpose of the BBN, namely, to assess "the risk of asset".

Step 2. Identify risk components and relationships from Risk Model which are relevant to the target entity. In our example, the identified entities are ASSETS, MISSION CRITICALITY, THREATS, VULNERABILITIES, COUNTERMEASURES and the relationships are INCREASE, EXPLOIT, MITIGATE, LEAD_TO, REDUCE, DAMAGE, HAVE. Note that these extracted entities or relationships may not be used in BBN directly. For example, we replace COUNTERMEASURE with COUNTERMEASURE_EFFECTIVENESS because we concern the effectiveness of countermeasures to mitigate the vulnerabilities, not the costs of countermeasures. There are two advantages by doing so. Firstly, it is for better understanding and communication, secondly, it tells analysts on what aspects (effectiveness or costs) of risk components they should concentrate in the following steps.

Step 3. Elicit risk factors based on identified risk components in step 2. We elicit all risk factors which have direct impact on each of the risk components. For example, we can elicit that the effectiveness of countermeasure is impacted directly by its configuration and applicability. Based on the extent that threat can exploit vulnerabilities, we can elicit vulnerability exposure and ease of exploit. In our example, the rest of elicited entities may be applicability of countermeasure, configuration of countermeasure, vulnerability type, and vulnerability ease of exploit, vulnerability exposure to public, threat type, and threat frequency of occurrence.

Step 4. Continue to elicit risk factors based on the new risk factors elicited in step 3. This step continues until we reach the “leaf” risk factors where the probabilities are available (i.e., directly from the statistical data or experts’ direct estimation). In our example, the complexity of countermeasure, usability of countermeasure, maintenance of countermeasure, countermeasure type, personal skills, and training plan can be elicited in this step.

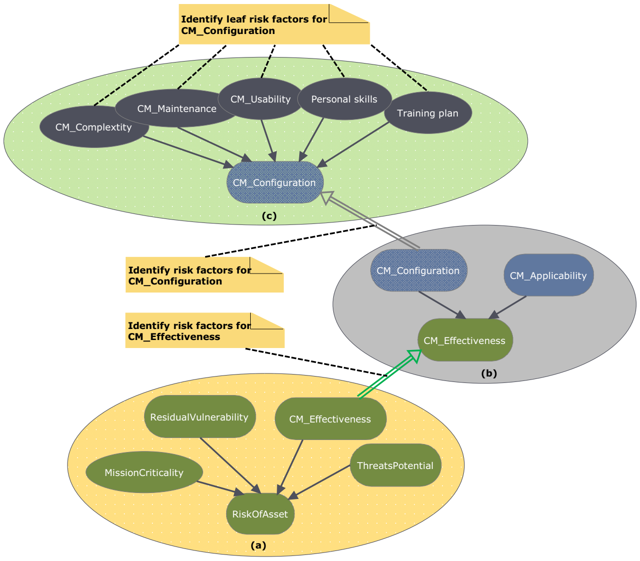


Figure 4. Identify (a) risk components related to target entity; (b) risk factors related to risk components; (c) risk factors until to reach leaf risk factors.

Step 5. Build elementary causal building blocks (idioms). We identify the entities from the candidate pool. We begin at the root level (i.e., the risk components level), then to the leaf risk factor level. Firstly, we identify risk components related to our target entity, namely, “RiskOfAsset”, so “ThreatsPotential”, “MissionCriticality”, “Vulnerability”, “CM_Effectiveness” can be identified, as shown in Figure 4 (a), then we identity risk factors for each of the parents in previous step, for example, we identify the risk factors of “CM_Applicability” and CM_Configuration” that are related to “CM_Effectiveness”, as shown in Figure 4 (b), continue this process until all the newly identified entities become leaf risk factors. For example, we can indentify from the candidate pool that “personal skills”, “CM_Complextity”, “CM_Maintenance”, “CM_Usability” and “training plan” have impacts on “CM_Configuration”, as shown in Figure 4 (c). Note that the causal fragments obtained here may be modified in the following steps.

Step 6. Sometimes the existing entities in the pool cannot be grouped directly in a reasonable way, so we should introduce “connector entities” to connect them. For example, we have to introduce the inherent vulnerability and residual vulnerability to reflect the mitigation effectiveness after certain countermeasures are applied. Also, after the new connector entities are introduced, we should go back to step 4.

Step 7. Evaluate obtained fragments. Once we get the initial causal fragments, we should evaluate them based on the needs and constraints within our domain. If the building block (idiom) is not suitable, it should be revised. In our example, though the “personal skills” and “training plan” have impacts on correctness of countermeasures, but our current discussion does not take much of practical operational details into account, so we will discard these two entities at this moment.

Step 8. Combine obtained fragments. It is easy to notice that some common entities are shared by different building blocks, after each building block has been re-evaluated; we can then combine them together based on these common entities. In addition, there may be some causal relationships among the risk factors in different fragments and they must be identified. For example, we can identity that the “vulnerability type” has impact both on “threat potential” and “countermeasure applicability”. Since the BBN is a Directed Acyclic Graph, we have to make sure that the composite model satisfies the constraints of BBN. Finally, the structure of BBN is obtained.

