STIDS 2016

The Eleventh International Conference on
Semantic Technology for Intelligence, Defense, and Security

Semantics in the Internet of Things

November 14-17, 2016

George Mason University
Fairfax, Virginia Campus

Conference Proceedings

Kathryn B. Laskey, Ian D. Emmons, Paulo C. G. Costa, Alessandro Oltramari, Eds.

GEORGE MASON UNIVERSITY | C4I CENTER

Preface

The Eleventh International Conference on Semantic Technology for Intelligence, Defense, and Security (STIDS 2016) provides a forum for academia, government, and industry to share the latest research on semantic technology for defense, intelligence, and security applications. Semantic technology is a fundamental enabler to achieve greater flexibility, precision, timeliness, and automation of analysis and response to rapidly evolving threats. The STIDS 2016 theme is Semantics in the Internet of Things. In addition, topics of general interest for STIDS include:

- Best practices in the engineering of ontologies
- Collaboration
- Command and Control (C2) and Situation Awareness (SA)
- Cyberspace: defense, exploitation, and counter-attack
- Decision making
- Economics and financial analysis
- Emergency response
- Human factors and usability issues related to semantic technologies
- Information sharing
- Infrastructure protection
- Intelligence collection, analysis, and dissemination
- Law and law enforcement
- Planning: representation of and reasoning over plans and processes
- Predictive analysis
- Provenance, source credibility, and evidential pedigree
- Resiliency, risk analysis, and vulnerability assessment
- Science and technology (biology, health, chemistry, engineering, etc.)
- Sensor systems
- Sociology (social networks, ethnicity, religion, culture, politics, etc.)
- Spatial and temporal phenomena and reasoning
- Uncertainty as it relates to ontologies and reasoning

Fairfax, VA
November 2016

Ian Emmons and Kathryn Laskey
STIDS 2016 Technical Chairs

Paulo Costa and Alessandro Oltramari
STIDS 2016 General Chairs

STIDS 2015 Committees

Program Committee

Carl Andersen	Raytheon BBN Technologies
Rommel Novaes Carvalho	Brazil's Office of the Comptroller General
Erik Blasch	AFRL
Paulo Costa	George Mason University
Timothy Darr	Knowledge Based Systems Inc.
Ian Emmons	Raytheon BBN Technologies
Matthew Fisher	Progeny Systems
Mark Greaves	Pacific Northwest National Laboratory
Richard Haberlin	EMSolutions, Inc.
Peter Haddawy	Mahidol University
Brian Haugh	IDA
Edward Huang	George Mason University
Gregory Joiner	Raytheon BBN Technologies
Anne-Laure Jousselme	NATO Centre for Maritime Research and Experimentation (CMRE)
Mieczyslaw Kokar	Northeastern University
Dave Kolas	Raytheon BBN Technologies
Kathryn Laskey	George Mason University
Louise Leenen	CSIR
Richard Markeloff	Raytheon BBN Technologies
David Mireles	Raytheon BBN Technologies
Ranjeev Mittu	US Naval Research Laboratory
Leo Obrst	MITRE Corporation
Alessandro Oltramari	Bosch Research and Technology Center
Patrice Seyed	Rensselaer Polytechnic Institute
Barry Smith	University at Buffalo
Andrew Perez-Lopez	BBN Technologies
Tony Stein	Raytheon BBN Technologies
Kathleen Stewart	University of Maryland
Gheorghe Tecuci	George Mason University
Brian Ulicny	Thomson Reuters
Amanda Vizedom	Criticollab, LLC
Andrea Westerinen	Nine Points Solutions, LLC
Duminda Wijesekera	George Mason University
Abbas Zaidi	George Mason University

STIDS Steering Committee

Paulo Costa	George Mason University
Ian Emmons	Raytheon BBN Technologies
Katherine Goodier	Xcelerate Solutions
Kathryn Laskey	George Mason University
Leo Obrst	MITRE Corporation
Barry Smith	NCOR, University at Buffalo

STIDS 2015 Organizing Committee

General Chairs

Paulo Costa

Alessandro Oltramari

Technical Chairs

Ian Emmons

Kathryn Laskey

Publicity Chair

Richard Markeloff

Local Arrangements Chair

André Negrão Costa

Classified Session Chair

Brian Haugh

Local Team (GMU)

Debra Schenaker (Administrative Chair)

Priscilla McAndrews

Nicholas Clark

Shou Matsumoto

Jeronymo Carvalho

Michael Dean Best Paper Award

August 7, 1961 - November 19, 2014

The Michael Dean Best Paper Award was established in 2014 in recognition of Michael Dean's many and diverse contributions to the STIDS community. In selecting the winner, the committee sought to highlight the qualities that made Mike such an asset to this community. The criteria for selection exemplify the very best contributions to the conference and the community. To this end, the Michael Dean Best paper is the one that, in the judgment of the award committee, best satisfies the following criteria:

