# High-Level Information Fusion of Cyber-Security Expert Knowledge and Experimental Data

Paulo C. G. Costa and Bo Yu
Volgenau School of Engineering
George Mason University
Fairfax, VA 22030 USA
Email: [pcosta;byu3]@gmu.edu

Michael Atighetchi
Raytheon BBN Technologies
Cambridge, MA 02138 USA
Email: michael.atighetchi@raytheon.com

David Myers
Air Force Research Laboratory
Rome, NY 13441 USA
Email: david.myers.35@us.af.mil

***Abstract*—High-Level Information Fusion (HLIF) provides the ability to combine data from diverse sources, including documents involving analyst assessment and raw sensor reports generated by sensors, in a coherent and consistent way. Command and Control (C2) in cyber infrastructure involves gathering information from experts, merging it with field knowledge and experimental results, and selected the most appropriate cyber assets to deploy at any given time in the mission cycle. When framing cyber asset selection as a HLIF problem, one key aspect involves estimation of network-wide impacts generated by cyber assets. Cyberspace is a highly dynamic man-made domain with a high degree of uncertainty and incomplete data which must be transformed into knowledge to support precise and predictable cyber effects estimation. Current systems have to rely on human subject matter experts (SMEs) for most tasks, rendering the cyber asset planning process too time consuming and therefore operationally ineffective. This paper proposes an architecture that leverages probabilistic ontologies to expedite the cyber asset planning process, allowing for the automation of most time-consuming, error-prone, SME-based knowledge elicitation under uncertainty. We illustrate the main aspects of the proposed architecture through examples taken from the Derived and Integrated Cyber Assets (DICE) project.**

DISTRIBUTION A: Approved for public release; distribution unlimited (Case Number 88ABW-2018-1403).
Work sponsored by AFRL under contract FA8750-17-C-0209; the views and conclusions contained in this document are those of the authors and not AFRL or the U.S. Government.

## I. Introduction

The cyber domain is inherently adversarial, as it always includes the presumption of rogue agents aiming to disrupt, deny, degrade, or deceive current missions. From a cyber defense perspective, these rogue agents are controlled by adversaries who are constantly attempting to access the defender's network for their advantage. The plethora of attack vectors available today, using techniques that cover a large spectrum in terms of complexity, size, capabilities, and other factors makes cyber defense a daunting task; one that is aggravated by the lack of information about the enemy, its capabilities, intentions, and modus operandi. Further, enhancing the effectiveness of cyber defenses requires understanding how they interact together in face of different attacks. From a full-spectrum cyber mission perspective, a similar problem exists of selecting the most appropriate asset to achieve the current mission objective and configuring the asset in a certain way that maximizes a utility function across multiple attributes, including the accuracy in achieving the intended affect (e.g., deny), the precision at which the affect is obtained (e.g., only intended hosts are denied), the latency associated with generating the intended effect, and the ability to remain undetected and non-attributable.

In an adversarial environment, fulfilling these knowledge needs involves estimating what is not known based on what is known. A common way of doing this involves capturing and modeling expert knowledge and using that model in conjunction with observations to estimate the most likely cause of what was observed. Another is to conduct experiments on a controlled environment in a way that generates results that can be extrapolated to more general but similar conditions.

In spite of the techniques used in either of these two approaches, data entering the system will be highly heterogeneous in nature and fraught with uncertainty. As a result, merging such data consistently and computing its aggregated results in a coherent way is a typical high-level information fusion problem. More specifically, in the information fusion community a distinction is commonly made between *low-level* and *high-level* fusion. Low-level Information Fusion (LLIF) combines sensor reports to identify, classify, or track individual objects. High-level Information Fusion (HLIF) extends it by also combining information from "soft sources" (e.g. SME reports, social media, etc.), as well as contextual information, to characterize a complex situation, draw inferences about the intentions of actors, and support process refinement (cf. [1], [2]). However, while LLIF has Probability Theory as the technology behind how uncertainty is handled, the same cannot be said about HLIF. For instance, ontologies have been widely considered as a means to enable automated systems to perform HLIF tasks (e.g., [3], [4], [5]), but they have no standardized support for managing uncertainty. Representing uncertainty with ontologies is an active area of research, especially in the area of the Semantic Web (e.g., [6], [7]). Proposed approaches include Bayesian probability (e.g., [8], [9], [10]), extensions to Description Logics (e.g.,[11], [12]), and programming logic alternatives (e.g., [13]).

In this paper, we propose an approach to the high-level information fusion of cyber-security data obtained from both subject-matter expert (SME) knowledge elicitation and

specifically-designed experiments. We illustrate our ideas in the context of work performed on the Derived and Integrated Cyber Effects (DICE) project. After this introduction, Section 2 provides a brief overview of the related work and Section 3 discusses challenges associated with eliciting cyber knowledge. Then, Section 4 illustrates how cyber-security knowledge can be elicited from both SME and controlled experiments. Section 5 presents the HLIF process under uncertainty that is the core subject of this paper, followed by our concluding remarks.

## II. Related Work

The described approach relates to a number of existing approaches for cyber experimentation and modeling.