Step 9. For each entity, it must have a finite number of mutually exclusive, collectively exhaustive states. Different entities may have different states or measurement scales. For example, “threat potential” may have three states (High, Medium, Low), while “countermeasure maintenance” may have four states based on the cost (under $1000, $1000-$2000, $2000-$3000, more than $3000). For simplicity, we assume that all the entities (except the nodes ThreatType, VulnerabilityType and CM_Type, their states are based on their possible types in the given scenario) have three possible states: High, Medium and Low. In real-world, we can use other available practical measurement scale to improve the accuracy and understanding of our assessment.

Step 10. Based on the historic statistical data and estimations from the experts, build the NPT for each entity. For the leaf node, directly assign the probability to each state; for non-leaf node, assign the probabilities to its states under all the possible combinations of its parents’ states. Table I shows the NPT of leaf node “CM_Complexity”.

Table II shows the NPT of non-leaf node “CM_Effectiveness”. When the whole process is finished, the final BBN is constructed. The nodes in the BBN are causally linked and the NPTs indicate the strength of these links. Our final BBN is shown in Figure 5. Due to the limited space, the entire NPTs are not shown in the paper.

C. Applying BBN into Operational Scenarios

In order to assess the risk of real software systems in a socio-technical environment [33], we have used the operational scenarios of the target system as “triggers” for the discovery of applicable C&A requirements. After the relevant C&A requirements are discovered, the risk components associated with them can also be obtained.

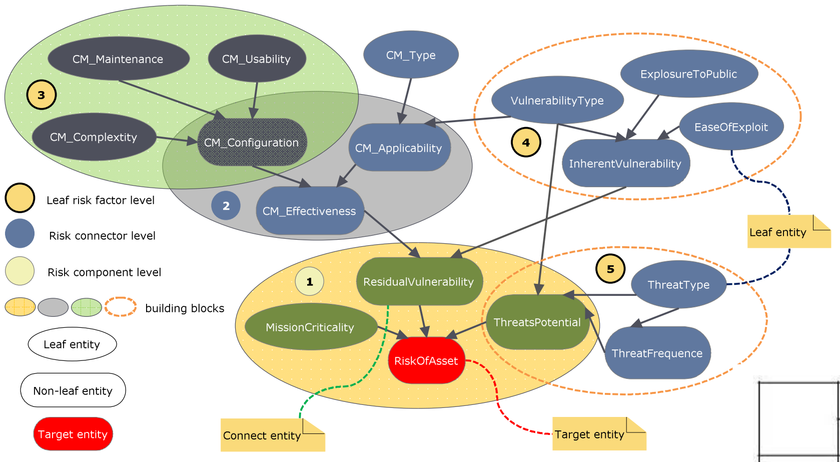


Figure 5. The BBN for Risk Assessment

TABLE I. NPT OF NODE CM_COMPLEXITY

Possible States	Probability
High	0.6
Medium	0.3
Low	0.1

TABLE II. NPT OF NODE CM_EFFECTIVENESS

	CM_Applicability	High (H)			Medium (M)			Low (L)		
	CM_Configuration	H	M	L	H	M	L	H	M	L
CM_Effectiveness	H	0.9	0.8	0.7	0.7	0.6	0.5	0.5	0.4	0.2
	M	0.1	0.1	0.2	0.2	0.3	0.3	0.3	0.3	0.2
	L	0	0.1	0.1	0.1	0.1	0.2	0.2	0.3	0.6

The BBN we built in section IV.B, then can be used into this scenario to assess the risk for the target assets. Risk assessment based on operational scenarios has two main advantages: 1) operational scenarios of the target system can be easily obtained from domain experts and other artifacts (e.g. use cases); and 2) assumptions about the coverage of the assessment are already addressed in the development of the scenario building process. In our previous research [16], we have learned that each operational scenario is related to multiple requirements. In this situation, we can assess the risk for the asset in each related requirement, and then calculate its risk as a whole by using appropriate aggregation methods (i.e. weight average method).

It is notable that the BBN shown in Figure 5 is useful when there is only one instance for each of the risk factors. However, in a real-world scenario, we have to add more nodes to represent each possible risk factor, but the numbers of threats, countermeasures and vulnerabilities are all different from scenario to scenario. Since it is impossible to build a BBN for each scenario, the expected BBN should have the ability to treat each instance of risk components individually, and systematically aggregate their impacts later together. In the next section, we will exploit the expressiveness and reasoning power of MEBN logic to address this specific problem.

D. *Modeling Risk Assessment with MEBN Logic*

1) Overview of MEBN Logic

In the previous section, we discussed the limited expressiveness of the standard BBN. Normally, BBN assumes a simple attribute-value representation, that is, each problem instance involves reasoning about the fixed number of attributes, with only the evidence values changing from problem instance to problem instance [7]. Thus standard BBNs lack the expressive power to represent entity types (e.g., threat) that can be instantiated dynamically, as many times as required for the situations. MEBN is a good fit for this need with the expressive power of first-order logic with a sound and logically consistent treatment of uncertainty, which provides syntax, a set of model construction and inference processes, and semantics that together provide a means of defining probability distributions over unbounded and possibly infinite numbers of interrelated hypotheses [7].