1. Conveys a clear, careful understanding of the problem or issue being addressed, and clearly states why it matters.
2. Conveys a thorough understanding of technical issues, and a well-grounded, pragmatic view of prior and related work.
3. Clearly identifies the specific semantic technologies being discussed, and their relationship to the problem.
4. Identifies specific experience or expertise on which the paper and its conclusions draw.
5. If a semantic system or application is being presented as part of a solution, clearly identifies and communicates the components of this system, including any ontologies, and how they interact, as well as their degree of actuality, availability, maturity and source.
6. Identifies whether and how such system/application/components have been evaluated and with what results.
7. Identifies outcomes, experiences, and lessons learned.
8. Demonstrates prioritization of greater technical and domain understanding and problem-solving over self-promotion, organizational promotion, partisan or programmatic scorekeeping, or other, narrower concerns.
9. Demonstrates knowledge of prior and current art, strengthens such knowledge in the community, and promotes better understanding by sharing the rationale for choices, especially when they diverge from common practice.
10. Demonstrates and strengthens the state of the art of semantic technology via the quality of the work described. Provides promising ways forward while negotiating known trade-offs and avoiding known pitfalls. Helps more junior technologists avoid repetition of old errors, and provides more senior technologists with new insights.

The winning paper was announced on the last day of the conference:

- *2016 Michael Dean Best Paper:* Michael Reep, Bo Yu, Duminda Wijesekera, Paulo Costa. Sharing Data under Genetic Privacy Laws.
- *Runner-up:* Frank Greitzer, Muhammad Imran, Justin Purl, Elise Axelrad, Yung Mei, Leong, D. E., Sunny Becker, Kathryn Laskey, Paul Sticha. Developing an Ontology for Individual and Organizational Sociotechnical Indicators of Insider Threat Risk.

2015 Michael Dean Award Committee

Leo Obrst (chair)	MITRE Corporation
Mark Underwood	Krypton Brothers LLC
Ian Emmons	Raytheon BBN Technologies
Richard Markeloff	Raytheon BBN Technologies

Table of Contents

STIDS 2016 Invited Talk: A Whistle-stop Tour of Ontology-based Solutions to Improve Situational Awareness for a Dull, Dirty, Diverse IoT

Dr. Leo Obrst and Dr. Mark Underwood

Abstract

The Internet of Things is already an awkward, overly confident adolescent learning to get along with the rest of computing society. When a 2005 paper asking for "Sympathy for the Sensor Network Debugger" was presented (Ramanathan et al., 2005), Juniper Network's recent estimate of 38 billion connected things would have seemed farfetched. IoT is not new. The old: closed real time sensor-edge networks, distributed computing, loosely coupled heterogeneous networks, bridge-building across disparate domains, software that reasons about real world events. But a lot is new. Streaming data platforms (Paypal, Prometheus, etc), software-defined networks, ready access to cloud-based reasoning APIs. In this talk we take a whistle-stop tour of use cases around a single family of sensors: the smart humidistat (e.g, Ecobee Si, Honeywell VisionPro and FocusPro). We identify the issues raised by the use case family and suggest aspects of those issues best addressed by ontology-based solutions. We conclude by citing opportunities to integrate ontologies into model-based engineering approaches, highlighting lessons from the 2015 Ontology Summit (Underwood et al., 2015). Domain models are needed for chillers, data centers, refrigeration units, etc. Cross-domain models are needed to "import" external events such as weather or temperature, and to reason about time and duration in terms that make sense for a local context. Manufacturer-supported (or, for widely-adopted consumer products, crowdsourced) models are needed to address subtle, implementation-specific management of battery life, calibration, duty cycle, performance range and maintenance best practices. At the whispered-about edge of the conversation about IoT, ontologies can be integrated with simulation, test and resilience exercises. IoT likely piles a layer of added complexity over nontrivial enterprise applications. Existing software design life cycle (SDLC) practices aren't much help.

Biography: Dr. Leo Obrst

Dr. Leo Obrst is Chief Scientist for Cognitive Science and Artificial Intelligence in the CogSci & AI department of MITRE's (www.mitre.org) Center for Connected Government (CCG), where he created and led, but now advises the Information Semantics Group (semantics, ontological engineering, knowledge representation and reasoning). He has been involved in projects on Semantic Web rule/ontology interaction, automated reasoning, context-based semantic interoperability, ontology-based knowledge management, conceptual/semantic search and information retrieval, metadata and taxonomy/thesaurus construction for community knowledge sharing, intelligent agent technology, semantic support for natural language processing, and ontology-based modeling of complex decision-making for situational awareness, command and control, cyberspace, information integration and analysis, intelligence and event analysis/prediction. His most recent research is as chief ontologist and chief computer scientist for a US Veteran's Health Administration project on next-generation semantic health care records, and patient-

centered clinical support, 2014-present. In 1999-2001, he was director of ontological engineering at Vertical-Net.com, a department he formed to create ontologies in the product and service space to support Business-to-Business e-commerce. Leo has worked over 32 years in computational linguistics, knowledge representation, and in the past 21 years in ontological engineering and more recently (since 2001) in Semantic Web technologies. Leo is co-author (with Mike Daconta and Kevin Smith) of the book "The Semantic Web: The Future of XML, Web Services," and Knowledge Management, John Wiley, Inc., June, 2003; co-editor (with Terry Janssen and Werner Ceusters) of the book "Ontologies and Semantic Technologies for Intelligence," IOS Press, August, 2010; and has published many book chapters, conference and workshop papers (over 70 refereed papers) and many reviews. He has organized or been a program committee member on more than 75 conferences/workshops, including Formal Ontology in Information Systems (FOIS), Ontologies for the Intelligence Community (OIC), the Association for the Advanced of Artificial Intelligence (AAAI), and the International Semantic Web Conference (ISWC). He is a Senior Member of AAAI, and a long term member of ACM and LSA.