The Cyber Quantification Framework (CQF) [14] provides a means for structured cyber experimentation and computation of derived attributes for cyber assets similar to the ones described in this paper. However, a number of differences exist between our work and the CQF. First, the CQF focuses on determining the local impact of cyber assets on target host. DICE, in contrast, aims to quantify the network-wide impact. Second, in contrast to DICE's semantic definitions, the CQF contains a syntactic definition of derived attributes. Finally, the CQF does not track uncertainty measures associated with knowledge, while DICE uses a bayesian approach to do so by encoding and managing conditional probability distributions.

The Attack Surface Reasoning (ASR) [15], [16] provides semantically defined metrics for quantifying the attack surface and mission-critical distributed systems. Both ASR and DICE leverage abstract ontologies for the purposes of cyber quantification. While ASR defines models of defenses and quantifies the impact of different defense configurations, DICE focuses on modeling of cyber assets and determining the network-wide impact of various asset configurations.

The Datasculptor framework provides tool support for performing Visual Analytics on Linked Data [17]. Datasculptor enables analysts to lifting and interconnect data from a variety of domains in a manner that explicitly describes the overall data transformation process through a sequence of functions, each defining its inputs through a SPARQL[1] SELECT query and its output through a SPARQL CONSTRUCT query. Definition and provenance of semantic data transformation can therefore be described through semantic overlaps between a chain of function executions. In DICE, we implemented several Datasculptor functions to build up the layered abstractions of the Cyber Impact Ontology and generate bayesian representations of derived attributes.

## III. Key Challenges for Eliciting Cyber-Security Knowledge

The overall objective of DICE is to quantify how well cyber assets will perform when being deployed in a target environment containing a number of cyber defenses that are under external control. How well a cyber asset performs can be described through a collection of derived attributes that align with cyber mission objectives.

[1]https://www.w3.org/TR/rdf-sparql-query/

| Derived Attribute | Aspect | Description |
|---|---|---|
| Success | | Degree to which the asset achieve the intended effect |
| Speed | | Latency associated with executing the asset |
| Detectability | Real-time | Degree to which the asset can be detected during execution |
| | Forensic | Degree to which the asset can be detected during forensic analysis after execution |
| Attribution | Asset | Degree to which asset launcher can be identified based on characteristics of the asset |
| | Location | Degree to which asset launcher can be identified based on asset launch location |
| Collateral Damage | | Degree to which unintended target components are affected |
| Adaptability | | Degree to which the cyber asset can succeed in dynamically changing environments |

Fig. 1. Example Derived Attributes

Figure 1 shows the current set of derived attributes we have started to formalize. First and foremost, the asset needs to achieve its desired effect, a derived attribute we call "Success". "Success" changes per asset type and mission phase. For an asset used during the reconnaissance phase of a cyber mission, this might mean that the asset needs to generate an accurate and precise picture of the target environment. An asset used during a later exploitation phase might be characterized by its ability to deny service to a specific target component for an extended period of time. Next, timeliness of asset impact can be expressed as "Speed", enabling the reasoning to pick the most appropriate asset given current mission timelines. Deployment of a cyber asset generates signals that are generally of value to adversaries. To quantify the impact of these signals, we introduce two more related by different derived attributes - detectability and attribution. While "Detectability" expresses the likelihood that a cyber asset is detected, either in real-time during execution or forensically after execution, "Attribution" captures the degree to which the asset can be linked to its originator after being detected, either based on the uniqueness of the asset itself or its specific invocation (including originating location). The final two attributes address the potential for unintended interaction effects caused by executing the cyber asset in the target environment ("Collateral Damage") and the ability of assets to adapt to changing operating conditions ("Adaptability").

Calculating these attributes in support of cyber C2 must address the following technical challenges associated with cyber asset affect estimation and experimentation.

### A. Aggregate Effects Estimates

Using existing experimentation techniques, it is easy to end up with a database full of low-level observables without a clear understanding of how these observables relate to aggregate measures at the level of abstraction of interest to cyber planners, e.g., the probability of mission success,

covertness, and non-attribution of operations. In recent years, cyber ranges have started to become more readily available across the DoD. Current capabilities used at cyber ranges such as the CQF provide means for producing experimentation topologies including definitions of Virtual Machine (VM) images, sensors, together with means for capturing configuration data from sensors. Current metrics focus on end-to-end probability of success across specific experiments together with performance, mostly expressed as latency measured via a time-to-succeed. Finer-grained metrics for capturing relevant attributes of cyber assets have started to emerge, but are limited to closed formulas that only apply to single hosts. To properly quantify cyber assets, a modeling platform is required that can (1) express important high-level attributes of cyber assets related to both security and cost dimensions in a manner that is understandable by cyber operators, (2) express a large number of low-level observables that can be readily gathered from existing experimentation platforms, and (3) encode logic for computing the high-level attributes from the low-level data in a manner that considers aggregate cross-node impact yet remains to be explainable, reusable, and maintainable.