MEBN logic represents the world as comprised of entities that have attributes and are related to other entities. Random variables (RVs) represent features of entities and relationships among entities. Knowledge about attributes and relationships is expressed as a collection of MEBN fragments (MFrags) organized into MEBN Theories (MTheories) [24]. An MFrag consists of RVs, a fragment graph, and a set of local distributions. Each MFrag has an associated set of RVs that are partitioned into context, input, and resident RVs. A variable in an MFrag may have a list of arguments that are placeholders for entities in the domain and makes MFrags parameterized. The local probability distribution for resident RV is defined in the MFrag itself, through a description function which specifies how to assign probabilities to the given its parents. The probability distribution for each input RV is defined in other MFrag in which it acts as a resident node. Context RVs are boolean nodes collectively specify conditions under which the local distributions for the resident RVs apply.

A generative MTheory summarizes statistical regularities that characterize a domain and also introduces necessary mechanism to ensure these MFrags collectively satisfies consistency constraints and the existence of a unique joint probability distribution over an unbounded, possibly infinite number of instances of the RVs represented in each of the MFrags within the set [7]. To apply a generative MTheory into a specific scenario, we need to instantiate it with specific information to form a Situation-Specific Bayesian Network (SSBN). Then, standard Bayesian inference can be used on this SSBN to answer query (e.g., what is the probability of target asset to have a high risk level?), to refine the MTheory (e.g., each new evidence gives us additional statistical data to refine the local distributions of the RVs), and to refine our underlying PDO (e.g., the low applicability of the countermeasures can promote experts to identify more effective applicable countermeasures from the given requirements).

One of the important advantages of MEBN is its clarity and modularity. This gives us a flexible and powerful way to build our knowledge base for a specific domain.

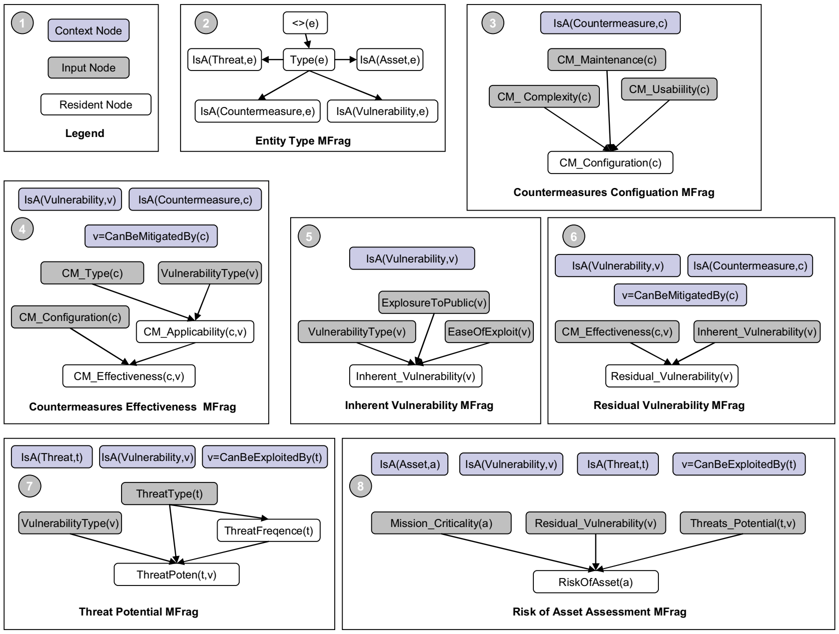


Figure 6. A Generative MTheory for Probabilistic Risk Assessment Model

```
distribution [Hi,Me,Lo]=function{
  for v in parents(Inherent_Vulnerability(v)) {
    if (Vi->c) {
        if(ExplosureToPublic(Vi))==Hi && EaseOfExploit(Vi)==Lo) then
            [Hi:0.35,Me:0.5,Lo:0.15];
        else if (ExplosureToPublic(Vi)==Hi && EaseOfExploit(Vi)==Me) then
            [Hi:45,Me:25,Lo:3];
        else if (ExplosureToPublic(Vi)==Lo && EaseOfExploit(Vi)==Hi) then
            [Hi:0.2,Me:0.4,Lo:0.4];
        else if (ExplosureToPublic(Vi)==Lo && EaseOfExploit(Vi)==Me) then
            [Hi:0.1,Me:0.3,Lo:0.6];
        else if (ExplosureToPublic(Vi)==Me && EaseOfExploit(Vi)==Me) then
            case VulnerabilityType(v)){
            Software:[Hi:0.4,Me:0.25,Lo:0.35];
            Network:[Hi:0.2,Me:0.3,Lo:0.5];
            Others:[Hi:0.1,Me:0.1,Lo:0.8];
            }
         else [Lo:15,Me:30,Hi:55];
       }}}
```