Biography: Dr. Mark Underwood

Dr. Mark Underwood is the CEO and co-founder of Krypton Brothers LLC, a consultancy specializing in Big Data security, rapid intranet exploitation, digital forensics, software quality and domain-specific frameworks.

Underwood has served as lead engineer or principal investigator on artificial intelligence projects for DARPA and for Army and Air Force research laboratories. Most recently, he is working with standards organizations to foster information assurance and provenance transparency. Underwood is co-chair of the NIST Big Data Public Working Group's security and privacy subgroup, and was co-chair of the 2015 Ontology Summit focused on the Internet of Things. In 2014, he served on the workshop committee for the IEEE Big Data Conference and moderated several panels. He is a NIST Guest Researcher and currently serves on the ISO/IEC JTC1 Working Group WG9 on Big Data.

Other standards and related work: Underwood is a member of the IEEE P 1915.1 Security for Virtualized Environments WG, working on a standard for SDN and NFV security. He is an ASQ Certified Software Quality Engineer. He has participated in reviews of audit practices in the HL7 PASS Healthcare Audit Services ad hoc committee and the 2016 version of the OMG Cloud Standards Customer Council 's Security Standards Whitepaper. In the forthcoming "White paper on Semantic Interoperability for the Web of Things" produced under the aegis of IEEE P2413, he drafted the section on API-first and microservices.

Underwood is an advocate for patient-managed health information, including access to automated decision support systems. He is a professional writer whose emphasis is technology and health topics. In recent years, articles have appeared in The Daily Beast, CBS TechRepublic, and Drugstore News for sponsors that range from GE and the Annenberg Center for Health Sciences to Time Warner Cable. He has written recurring columns for Syncsort, Ipswitch and ADP.

Underwood also plays electric violin and edits the sites PoetryandScience.com and BigDataStandards.com.

Recent publications

Underwood M. Intranet Exploitation of Social Network Knowledge Intelligence. In: Chugh R, ed. *Harnessing Social Media As a Knowledge Management Tool (Advances in Knowledge Acquisition, Transfer, and Management*. Hershey, PA: IGI Global; 2016.

Underwood M. Big Data Complex Event Processing for Internet of Things Provenance: Benefits for Audit, Forensics and Safety. In: Brooks T, ed. *Cyber-Assurance for the Internet of Things*. Hoboken NJ: Wiley; 2016.

STIDS 2016 Invited Talk:
Importance of Semantic Ontologies in Information Fusion

Dr. Erik Blasch

Abstract

The use of semantic technologies has essential implications for information fusion systems solutions. An emerging development in high-level information fusion (HLIF) is the importance of the user for mission management, command and control, as well as process refinement. The ability of the user to be part of the systems solution supports low-level information fusion (LLIF) functions of object, situation, and impact assessment. Future technology designs will require coordinating the LLIF physics-based big data measurements with the HLIF human-derived information content. A semantic ontology is necessary for physics-based and human-derived information fusion (PHIF). The fusion of measurements and content should augment contextual understanding, refine uncertainty estimates, and provide robust decision support. This talk will provide trends in high-level information fusion, address developments in an uncertainty ontology, and provide examples of PHIF. Examples include unmanned aerial vehicle (UAV), multi-intelligence, and space situation awareness.

Biography: Dr. Erik Blasch

Dr. Erik Blasch is a principal scientist at the the United States Air Force Research Laboratory (AFRL) in the Information Directorate at Rome, NY, USA. From 2009-2012, he was an exchange scientist to Defence Research and Development Canada (DRDC) at Valcartier, Quebec. From 2000-2009, Dr. Blasch was the Information Fusion Evaluation Tech Lead for the AFRL Sensors Directorate - COMprehensive Performance Assessment of Sensor Exploitation (COMPASE) Center supporting design evaluations in Dayton, OH. Dr. Blasch has been an Adjunct Electrical Engineering Professor at Wright State University teaching signal processing, target tracking, and information fusion.

He is a member of the International Society of Information Fusion (ISIF) Evaluation of Technologies for Uncertainty Reasoning Working Group (ETURWG) Automatic Target Recognition Working Group (ATRWG), and the Dynamic Data Driven Applications System (DDDAS) community. He served as a member of IEEE Aerospace and Electronics Systems Society (AESS) Board of Governors (BoG), was a founding member of the International Society of Information Fusion (ISIF) (www.isif.org), and the 2007 ISIF President. Recently, he was the Chairman for the AIAA/IEEE AESS Digital Avionics Systems Confernce focusing on the future for Unmanned Aerical Vehicle (UAV) traffic management (UTM).

He has focused on information fusion, target tracking, pattern recognition, and robitcs research compiling 600+ scientific papers and book chapters. He holds 10 patents, presented over 30 tutorials, and is an associate editor of three academic journals. He was a recipent of the Military Sensing Symposium Ledaership in Data Fusion Award, Fellow of SPIE, Associate Fellow of AIAA, and a senior member of IEEE.