### B. Uncertainty Representation and Analysis

Full-spectrum mission execution needs to cope with highly dynamic environments, including rapidly changing networks, unreliable asset effects, and highly non-linear adversarial activities. Uncertainty permeates US military, commercial, and adversarial environments used in full-spectrum cyber operations. Uncertainty exists in all information related to cyber asset quantification, including capabilities of the assets themselves as well as configuration information about nodes, networks, and defenses. A successful cyber quantification solution must track uncertainty all the way through the entire set of environments and associated networks. In particular, while it is relatively straight-forward to set up well-defined experiments and evaluate assets, it is guaranteed that the environment encountered in real-world cyber mission execution will be different from what has been set up in virtual environments at home. To be useful and pragmatic, solutions need to cover the case in which cyber operators do not have access to the defense software set up by adversaries. Furthermore, cyber operators may only have a limited understanding of the types of defenses and their configurations. These aspects not only contribute to uncertainty but also significantly limit the usefulness of approaches that solely rely on experimentation, such as the CQF.

## IV. Eliciting Cyber-Security Knowledge

To systematically compute aggregate effects of cyber assets, we propose to represent knowledge about assets, defenses, and target environments in the form of probabilistic models expressed via a collection of cyber ontologies. This allows disparate sources of information to be automatically integrated into a rich graph of interconnected information that is easy to understand and easy to extend.

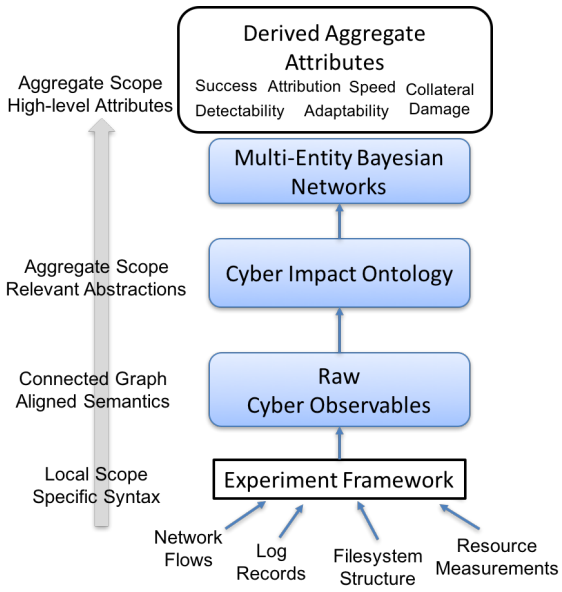


Fig. 2. The Semantic Processing Pipeline

For the purposes of representing knowledge that enables probabilistic quantification of aggregate effects of cyber assets, we have developed a layered approach, as shown in Figure 2, consisting of (1) an ontology for representing detailed cyber observables (a system model) and (2) an ontology that uses specific abstractions that more readily support analysis of interactions between assets and defenses (a cyber impact model), and (3) a collection of Multi-Entity Bayesian Networks (MEBNs) for computing derived attributes.

Figure 3 shows the current implementation prototype that supports quantification of Success, Speed, and Detectability through a collection of implementation classes that parse raw data into triples (Lifters), build up abstractions of the cyber impact ontology (Plugins), and capture the resulting knowledge in the form of Bayesian Networks (Knowledge Fragment).

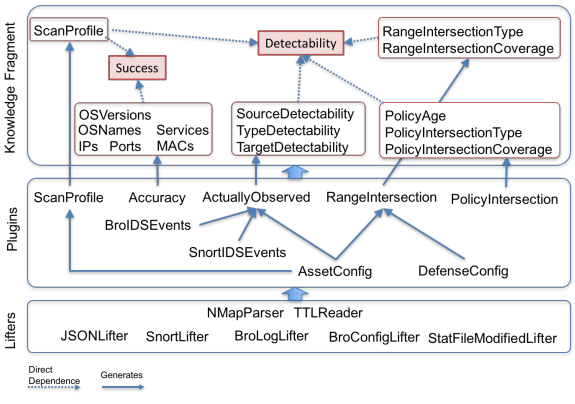


Fig. 3. A current version of the Semantic Processing Pipeline for characterizing the NMap cyber asset

We differentiate between two main types of reasoning in

the cyber impact ontology, effects reasoning and configuration reasoning.

Effects reasoning uses direct observations derived from executing a cyber asset with a specific configuration in a specific target environment. Using experimental data, we can for instance determine how many IP addresses where found by an nmap invocation, how many ports were found, and whether the nmap cyber asset was able to accurately determine the type of operating systems across the target nodes in the experiment topology. We can use this information to calculate the "Success" of an nmap invocation as a measure of accuracy.

Configuration reasoning, in contrast, focuses on analyzing the configurations of various components in the target environment and making predictions based on comparing component configuration with the configuration of the cyber asset. For instance, to predict how likely the nmap cyber asset will be detected, we can analyze the configuration of intrusion detection systems for monitoring range and age of policy, both of which will impact the sensors ability to detect cyber assets. We expect that configuration information plays a central role in cyber C2, as cyber missions are generally focused on obtaining a detailed situational awareness picture of the target environment in early phases of the mission. The power of configuration reasoning is to enable cyber C2 planner to make immediate use of such information to plan asset deployments, without the need to go through a full experimentation cycle.