(5) in Figure 6

Figure 8. Local Distribution for Inherent_Vulnerability(v)

```
distribution [Hi,Me,Lo]=function{
  for c,v in parents(Residual_Vulnerability(v)) {
    if any(Ci->c) {
        if Inherent_Vulnerability(v)==Hi then
              [Hi: 1-min(1:(0.8*number(Ci,have(CM_effectiveness(Ci)==Hi))+
                     (0.6*number(Ci,have(CM_effectiveness(Ci)==Me))+
                     (0.4*number(Ci,have(CM_effectiveness(Ci)==Lo)),
               Me:(1-Hi)*0.6, Lo:(1-Hi)*0.4]
         else if Inherent_Vulnerability(v)==Me then
              [Hi: 0.6-min(0.6:(0.8*number(Ci,have(CM_effectiveness(Ci)==Hi))+
                     (0.6*number(Ci,have(CM_effectiveness(Ci)==Me))+
                     (0.4*number(Ci,have(CM_effectiveness(Ci)==Lo)),
               Me:(1-Hi)*0.8, Lo:(1-Hi)*0.2]
         else if Inherent_Vulnerability(v)==Lo then
               [Hi: 0.2-min(0.2:(0.8*number(Ci,have(CM_effectiveness(Ci)==Hi))+
                     (0.6*number(Ci,have(CM_effectiveness(Ci)==Me))+
                     (0.4*number(Ci,have(CM_effectiveness(Ci)==Lo)),
               Me:(1-Hi)*0.2, Lo:(1-Hi)*0.8]
      }} }
```

(6) in Figure 6

Figure 9. Local Distribution for Residual_Vulnerability(v)

In addition, new MFrags can be added into MTheory without impacting existing ones, as long as these MFrags together satisfy the constraints (existence of a unique joint probability distribution) of the MTheory. Although other modeling techniques are also attractive, they are less suitable in the domain of risk assessment. As an example, OOBN provides a natural way to represent uncertainty about the attributes of instances of different types of objects (risk factor), but the problem of OOBN is that the instances of a same object have not only the same structure but also the same probability distribution. In real risk assessment process, the probability distribution of instances of the same object may have a big difference. For instance, if we have an object of "Threat Mitigates Vulnerability", two instances can be "Threat T1 mitigates Vulnerability V" and "Threat T2 mitigates Vulnerability V", obviously, T1 and T2 may have totally different effect on vulnerability V, and thus we should assign different probability distribution to these two different threats instances. On the contrary, MEBN can easily represent this situation.

2) *Building MEBN Model for Risk Assessment*

Based on the BBN shown in Figure 5, we can extract relevant MFrags, then we can collect these MFrags to form an MTheory. Figure 6 shows an example of generative MTheory. The first MFrag is the Entity Type MFrag which is used to formally declare the possible types of entities in the model. Other MFrags are used to model the risk factors and their causal relationships.

MEBN uses an expressive language to define the probability distributions for RVs. Due to the limited space, we only give the local distribution functions for the RVs VulnerabilityType(v), Initial_Vulnerability(v) and Residual_Vulnerability(v), other local distribution functions and the default distributions are not shown here.

Local distributions in standard BBNs are typically represented by static tables (NPTs), which limit each node to a fixed number of parents, while an instance of a RV in an MTheory might have any number of parents, because its parents are also RVs and can be instantiated as many times as necessary. We use pseudo-code suggested in [7] to convey the idea of using local distributions to specify probability distributions. It is possible that different people have different local distribution functions for the same RV.

```
distribution [Software,Network,Others]=function{
 for v in VulnerabilityType(v)){
    if (Vi->v) then [Software:0.4,Network:0.2,Others:0.2];
    }}
```

Figure 7. Local Distribution for VulnerabilityType(v)

Leaf RV has no parents and its distribution is assigned directly. Figure 7 gives an example to define the local distribution for leaf RV VulnerabilityType(v). For any given vulnerability, the probability of vulnerability type to be "Software", "Network", "Others" is 0.4, 0.2, and 0.2, respectively. The vulnerability types and their probabilities can be different in a specific environment.

Figure 8 defines the local distribution for RV Inherent_Vulnerability(v). This variable has its parents but they share the same ordinary variable. Thus, for each instance of Initial_Vulnerability(v), it has fixed number of parents, namely, the three instances of VulnerabilityType(v), ExplosureToPublic(v) and EaseofExploit(v). The same way to build the NPTs for nodes in the standard BBNs, we use description function to assign the probabilities covering all of the possible combinations of its parents.

Figure 9 defines the local distribution for RV Residual_Vulnerability(v). This variable has parents and there is one external variable c in its parent CM_Effectiveness(c,v).

In this situation, the instances of CM_Effectiveness(c,v) depend on the number of countermeasures which can mitigate this vulnerability in the given scenario, therefore, we must take the number of countermeasures into account. For example, as shown in Figure 10, if there are two countermeasures !C0 (entity identifier, begins with an exclamation point, represent an instance of an entity) and !C1 can mitigate the vulnerability !V. When the Inherent_Vulnerability(!V) is High, the CM_Effectiveness(!C0,!V) is High and CM_Effectiveness(!C1,!V) is Medium, we can identify the number of countermeasure with High, Medium, Low effectiveness are 1, 1, 0, respectively. According to the local distribution described in Figure 9, the final probability distribution for Residual_Vulnerability(!V) will be [Hi:1-min(1:(0.8*1+0.6*1+0.4*0)), Me:(1-Hi)*0.6, Lo:(1-Hi)*0.4], namely, [Hi:0,Me:0.6,Lo:0.4] as shown in Figure 10.