Technical Papers

Scalable Semantically Driven Decision Trees for Crime Data

Shawn Johnson, George Karabatis
Department of Information Systems
University of Maryland, Baltimore County (UMBC),
1000 Hilltop Circle, Baltimore, MD 21250, USA
{yv74924, GeorgeK}@umbc.edu

Abstract. **When dealing with large volumes of data in organizations, there is always a need to associate data with its appropriate meaning, since the same data object may have different meaning to different users. This creates a problem of delivering search results that is different from a requester's intended purpose. To solve this problem, we propose a parallelizable framework capable of capturing user specified constraints that are both semantically relevant to a search/domain in question as well as contextually relevant to a user and/or organization.**

I. INTRODUCTION

When attempting to find data that is relevant to a user and/or organization based on a query, it is very common to retrieve exact matches in response to a search. While using the exact matching of strings as user search keywords can retrieve exact results, a user may be looking for data with a specific meaning based on his or her intended search preferences but the data that is provided by a system or organization may have a completely different meaning. This problem is further compounded by having to ensure that data that has recently been ingested into a system is consistent with data that currently resides on the same system. This can create an intractable problem for a user such as having to constantly poll and classify new data or accept new data that may be inappropriately classified to ensure the semantic meaning of the data is consistent. In addition, the problem of new data being added to a system on a massive scale necessitates a scalable solution. We propose an approach which allows users to personalize search terms according to the same set of concepts where search results can be universally understood by the same community of users according to personalizable search profiles. Our approach also enhances the accuracy of search results by returning semantically similar results from a specific domain. The impact of our approach dramatically enhances the ability of users to personalize a search thus retrieve more accurate results. With the introduction of our framework, we make a few key contributions:

1) We allow user specified search preferences to be expressed with robust semantics resulting in higher precision and recall that closely match the user's intended search terms.

2) We have developed a method for user specified semantics to be expressed probabilistically in the search. Search results that have a semantic similarity to a set of user specified preferences are also returned enabling us to handle uncertainty in our approach as well.

3) We have developed a way for multiple sets of user specified preferences to be expressed using the same ontology.

4) Our methodology allows for robust semantics to be expressed in a parallelizable way.

5) We have implemented a prototype and evaluated our methodology by conducting experiments using precision and recall as metrics.

Motivating Example: Many municipalities often have a need to capture semantically relevant data for an intended purpose. For example, Bob, a detective with the city of Baltimore, needs to get the current statistics of all aggravated theft incidents between 1964 and 2013, in Baltimore, Maryland, USA. Bob wants to compare this data with the same data on a national scale over the same time period. Furthermore, there are different types of data on thefts that Bob wants to compare against. To further compound the issue, different vendors provide different labels for the same type. The impact of capturing more semantically relevant data and getting more accurate results enables law enforcement officials to make better informed decisions because the information they are looking for is more precise and relevant to a specific situation for which the officials need to make a decision about.

We describe our approach in section II, and then we continue on with a discussion of our experiments

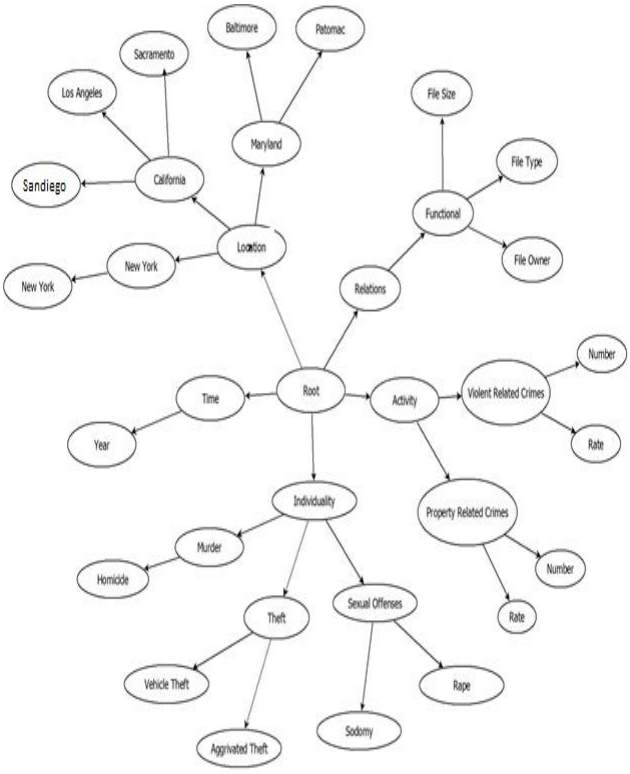
in section III, a validation of our approach in IV, and then a discussion of relevant work in section V. We then finish up our work with concluding thoughts and proposed future work in section VI.

II. Approach

To begin our learning, we make use of a single ontology that represents our categories of context, with each category of context represented via separate branches of the ontology.

Continuing our motivating example, we show a sample representation of our ontology using figure 1 to show the taxonomical tree of terms that a user can select as user preferences and how they organized. Utilizing the Operational of Context [1] we model our ontology using 5 separate categories of context.