### A. Modeling SME Knowledge

The representational framework used in DICE, as explained above, has the following key requirements:

1) Represent intricate patterns of uncertainty,
2) Capture knowledge from both SMEs and experiments,
3) An underlying rigorous mathematical foundation, and
4) Efficient and scalable support for probabilistic and logical reasoning.

Bayesian Networks [18] are a very popular way for fusing information under uncertainty, but their expressiveness is limited to propositional logic and thus Bayesian Networks cannot capture the complexity involved in DICE operations. As a result, DICE uses probabilistic ontologies [8], [9]. More specifically, PR-OWL, a probabilistic extension to the web ontology language OWL is used. The mathematical foundation of PR-OWL is Multi-Entity Bayesian Networks (MEBN), which can be seen as a first-order logic version of Bayesian Networks [19].

MEBNs encode probabilistic knowledge as MEBN Fragments (MFrags), which can be seen as templates to build BNs for a given scenario. In general, an MFrag captures a repeatable pattern (e.g. the effect of a specific cyber asset to a network node) and can be instantiated many times as needed to match a specific situation (e.g. that specific cyber asset being employed to a 10-node network). An MFrag is a parametrized fragment of a directed graphical probability model. It represents probabilistic relationships among uncertain attributes of and relationships among domain entities. A set of MFrags that collectively satisfies constraints ensuring a unique joint probability distribution is a *MEBN Theory* (MTheory).

MFrags are templates that can be instantiated to form a joint probability distribution involving many random variables. Such a ground network is called a *situation-specific Bayesian network* (SSBN). MEBNs provide a compact way to represent repeated structures in a Bayesian Network. There is no fixed limit on the number of random variable instances, which can be dynamically generated as needed. The ability to form a consistent composition of parametrized model fragments makes MEBN well suited for knowledge fusion applications [20]. MEBN inference can be performed by instantiating relevant MFrags and assembling them into SSBNs to reason about a given situation. As evidence arrives, it is fused into the SSBN to provide updated hypotheses with associated levels of confidence. These are very convenient features for representing diverse information coming from various sources, which make MEBN attractive as a logical basis for probabilistic ontologies.

## V. HLIF with Probabilistic Ontologies

Ontologies have been used extensively in HLIF applications (e.g. [3], [4], [5]). However, current ontology formalisms deliver a partial answer to requirements listed above, but lack a principled, standardized means to represent uncertainty. This has spurred the development of palliative solutions in which probabilities are simply inserted in an ontology as annotations (e.g. marked-up text describing some details related to a specific object or property). These solutions address only part of the information that needs to be represented, and too much is lost due to the lack of a good representational scheme that captures structural constraints and dependencies among probabilities. A true probabilistic ontology must be capable of properly representing those nuances. More formally:

> Definition 1 (from [8]): A probabilistic ontology (PO) is an explicit, formal knowledge representation that expresses knowledge about a domain of application. This includes:
>
> - Types of entities that exist in the domain;
> - Properties of those entities;
> - Relationships among entities;
> - Processes and events that happen with those entities;
> - Statistical regularities that characterize the domain;
> - Inconclusive, ambiguous, incomplete, unreliable, and dissonant knowledge related to entities of the domain; and
> - Uncertainty about all the above forms of knowledge;
>
> where the term entity refers to any concept (real or fictitious, concrete or abstract) that can be described and reasoned about within the domain of application. ■

POs provide a principled, structured, sharable formalism for describing knowledge about a domain and the associated