By modeling risk assessment using MEBN logic, each risk factor is represented by a corresponding RV which can be instantiated dynamically as many times as necessary, so all the risk factors in the given scenario can be treated at the instance level. In addition, MEBN logic provides a consistent, flexible way to define the local probabilities for the RVs. Through this way, different group stakeholders will have a clear common understanding about how the original information is collected, how these information are combined or integrated, how these information are transformed from one node to other nodes and how these information are used to support or deny an argument from the risk assessment. The risk components and causal relationships in the Risk Model have their corresponding matches in the MTheory. Other relationships become the foundation or context of the operational scenario under which the MTheory is applied. Therefore, we can assume that the MTheory is consistent with the Requirements and Risk Model in our discourse of domain. The complete mappings between Risk Model and MTheory are shown in Table III.

3) *Inference by using MEBN model for Risk Assessment*

The generative MTheory must be instantiated into an SSBN to compute the response to a query under the given scenario. Although the generative MTheory in Figure 6 implicitly represents infinite possible operational scenarios in our domain, we take the simplest one, namely, there is only one instance for each type risk component, to demonstrate how it can be used in a real risk assessment. Figure 11 illustrates a SSBN instantiated from the generative MTheory. We assume the instances of risk components in this simplest scenario are asset !A(ENCLAVE), threat !T(UNAUTHORIZED INTERNET ACCESS), vulnerability !V(FIREWALL AND IDS MIS-CONFIGURATION), countermeasure !C(INSTALL FIREWALLS & IDS AT KEY POINTS IN THE ENCLAVE WITH APPROPRIATE CONFIGURATIONS), These scenario specific information is used to instantiate the MTheory and create instances of RVs. The details of the algorithm and processes of how to construct SSBN are in [7] [24].

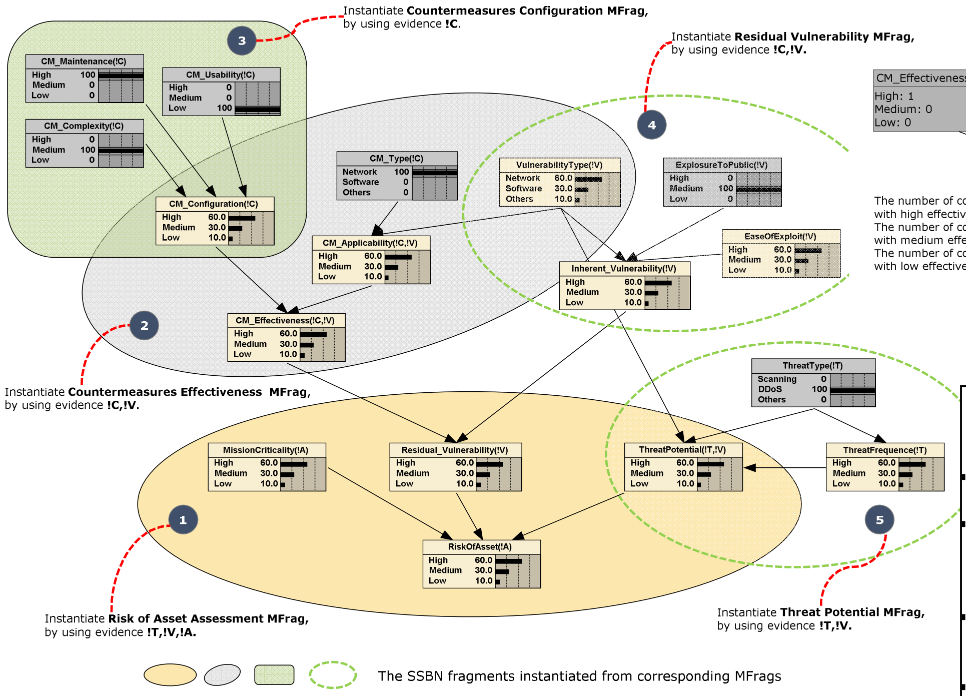
After the SSBN has been constructed, it can be treated as a standard BBN. When a query (which contains a finding set of particular information about the situation and a target set of the nodes of interests) comes into SSBN, the inference works the same way as a standard BBN to calculate the response for input query. For example, Figure 11 can answer a set of query: “When there is a threat DDoS can_exploit the Software vulnerability while suggested countermeasure type is Network with low complexity, high usability and medium maintenance, what is the risk of asset A?” Or “what is the countermeasure applicability of C?”.

TABLE III. THE MAPPING BETWEEN RISK MODEL AND MTHEORY

(Requirements and Risk Model) => (Generative MTheory)
(Security Requirements and Assets, Security Requirements suggest Countermeasures, Security Requirements prevent Vulnerabilities, Security Requirements driven by Threats, Countermeasures have Vulnerabilities, Security Requirements apply to Assets) => (The operational scenario under which the risk components are discovered and the MTheory applied to assess the risk of the given assets)
(Threats) => (Threat_Potential(t,v), and Threat Potential MFrag)
(Vulnerabilities) => (Inherentl_Vulnerability(v), Inherent Vulnerability MFrag and Residual_Vulnerability(v), Residual Vulnerability MFrag)
(Countermeasures) => (CM_Effectiveness(cm,v))
(Risks) => (RiskOfAsset(a))
(Mission Criticality) => (Mission_Criticality(a))
(Threat Exploit Vulnerabilities, Threats increase risks, Threats damage assets, Assets have mission criticality, Vulnerabilities lead to risks) => (Risk of Asset Assessment MFrag)
(Countermeasures mitigate Vulnerabilities) => (Residual Vulnerability MFrag) (Countermeasures reduce risks) => (Ignored since the countermeasures reduce the risks by mitigating the vulnerabilities)