Fig. 1 Sample representation of an ontology:



Operational Definition of Context [1], we model our ontology using 5 separate categories of context. They are:

1. Individuality Context - The individuality context are attributes that describe an entity's type such as specifying the type of theft

2. Time Context - The time context is anything that describes any kind of temporal attributes related to an entity such as a year

3. Location Context - The location context describes any kind of location attribute related to an entity such as a city or state

4. Activity Context - The activity context describes a goal related to entity such as how much of something a law enforcement official may be looking for such as a total amount of violent crime

5. Relations Context - The relations context are attributes of an entity that describe an entity's relation to another entity or its parts

User preferences in our ontology are saved with literals that are added to each node a user has selected. We chose not to add countries in our ontology because we are assuming that our domain is within all 50 states within the USA. Bob the detective stores his preferences in a user profile that matches parts of the ontology (in figure 1) such as all aggravated thefts that have occurred in Baltimore, Maryland, between 1964 and 2013. A special depth first search algorithm then searches the ontology and matches saved user specified literals for each node within each branch of the ontology and creates a special in memory tree model. This in memory tree model is a sub-tree of the ontology that matches all of the user selected preferences that were found in the original ontology. Referring again to figure 1, only parts of the ontology that match aggravated theft, Baltimore, Maryland that are between 1964 and 2013 are copied into the new in memory tree model.

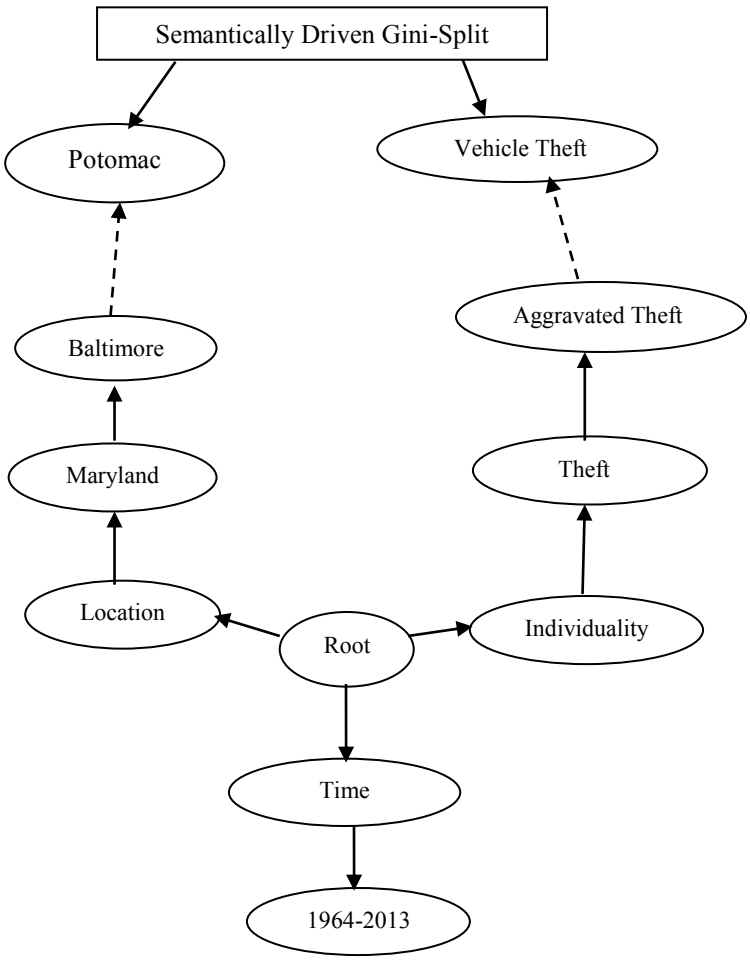
After the original ontology has been parsed and the in memory tree model has been implemented, we are now able to build the rest of our decision tree. Records (files) are initially classified based on user preferences using the in memory tree model provided in figure 2. For example, all records that are classified must include records that have the attributes Baltimore, Maryland, aggravated theft, or between the years 1964-2013 (meeting Bob's specified user preferences). Now that only instances remain that match the user specified constraints above, (i.e., the records are all containing values for the attributes Baltimore, Maryland, aggravated theft, or in the range 1964-2013) we must build out the remaining part of a tree based on the number of user specified splits or criteria to enable the system to return search results with increasing levels of precision by dividing the records into further subsets as specified by the user

with each additional split. For example, if the user has specified that he or she wishes to have the ontology be 5 hops deep and the ontology is only 3 hops deep, additional splits must be completed (if there are enough records left to split against in order to meet the user's specified preferences). Referring to figure 2, additional splits are made using the lowest level collection of leaves for the individuality and location branches of the ontology (additional splits were not made to time branch of the ontology because the user chose not to do so since year was the most granular measure of time specified in the data). Because the user has selected the decision tree to be 4 hops deep, additional splits are made using the lowest collection of leaves for each branch of the ontology. In the location ontology, the lowest collection of leaves in that branch of the ontology are Baltimore, Potomac, Los Angeles, Sacramento, Albany, and New York City. Because Potomac has the lowest impurity (see Semantically Driven Gini Algorithm) computed from the lowest collection of leaves, it is selected as the first candidate split using the Semantically Driven Gini-Index Algorithm. No more splits are performed on the location branch of the ontology because it is now 4 hops deep. On the individuality branch of the ontology, a single split is made using The Semantically Driven Gini-Index Algorithm, with vehicle theft being selected because it had the lowest impurity (see Semantically Drive Gini Algorithm) of the lowest leaves. No more splits are made for the individuality branch of the ontology because it is now 4 hops deep as well. Our special in memory tree model/user driven ontology looks (logically) like Figure 2.