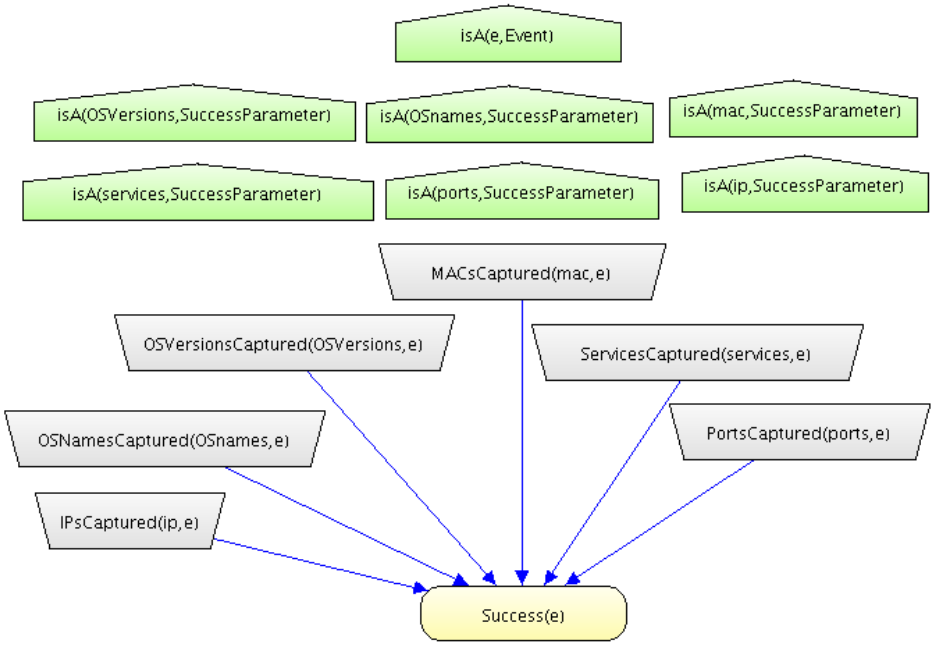


Fig. 4. The Success MFrag

uncertainty and could serve as a formal basis for representing and propagating fusion results in a reasoning system such as DICE. They expand the possibilities of standard ontologies by introducing the requirement of a proper representation of the statistical regularities and the uncertain evidence about entities in a domain of application. PR-OWL provides a means to express MEBN theories in OWL. The technique has been applied to provide HLIF in other domains, such as maritime domain awareness, which also had the requirement of capturing information from SMEs and other types of sources [21], [22]. PR-OWL ontologies can be developed using the graphical probabilistic knowledge package UnBBayes[2], an open source, Java™-based application developed at the University of Brasilia (UnB). It includes a MEBN/PR-OWL plugin developed by UnB with participation from George Mason University, in Fairfax, VA. The plugin provides both a GUI for building probabilistic ontologies and a reasoner based on the MEBN/PR-OWL framework [23], [24]. Reasoning in the UnBBayes MEBN/PR-OWL plugin involves SSBN construction, which can be seen type of propositionalization, and the subsequent inferential process over the resulting SSBN.

Figure 4 depicts a simplified version of the*Success* MFrag, which is part of the DICE MFrag Library. This MFrag captures some of the concepts and relationships that are useful to infer the level of success a specific cyber event (e.g. an asset execution event) would have based on the specific conditions of that event (not shown in the picture). The three different types of MFrag nodes can be seen: *Context*, *Input*, and *Resident* nodes.

*Resident* nodes are the random variables that form the core subject of an MFrag. The MFrag defines a local distribution for each resident node as a function of the combination of the states of its parents in the fragment graph. Resident nodes can be discrete or continuous. There is one discrete resident node in the *Success* MFrag, which is depicted as yellow rounded rectangles in the picture. Continuous resident nodes, which do not appear in the *Success* MFrag, are depicted as rounded rectangles with double lines. The distributions of both continuous and discrete resident nodes are defined in the MFrag they are resident to. In this case, the discrete resident node *Success(e)* is a boolean random variable that conveys the probability of success for event *e*.

*Input* nodes, depicted as gray trapezoids in the figure, serve as "pointers" referring to resident nodes in other MFrags. Input nodes influence the local distributions of resident, but their own distributions are defined in the MFrags in which they are resident. For instance, input node *IPsCaptured(ip, e)* is a resident node in the *IPresults* MFrag, which conveys the distribution of IPs captured in event *e*, as well as its relationship with other nodes in that MFrag.

*Context* nodes are Boolean (i.e., true/false) random variables representing conditions that must be satisfied for the probability distribution of the resident node in an MFrag to apply. The same way it happens with input nodes, context nodes have distributions defined in their respective resident MFrags.

As an example of how the representation works, suppose that for an attack event $\alpha$ all criteria are important. Figure 5 depicts the resulting SSBN for a query in which the input information defines such requirements. In this case, all the resident nodes to which the input nodes of the *Success* MFrag were pointing are now instantiated from their respective

[2]http://unbbayes.sourceforge.net

resident MFrag and built into the SSBN.

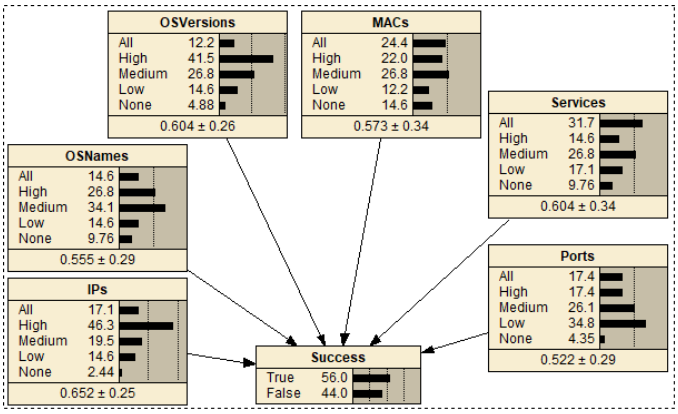


Fig. 5. The Success SSBN for event $\alpha$ with six criteria

Now, supposing that another event, $\beta$, only 3 criteria are of importance. In this case, the same MFrag would generate the SSBN in Figure 6. That is, the data entered in the probabilistic reasoner would result in a Bayesian network taylored for the conditions of each specific event (a.k.a., an SSBN).

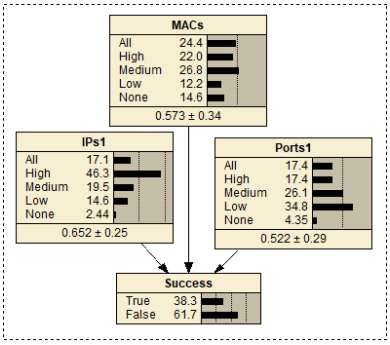


Fig. 6. The Success SSBN for event $\beta$ with three criteria

By allowing uncertainty on context nodes, MEBNs can represent several types of sophisticated uncertainty patterns, such as relational uncertainty or existence uncertainty. This proves to be very convenient in HLIF within complex domains such as estimation of impact of cyber assets, where it is not always possible to know whether a given result refers to each specific node in the network. In order to leverage the representational power of MEBN/PR-OWL, DICE employs a carefully crafted architecture that integrates SME-provided knowledge with results from experiments.