Simple sensitivity analysis and comparison can be done based on Figure 11 to support the decision making process. For example, in order to reduce the risk of A, we can either improve the maintenance of countermeasure or apply a new countermeasure. So we can fix other conditions and only change the CM_Maintenance(!C) from High to Low, we can get corresponding results of RiskOfAsset(!A), as shown in Table IV. Similarly, we can check CM_Type(!C) and make another table, as shown in Table V. Therefore, we can find out that changing the countermeasure type is more cost-effective than improving the maintenance of countermeasure. These types of sensitivity analysis and comparisons can be used to support the complex decision-making and help finding the “best” solution to reduce the risk of target asset.

MEBN can be instantiated in a systematic way and the final SSBN can be generated automatically. For example, some risk factors have multiple instances and the risk assessment scenario can be described as follows: The target asset !A0 (ENCLAVE) may be damaged by threats !T0 (UNAUTHORIZED INTERNET ACCESS) and !T1 (UNAUTHORIZED NETWORK TRAFFIC) which may exploit the vulnerabilities !V0 (INTERNET ACCESS NOT PROXIED), !V1 (FIREWALL AND IDS MIS-CONFIGURATION) and !V2 (USE OF TAMPERED SOFTWARE), to mitigate the vulnerabilities, the suggested countermeasure are !C0 (INSTALL FIREWALLS & IDS AT KEY POINTS IN THE ENCLAVE WITH APPROPRIATE CONFIGURATIONS) and !C1(MANAGED INTERNET ACCESS CONTROL POINTS (DMZ)). Based on the above information, the generative MTheory can be instantiated to a new SSBN (top of Figure 12) that has relevant RVs and MFrags instantiated multiple times. The bottom of Figure 12 shows a subset (the shadowed part) of the SSBN shown in the top of Figure 12.



Instantiate **Countermeasures Configuration MFrag**, by using evidence **!C.**

Instantiate **Residual Vulnerability MFrag**, by using evidence **!C,!V.**

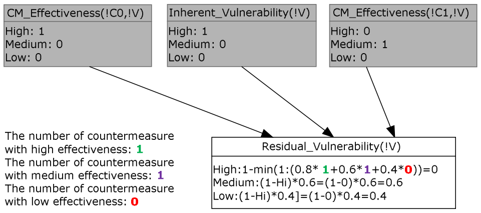
Instantiate **Countermeasures Effectiveness MFrag**, by using evidence **!C,!V.**