Any documents that match any of the user preferences or Semantically Driven Gini-Index Algorithm are then tagged using a custom document summarization algorithm. The tagged files are then copied to The Hadoop Distributed File System (HDFS). Separate mapper calls are then kicked off using search conditions that meet the user specified preferences. A single reducer job is kicked off consolidating a set of files that meet a final set of preferences as specified by the user in the in-memory tree model. A final set of key value pairs of files using the key as the name of the matching file and a value of 1 that matches all of the user specified preferences are emitted signifying that the algorithm is complete. For example, only the names of files that meet all the user preferences Baltimore, aggravated theft, and between the time range of 1964-2013 are emitted.

In Memory Tree Building Algorithm: To utilize only paths of the ontology that the user has selected as his or her search preferences, a smaller graph of the in-memory tree model is created copying each of the paths that a user has selected. First, the user specifies the number of hops deep (or total depth) he or she wishes for the size of the decision tree to be. Second, the user picks from a list of selected labels that have already been mapped via our ontology via a tree like drop down list or via a search box. (The mappings are a graphical representation of the ontology, where the user can select any one of the nodes of the ontology as a user preference). After our search is complete we populate a list of labels from our ontology and copy the names of the nodes (and their parents) that the user has selected. The same set of user preferences will be copied from all five categories of the ontology below (with each branch of the ontology being stored as a separate but parallel part of our in memory tree model). The in memory tree model persists in memory as a service. Files are matched against the user preferences that have been stored in the in-memory tree model.

Fig. 2. A logical representation of an in memory tree model (including splits from the Semantically Driven Gini-Index Algorithm):



We implement our depth first search algorithm using the following representation: C for Category Contextual Model Data, O for Ontology, T_x for each branch of the ontology (each primary branch of the ontology connected to the root), where p_{abcd} represents a current node, c_{abcd} represents a child node, a indicates the current level (depth) of the ontology, b where a holds a 1 or 0 (1 if for each node selected by user or 0 if not selected by the user holds a pointer to the child node, and c is a list of all other user specified preferences, and d is a list of pointers to all child nodes of the current (parent) nodes.
In memory tree algorithm pseudocode:

Input: Ontology O
Output: in-memory tree model

LOOP: Repeat the following steps (for a + 1 of the current node until a = the lowest leaf in the tree), for T_1 - T_5, until all of O or the entire ontology is finished

1) For each category of context from Root we represent our Ontology as follows: O = $\{T_1, T_2, T_3, T_4, T_5\}$. For each branch of the tree T_x=C_{abcd} and C is the starting node for each tree (i.e., the starting node after root is Location, Time, Individuality, Activity, and Relations for each of the 5 categories of context).

2) For a of p_{abcd} of T_x add c_{abcd} to c of $p_{x\ bcd}$ where p_{abcd} is a current node, C_{abcd} is the child node of p_{abcd}, a is the current depth of T_x, b holds a 1 or 0 (1 if a user has chosen a user preference or 0 if the node has not been selected as a user preference), c holds a list of all other user specified preferences and d is a list of pointers from p_{abcd} to children nodes C_{abcd}

3) If no children exist for d, d = NULL.

END LOOP;

Semantically Driven Gini-Index Algorithm: Once the in-memory tree model building algorithm is complete, our custom Semantically Driven Gini-Index Algorithm is kicked off. First, we define our Semantically Driven Gini-Index Algorithm which calculates the impurity of each node as follows:

$$GINI(t) = 1 - \sum_j [p(j \mid t)]^2 \qquad (1)$$

Input: Lowest collection of leaves for each branch of the in-memory model

Output: Additional children nodes for each branch of the in-memory model

Where t is the node, j is the class, and p is probability of class j given a node t. The algorithm works as follows:

1) Initial splits are made based on classes specified in the in memory tree model. For example, aggravated theft is a candidate for a decision tree split since aggravated theft is a class in the tree (originally specified within the ontology).

2) Additional splits at the next level of the tree are implemented using the same collection of leaves (i.e., the lowest collection of leaves for a given branch of the ontology stored in the memory tree model) with the next split being the node with next lowest impurity using the calculation defined in (1).

3) Additional splits are induced using the same lowest collection of leaves (for each branch of the tree) until the max number of splits has been reached either by the following:

- A program driven default - This condition happens when a program driven default number of splits is reached for a given branch of the ontology. For example, if the program default is set to 5 splits, then the decision tree will not split beyond 5 hops deep for that given branch. This is true regardless of whether the split was based on a modeling part of the ontology or a part of The Semantically Driven Gini-Index Algorithm used to calculate a split.

- A user specified limit is reached - This condition happens when the user specifies a max number of splits he or she sets for a given branch of a decision tree. For example, if the max number of splits that is specified is 7 for a particular branch of the decision tree then the decision tree will stop inducing additional splits in that branch of the decision tree beyond that number regardless of whether or not the nodes being split come from the ontology or

additional splits are determined by The Semantically Driven Gini-Index Algorithm.