### *A. Description of the HLIF architecture*

Figure 7 depicts a diagram with the proposed architecture for HLIF. For clarity and simplicity, the diagram only shows the components directly related to the HLIF workflow. In the diagram, the first step is to define a knowledge need, which is conveyed as a self-contained set of SPARQL queries. The queries included must be in a specific format, since they have to be parsed by both the logical and probabilistic reasoners in the workflow. For instance, the system must determine whether new experiments are needed to satisfy the query set or whether experimental data already exists. In case new experiments are required, then they will be conducted and their results stored in the Experiment Results Library, which is the second step in the diagram.

Once the Experiment Results Library has all the experiments required to respond the query set, then the SSBN controller module will start the SSBN construction process, the third step in the diagram.

The SSBN construction algorithm implemented in UnBBayes involves an iterative process in which metadata from the experiments will be input to the probabilistic reasoner, and then used to define which MFrags should be instantiated. For example, supposing that the metadata of one experiment input to the reasoner includes information that the experimental network size was between 10-50 computers, then only the MFrags which have context nodes matching that size criteria would be instantiated. The difference between the bi-directional arrows linking the SSBN Controller module accounts for the distinct interaction between the module and the libraries. More specifically, based on the SPARQL query set it receives, the reasoner will request the experiments' metadata included in the queries and use it to define which MFrags must be instantiated and how many times they must be instantiated. Thus, while metadata from experiments are asked for and received once, MFrags will be visited by the reasoner and instantiated many times.

The end result of this process is the SSBN shown in step 5 of the diagram, which contains the prior knowledge for that specific configuration. That is, the probability information stored in the resident nodes of the MFrags being instantiated will dictate the CPTs of the resulting SSBN. Note that the structure of the SSBN was defined by the query set received and by the metadata of the experiments selected. The first conveys what needs to be learned, while the second defines what should be the structure of the SSBN to provide the answer. The resulting SSBN brings the "current answer" of the system based on the prior information it had, which was not necessarily obtained by a set of experiments hand-crafted for that specific query set.

The above distinction illustrates the possibility that the situation involved in a given query set was never specifically tried in experiments, so the system would instantiate MFrags that were learned in situations that were similar enough. In other words, the SSBN reflects the best knowledge available at the time.

When new experiments are performed, then the Bayesian parameter learning module is invoked and receives both the current SSBN and the results of the new experiments. It then computes the posterior probabilities through parameter learning. The current DICE system has different versions of both regular and incremental parameter learning, and the choice of which is based on system design details that are outside the scope of this paper.

The result of the parameter learning process, step 7 in the diagram, is a SSBN with the same structure of the input SSBN