Instantiate **Risk of Asset Assessment MFrag**, by using evidence **!T,!V,!A.**

Instantiate **Threat Potential MFrag**, by using evidence **!T,!V.**

The SSBN fragments instantiated from corresponding MFrags

Figure 11. SSBN for the simplest risk assessment scenario

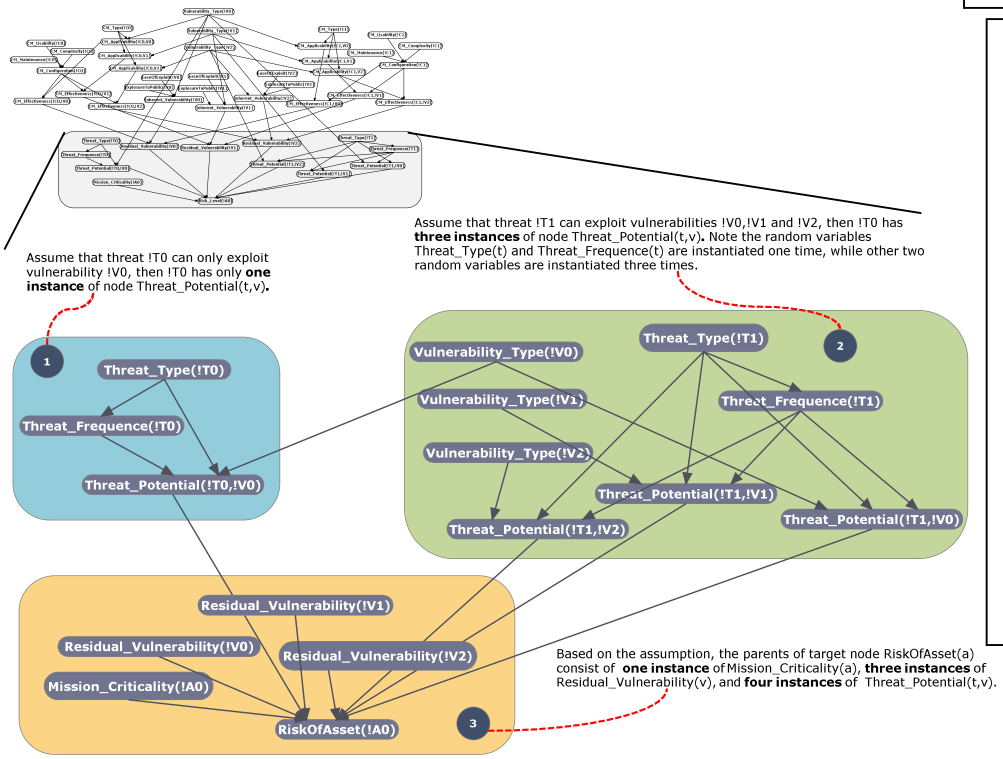

CM_Effectiveness(!C0,!V)
High: 1
Medium: 0
Low: 0
Inherent_Vulnerability(!V)
High: 1
Medium: 0
Low: 0
CM_Effectiveness(!C1,!V)
High: 0
Medium: 1
Low: 0
Residual_Vulnerability(!V)
High:1-min(1:(0.8* 1+0.6*1+0.4*0))=0
Medium:(1-Hi)*0.6=(1-0)*0.6=0.6
Low:(1-Hi)*0.4]=(1-0)*0.4=0.4

The number of countermeasure with high effectiveness: 1
The number of countermeasure with medium effectiveness: 1
The number of countermeasure with low effectiveness: 0

Figure 10. Example of local distribution calculation

The process of inference in MEBN Model for Risk Assessment:

- *Specify the operational scenario, identify the relevant risk components.*
- *Based on the information in the given operational scenario, construct the SSBN by creating and combining instances of the MFrags in the generative MTheory.*
- *Create query and apply standard Bayesian network inference algorithm in SSBN*
- *Answering the query by inspecting the posterior probabilities of the target nodes.*



Assume that threat !T0 can only exploit vulnerability !V0, then !T0 has only **one instance** of node Threat_Potential(t,v).

Assume that threat !T1 can exploit vulnerabilities !V0,!V1 and !V2, then !T0 has **three instances** of node Threat_Potential(t,v). Note the random variables Threat_Type(t) and Threat_Frequence(t) are instantiated one time, while other two random variables are instantiated three times.

Based on the assumption, the parents of target node RiskOfAsset(!A) consist of **one instance** of Mission_Criticality(a), **three instances** of Residual_Vulnerability(v), and **four instances** of Threat_Potential(t,v).

Figure 12. Generated SSBN for the simple risk assessment scenario (Top) + Details of the subset of SSBN (Bottom)

TABLE IV. NPT OF NODE CM_MAINTENANCE

CM_Maintenance(!C)	Cost	RiskOf Asset(!A)
High	$1000	[0.4,0.2,0.4]
Medium	$500	[0.5,0.2,0.3]
Low	$200	[0.6,0.2,0.2]

TABLE V. NPT OF NODE CM_TYPE

CM_ Type (!C)	Cost	RiskOf Asset(!A)
Network	$1000	[0.1,0.2,0.7]
Software	$500	[0.4,0.3,0.3]
Others	$200	[0.6,0.2,0.2]

This subset contains two MFrags, "Threat Potential MFrag" and "Risk of Asset MFrag". Based on the given operational scenario and the constraints of the context node such as v=CanBeExploitedBy(t), assume that threat !T0 can only exploit vulnerability !V0, whilst !T1 can exploit vulnerabilities !V0, !V1 and !V2 to represent the risk level of asset !A0, use (!T0 and !V0), (!T1 and !V0), (!T1 and !V1), (!T1 and !V2) to instantiate the RVs and two MFrags, then we will get one instance of Mission_Criticality(a) and RiskOfAsset(a), two instances of Threat_Type(t) and Threat_Frequence(t), three instances of Vulnerability_Type(v) and Residual_Vulnerability(v), and four instances of Threat_Potential(t,v). Since the number of risk factors in real-world risk assessment case can be huge, this paper uses the open source software, UnBBayes tool [4], to compute with MEBN models.

V. SUMMARY AND FUTURE WORK

In this paper, we introduce the use of BBNs and MEBN logic to overcome the limited expressiveness and model the process of risk assessment and analyze the correlations among different risk components. This paper presents a step-by-step approach to indentify the risk related factors based on the Risk Model and build a causal model for the probabilistic risk assessment. By using MEBN logic, we build a generative MTheory for risk assessment which implicitly represents infinite possible operational scenarios in the domain of risk assessment. The generative MTheory for Risk Assessment can be instantiated by using the specific information in the given operational scenario; therefore our model can treat each individual instance of risk components dynamically and process automatically. Our current model is based on a simple operational scenario and there is an implicit assumption in our model, that the relevant risk components are completely discovered via security requirements. But in a real-world situation, it is highly possible that the risk components may not be fully discovered. Therefore, it is very important to take advantage of the known explorative study on the types of threats, vulnerabilities and countermeasures for different kinds of platforms and situations. We plan to apply our models and methods into more complex operational scenarios which contain multiple correlated requirements from different level of abstractions, with the extension of the models to support the temporal recursion. We will also develop a well-designed case study to experiment and validate our Probabilistic Risk Assessment Model.

REFERENCES

[1] Common Criteria, Ver. 2.1. ISO/IEC 15408-1, 1999.

[2] DoD Instruction 5200.40: DITSCAP, 1997.

[3] DoDI 8500.2. IA Implementation, 2003.