- The max number of possible splits has been reached - This condition occurs when all possible splits from within an ontology as well as all of the lowest level of leaves have been utilized in a split resulting in a max number of splits that can be used to build a given decision tree. For example, if the ontology is 4 hops deep and the lowest level of leaves totals 4 leaves as well, this makes the max number of splits possible for the decision tree to be 8 split

III. EXPERIMENTS

To validate our approach, we took roughly 100,000 files from the UCR Data Repository and processed them against 60 user specified preferences stored within the in memory tree model. The semantically mapped features were saved as tags in a modified version of each file, making matching for each set of user preferences a matter of matching the tags that have been specified by the user saved in the in memory tree model. Finally, the resulting MapReduce Jobs generated a file name with a value of 1 for each set of semantically mapped preferences that were matched. We used the following sample Scenario 1: Bob is looking for all crimes that occurred within the state of Maryland, between 1998 and 2002, he is searching for crime totals for larceny theft in which the files are saved as .xls files. Scenarios 2 through 10 are variants of scenario 1, where semantically mapped preferences were matched against the same files from the UCR Data Repository Values for scenarios 2-10 were chosen at random.

We have run three sets of experiments:

1) No Context and No Ontologies - Experiments with user preferences as exact search terms on MapReduce Jobs.

2) Ontologies but No Context - Experiments with files that were tagged using ontologies in RDF/OWL Files saved in the in memory model, but without any user preferences saved within them (copying the entire ontology for each category of context). The tagged files were then processed in a MapReduce Job producing the results.

3) Ontologies and Context - These experiments were executed using the files that were tagged using both the ontologies modeled in RDF/OWL Files, but also with the saved user preferences that were parsed from the RDF/OWL Files as well. The tagged files were then processed using MapReduce producing the results.

IV. VALIDATION

We use two well-known metrics to validate our approach: Precision and Recall. Recall is defined as the fraction of the records retrieved that are relevant to the query. In other words, recall reveals the percentage of the retrieved and relevant records that are relevant (whether in the answer set or outside the answer set). Precision is defined as the fraction of the retrieved records that are relevant to the search. In other words, precision reveals the percentage of retrieved and relevant records in the answer set. Recall and precision are calculated as follows:

$$recall = \frac{|\{relevant\ records\} \cap \{retrieved\ records\}|}{|\{relevant\ records\}|} \quad (2)$$

$$precision = \frac{|\{relevant\ records\} \cap \{retrieved\ records\}|}{|\{retrieved\ records\}|} \quad (3)$$

Fig. 3. Experiment results using recall

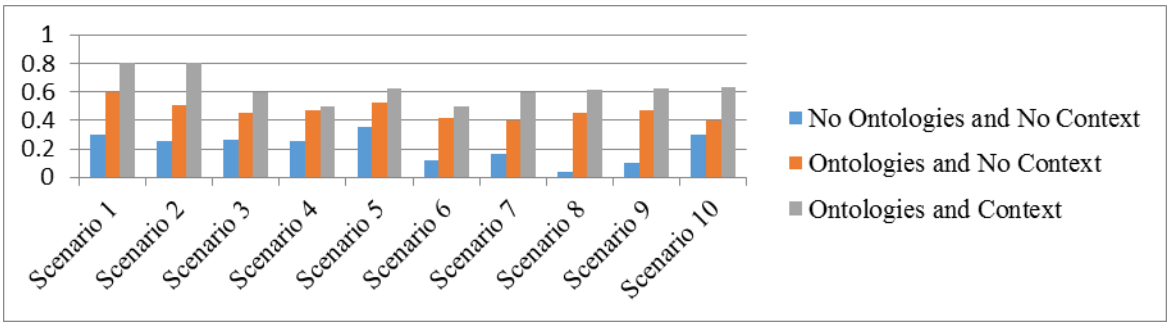
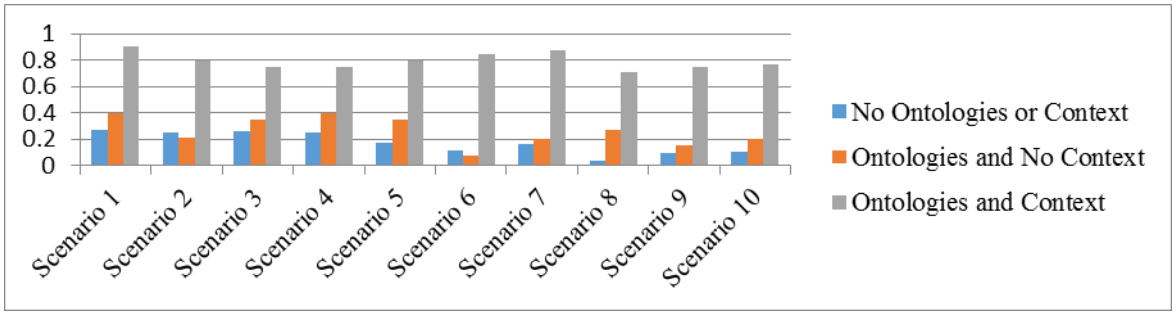


Fig. 4. Experiment results using precision



In the above scenarios or sets of test conditions that simulated using sets of user specified preferences (like in Scenario 1 discussed above), the recall was computed for no ontologies and no context by dividing matching exacting search terms from a user's search against the content of the files that are being parsed. The recall was extremely low for no ontologies and no context because the exact key words used in MapReduce Jobs matched a very small percentage of the files that were relevant to a user's search. The mean average of all 10 scenarios was 28%. The recall for ontologies and no context was computed by taking the matching terms in the in memory tree model that were parsed from the ontology and matching them against properties and content in the files being parsed. The searches for ontologies and no context yielded a higher recall because all the files that were tagged using the in memory tree model matched 1 or more of the classes specified in the ontology with a total of 47% recall, resulting in almost a 20% increase from no ontologies and no context. The recall for ontologies and context was computed by taking the matching preferences stored in the in memory tree model and matching them against properties and content of the files being parsed. Searches involving ontologies and context had a very high recall because all results retrieved matched both the semantically driven preferences that were saved in the in memory tree model as well as with the semantically specified constraints from the ontology resulting in 63% recall, a 16% increase from ontologies and no context and a roughly 35% increase improvement in recall vs. no ontologies and no context.