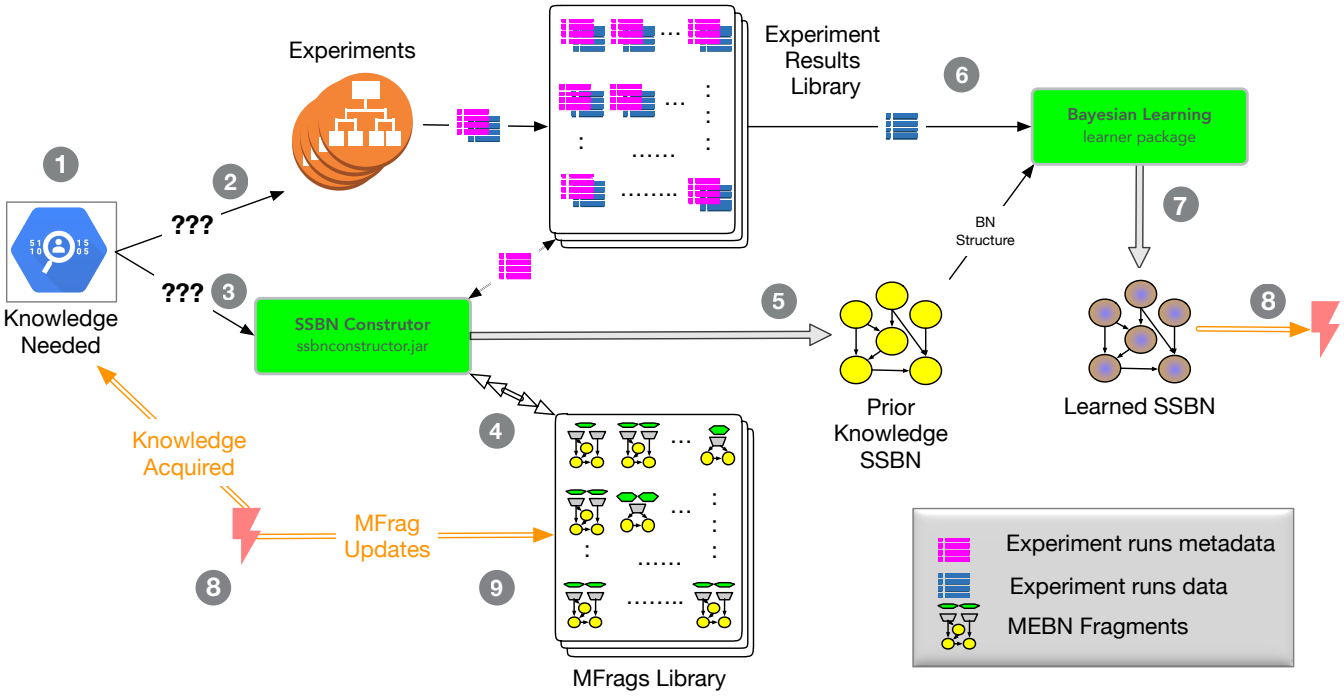


Fig. 7. The HLIF architecture fuses knowledge elicited from experiments and SMEs

but with a new joint probability distribution reflecting the learned data.

Step 8 illustrates the feedback mechanism, in which the original query set is answered and, in case new experiments were included, the MFrags that were originally instantiated are updated to reflect the new information acquired.

This architecture allows for the modeling of both experiment-based and SME generated knowledge, while implementing a consistent way of merging disparate knowledge on evaluation of cyber assets. It also leverages Bayesian learning to enhance existing knowledge based on the experimental results, as well as data collected from other sources.

## VI. Conclusions

This paper describes an approach together with prototype results for fusing multiple cyber observables to compute probabilistic attributes of cyber assets, including probability of success, detection, attribution, collateral damage, and speed. The approach achieves data fusion and alignment through a semantic I/O pipeline that achieves semantic provenance from end-results to initial data items through backwards chaining across the collection of SPARQL I/O pairs. Knowledge is built as a collection of modular knowledge fragments (MFrags) that can be dynamically instantiated to predict cyber asset effects on never-before-seen target environments. Finally, the incremental nature of knowledge acquisition integrates experimentation with knowledge extraction, flagging the need for more experiments to be executed to reduce uncertainty, and using the results from new experiments to update the set of knowledge fragments. Going forward, we intend to expand the set of derived attributes and cyber assets that are covered, and provide enhanced implementation support for incremental learning. In addition, we propose to further reduce the need for manual knowledge engineering by completely automating the MFrag assembly process and implement structure learning for MFrags.

## Acknowledgment

The authors would like to acknowledge the Air Force Research Laboratory, Rome, NY, for supporting the research described in this paper. In addition, the authors like to thank Thomas Eskridge, Evan Stoner, and Adrian Granados from Florida Institute of Technology for providing cyber experimentation support; Kasia Olejnik and Stephane Blais for developing the semantic I/O pipeline for cyber asset quantification; and John Gancasz for proof-reading versions of the paper and handling public release approvals.

## References

[1] A. N. Steinberg, C. L. Bowman, and F. E. White, "Revisions to the JDL data fusion model," in *Sensor Fusion: Architectures, Algorithms, and Applications III*, B. V. Dasarathy, Ed., vol. 3719. Orlando, FL, USA: SPIE, Mar. 1999, pp. 430–441. [Online]. Available: http://link.aip.org/link/?PSI/3719/430/1

[2] E. Blasch, . Boss, and D. Lambert, *High-Level Information Fusion Management and System Design*, 1st ed. Boston: Artech House, Apr. 2012.

[3] D. McGuinness, "Ontologies for information fusion," in *Information Fusion, 2003. Proceedings of the Sixth International Conference of*, vol. 1, 2003, pp. 650 – 657.

[4] E. G. Little and G. L. Rogova, "Designing ontologies for higher level fusion," *Information Fusion*, vol. 10, no. 1, pp. 70–82, Jan. 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1566253508000377

[5] E. Blasch, E. Dorion, P. Valin, E. Bosse, and J. Roy, "Ontology alignment in geographical hard-soft information fusion systems," in *Information Fusion (FUSION), 2010 13th Conference on*, Jul. 2010, pp. 1 –8.

[6] K. Laskey and K. Laskey, "Uncertainty reasoning for the world wide web: Report on the URW3-XG incubator group," W3C, URW3-XG, 2008. [Online]. Available: http://ite.gmu.edu/~klaskey/papers/URW3_URSW08.pdf

[7] L. Predoiu and H. Stuckenschmidt, "Probabilistic extensions of semantic web languages - a survey," in *The Semantic Web for Knowledge and Data Management: Technologies and Practices*. Idea Group Inc, 2008.