[4] Vulnerability Analysis of Certificate Validation Systems, www.corestreet.com

[5] Aagedal, J.O. and den Braber, F., Model-based risk assessment to improve enterprise security. in *Proc. of the 6th Int'l Enterprise Distributed Object Computing Conf.*, (2002), 51-62.

[6] Butler, S.A., Security Attribute Evaluation Method: A Cost-Benefit Approach. in, (2002), 232.

[7] da Costa, P.C.G. and Laskey, K.B., Multi-Entity Bayesian Networks Without Multi-Tears. *George Mason University, SEOR* (2005).

[8] Davis, T. Federal Computer Security Report Card Grades 2004 *Press Release*, Government Reform Committee, 2004.

[9] Ekelhart, Fenz, Klemen and Weippl, Security Ontologies: Improving Quantitative Risk Analysis. in, (2007), 156a.

[10] Elliott, R.J., Aggoun, L. and Moore, J.B. Hidden Markov Models: Estimation and Control, Springer-Verlag, Berlin, 1995.

[11] Ernst and Young. Report on the Widening Gap *8th annual Global Information Security Survey*, Netherland, 2005.

[12] Feather, M. and Cornford, S. Quantitative risk-based requirements reasoning. *Requirements Engineering*, *8* (4). 248-265.

[13] Feather, M.S., Maynard-Zhang, P. and Kiper, J.D. Modeling uncertainty in requirements engineering decision support, Pasadena, CA : Jet Propulsion Laboratory, NASA, 2005.

[14] Fenton, N. and Neil, M. A Critique of Software Defect Prediction Models *IEEE Transactions on Software Engineering*, 1999, 675-689.

[15] Fenton, N. and Neil, M. Making Decisions: Using BNs and MCD, CS Dept., Queen Mary and Westfield College, London, 2000.

[16] Gandhi, R.A. and Lee, S.-W., Discovering and Understanding Multi-dimensional Correlations among Certification Requirements with application to Risk Assessment. *Req Engg Conf.* (2007), 231-240.

[17] Getoor, L., Friedman, N., Koller, D. and Pfeffer, A. Learning Probabilistic Relational Models *Relational Data Mining*, Berlin: Springer-Verlag, Saso Dzeroski and Nada Lavrac (ed.), 2001.

[18] Gilks, W., Thomas, A. and Spiegelhalter, D.J. A language and program for complex Bayesian modeling *The Statistician*, 1994.

[19] Gregoriades, A. and Sutcliffe, A. Scenario-based assessment of nonfunctional requirements. *IEEE Trans on Soft Engg*, *31* (5). 392-409.

[20] Grenander, U. Elements of Pattern Theory, Johns Hopkins 1996.

[21] Hu, Y., Chen, J., Huang, J., Liu, M.A.L.M. and Xie, K.A.X.K., Analyzing Software System Quality Risk Using Bayesian Belief Network. in *Granular Computing, 2007. GRC 2007.*

[22] Hui, A.K.T. and Liu, D.B. A Bayesian Belief Network Model and Tool to Evaluate Risk and Impact in Software Development Projects *Reliability and Maintainability,2004 Annual Symposium*, 2004, 297-301.

[23] Koller, D. and Pfeffer, A. Object-Oriented Bayesian Networks *Uncertainty in Artificial Intelligence: Proceedings of the Thirteenth Conference*, San Francisco, CA, Morgan Kaufmann., 1997.

[24] Laskey, K.B. MEBN: A language for first-order Bayesian knowledge bases. *Artificial Intelligence*, *172* (2-3).2005, 140-178.

[25] Lee, S.W., Muthurajant, D., et al. Building decision support problem domain ontology from natural language requirements for software assurance. *IJSEKE*, *16* (6). 851-884.

[26] Littlewood, B., Popov, P. and Strigini, L. Assessment of the reliability of fault-tolerant software: A Bayesian approach. *Computer Safety, Reliability and Security, Proceedings*, *1943*. 294-308.

[27] Littlewood, B. and Wright, D. The Use of Multilegged Arguments to Increase Confidence in Safety Claims for Software-Based Systems: A Study Based on a BBN Analysis of an Idealized Example. IEEE *Transactions on Software Engineering*, *33* (5). 347-365.

[28] Mkpong-Ruffin, I., Umphress, D., Hamilton, J. and Gilbert, J. Quantitative software security risk assessment model *ACM workshop on Quality of protection*, Alexandria, Virginia, USA, 2007.

[29] Neil, M. The SERENE Method Manual versoin 1.0, 1999.

[30] Neil, M., Fenton, N. and Nielsen, L. Building large-scale Bayesian networks. *Knowledge Engineering Review*, *15* (3). 257-284.

[31] Report, U.G. Department of Homeland Security Needs to Fully Implement its Security Program *05-700*, 2005.

[32] Verdon, D. and McGraw, G. Risk Analysis in Software Design *IEEE Security & Privacy Magazine*, 2004, 79-84.

[33] Voas, J. Certifying software for high-assurance environments. *IEEE Software*, *16* (4). 48-54.

[34] Yacoub, S.M. and Ammar, H.H. A methodology for architecture-level reliability risk analysis. *IEEE Trans on Soft Engg*, *28* (6). 529-547