[8] P. C. G. Costa, "Bayesian semantics for the Semantic Web," PhD Dissertation, George Mason University, Fairfax, VA, USA, Jul. 2005, brazilian Air Force. [Online]. Available: http://hdl.handle.net/1920/455

[9] P. C. G. Costa and K. B. Laskey, "PR-OWL: a framework for probabilistic ontologies," in *Proceedings of the International Conference on Formal Ontology in Information Systems (FOIS 2006)*, ser. Frontiers in Artificial Intelligence and Applications, B. Bennet and F. Christiane, Eds., vol. 150. Baltimore, MD, USA: IOS Press, Nov. 2006, pp. 237–249. [Online]. Available: http://www.booksonline.iospress.nl/Content/View.aspx?piid=2210

[10] R. Pan, Z. Ding, Y. Yu, and Y. Peng, "A Bayesian network approach to ontology mapping," in *The Semantic Web ISWC 2005*, ser. Lecture Notes in Computer Science, Y. Gil, E. Motta, V. Benjamins, and M. Musen, Eds. Springer Berlin / Heidelberg, 2005, vol. 3729, pp. 563–577, 10.1007/11574620_41. [Online]. Available: http://dx.doi.org/10.1007/11574620_41

[11] R. Giugno and T. Lukasiewicz, "P-SHOP(D): A probabilistic extension of SHOQ(D) for probabilistic ontologies in the semantic web," in *Proceedings of the Eight European Conference on Logics in Artificial Intelligence*, S. Flesca, S. Greco, N. Leone, and G. Lanni, Eds., vol. 2424. Cosenza, Italy: Springer LNCS, Sep. 2002, pp. 86–97. [Online]. Available: http://cat.inist.fr/?aModele=afficheN&cpsidt=14666085

[12] D. Koller, A. Levy, and A. J. Pfeffer, "P-CLASSIC: A tractable probablistic description logic," in *Proceedings of the National Conference on Artificial Intelligence*, 1997, pp. 390–397.

[13] H. Nottelmann and N. Fuhr, "Adding probabilities and rules to owl lite subsets based on probabilistic datalog," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 14, pp. 17–42, 2006.

[14] K. Zaffarano, J. Taylor, and S. Hamilton, "A Quantitative Framework for Moving Target Defense Effectiveness Evaluation," in *Proceedings of the Second ACM Workshop on Moving Target Defense*, ser. MTD '15. New York, NY, USA: ACM, 2015, pp. 3–10. [Online]. Available: http://doi.acm.org/10.1145/2808475.2808476

[15] M. Atighetchi, B. I. Simidchieva, F. Yaman, T. Eskridge, M. Carvalho, and N. Paltzer, "Using Ontologies to Quantify Attack Surfaces," in *Semantic Technology For Intelligence, Defense, and Security (STIDS)*, Fairfax, VA, Nov. 2016.

[16] M. Atighetchi, B. Simidchieva, N. Soule, F. Yaman, J. Loyall, D. Last, D. Myers, and C. B. Flatley, "Automatic Quantification and Minimization of Attack Surfaces," in *27th Annual IEEE Software Technology Conference (STC)*, Long Beach, CA, Oct. 2015.

[17] T. Lebo, N. Del Rio, P. Fisher, and C. Salisbury, "A five-star rating scheme to assess application seamlessness," *Semantic Web*, vol. 8, no. 1, pp. 43–63, 2017.

[18] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, 1st ed. Morgan Kaufmann, Sep. 1988.

[19] K. B. Laskey, "MEBN: a language for first-order Bayesian knowledge bases," *Artificial Intelligence*, vol. 172, no. 2-3, pp. 140–178, Feb. 2008. [Online]. Available: http://portal.acm.org/citation.cfm?id=1327646

[20] P. C. G. Costa, K. Chang, K. B. Laskey, and R. N. Carvalho, "High level fusion and predictive situational awareness with probabilistic ontologies," in *Proceedings of the Thirteenth International Conference of Information Fusion (FUSION 2010).*, George Mason University, Fairfax, VA, USA, May 2010. [Online]. Available: http://c4i.gmu.edu/events/reviews/2010/abstracts_bios.php#p3A

[21] P. C. G. Costa, K. B. Laskey, and K. Chang, "PROGNOS: applying Probabilistic Ontologies to distributed predictive situation assessment in naval operations," in *Proceedings of the Fourteenth International Command and Control Research and Technology Conference*. Washington, DC, USA: CCRP Publications, Jun. 2009, best paper award of the Collaborative Technologies for Network-Centric Operations Track. [Online]. Available: http://www.dodccrp.org/events/14th_iccrts_2009/papers/108.pdf

[22] R. N. Carvalho, P. C. G. Costa, K. B. Laskey, and K. Chang, "PROGNOS: predictive situational awareness with probabilistic ontologies," in *Proceedings of the Thirteenth International Conference of Information Fusion (FUSION 2010).*, Edinburgh, UK, Jul. 2010.

[23] R. N. Carvalho, L. L. Santos, M. Ladeira, and P. C. G. Costa, "A GUI tool for plausible reasoning in the semantic web using MEBN," in *Seventh International Conference on Intelligent Systems Design and Applications(ISDA)*. Los Alamitos, CA, USA: IEEE Computer Society, Oct. 2007, pp. 381–386.

[24] P. Costa, M. Ladeira, R. N. Carvalho, K. Laskey, L. Santos, and S. Matsumoto, "A first-order Bayesian tool for probabilistic ontologies," in *Proceedings of the Twenty-First International Florida Artificial Intelligence Research Society Conference (FLAIRS 2008)*. Coconut Grove, FL, USA: AAAI Press, May 2008, pp. 631–636.