Precision was computed for no ontologies and no context by dividing matching exacting search terms from a user's search against the content of the files that are being parsed. The precision for no ontologies and no context was extremely low because the overwhelming majority of the search results that were returned did not match the intended user search preferences because exact key words were used for each search resulting in a 17% precision. The precision for ontologies and no context was computed by taking any of the matching terms in the in memory tree that were parsed from ontology and stored in the in memory tree model without any user specified preferences and matching them against the properties and content of the files that are being parsed. Ontologies and no context resulted in a low precision as well because the entire ontology was stored in each in memory tree model resulting in tagged files that only partially or did not match a user's intended search terms at 26% precision. The precision for ontologies and context was computed by taking any of the matching preferences stored in the in memory tree model that parsed from the ontology and selected by the user from the ontology and matching them against properties and content and of the files being parsed. Ontologies and context resulted in a very high precision versus no ontologies and no context and ontologies and no context, because documents that returned key value pairs in our MapReduce Jobs matched the semantically mapped user preferences as well the semantic constraints specified in the ontology resulting in an 80% precision; a 54% increase in ontologies and no context and a 63% increase in accuracy vs. no ontologies and no context.

In summary we learned that enabling a user to pick semantically enriched preferences from an ontology of terms that reflect an existing domain from a corpus can lead to a much higher precision and recall than using exact search terms or just using semantically enriched search terms parsed from an ontology. By building an in memory tree model we are able to both represent the knowledge representative of a domain and we also enable personalization by a user of that knowledge as well. In order to enable robust personalization, semantics must first be reflected in a model before a user can pick them. This is depicted in our results by showing that the precision and recall is higher than with just choosing exact search terms and the precision and recall further improves when allowing a user to pick attributes he or she wishes to use in a search with terms picked from the ontology.

V. RELATED WORK

Zhang et al. [2] formulated an approach that calculated the information gain for finding the best split for a node

between two or more separate taxonomies. This approach does not incorporate any kind of semantic inference or contextually driven attributes for building a decision tree, neither does it address any issues with trying to make a decision tree parallelizable. Gajderowicz et al. formulated an approach for enriching manually created ontologies using decision trees [3]. They also developed a system for using decision trees for ontology matching [4]. Johnson et al., formulated an approach for enriching ontologies off of custom built decision trees [5]. Bouza et al. described an approach by using an ontology to build user profiles to make various recommendations on user behavior [6]. Our approach not only allows a user to specify preferences, but also ensures that they are semantically similar to any search results; it is parallelizable too. Fanizzi et al. developed a novel framework for learning custom description logic learning languages using decision trees. While this approach is novel for learning description logic concept definitions, it does not incorporate user preferences [7].

Nenkova et al. developed a technique for summarizing documents based on frequency [8]. Arun and Gunavathi developed a technique for summarizing documents using context sensitive weights for indexing [9]. This work did not utilize the contextual properties of parts of a document nor were user preferences utilized when creating the summaries. Witte et al. developed a fuzzy graph technique for multi-document summarization [10]. Barzilay et al. developed a technique for summarizing documents using the contextual attributes found in text across a series of documents [11]. Yang et al. developed a summarizing framework using the social contextual information, but they did not utilize user specified preferences beyond social ones such as a time or location that a document was created [12].

VI. CONCLUSIONS

We described and validated an approach to specify semantically driven user preferences in a parallelizable way. We also encountered a few limitations. First, we found that extensive exploration and sampling of data files was needed to be able to properly model preferences in our ontology to confirm a consistent structure of a file format when creating our in memory tree model. Second, we found that the user preferences that were specified in the in memory model needed to closely mirror the user preferences that were specified in the ontology or this would lead to searches returning incorrect results or errors resulting in our program because the structure of RDF/OWL Model was incorrect. For our future work we plan on testing much larger datasets. Also planned are further attempts to model more expressive attributes for user specified preferences. Finally, we plan on utilizing Apache Spark [13] and new versions of Hadoop to allow for both a more novel design and implementations of our approach.

References

[1] A. Zimmermann, A. Lorenz and R. Oppermann, "An Operational Definition of Context," 2007.

[2] J Zhang, A. Silvescu and V. Honava, "Ontology-Driven Induction of sion Trees at Multiple Levels of Abstraction," Berlin Heidelberg, 2002.

[3] B. Gajderowicz, M. Soutchanski and A. Sadeghian, "Trees, Ontology Enhancement through Inductive Decision Trees," in *Uncertainty Reasoning for the Semantic Web II*, Springer Berlin Heidelberg, 2013.

[4] B. Gajderowicz, "Using Decision Trees for Inductively Driven Semantic Integration and Decision Matching," Ryerson University, Toronto, 2011.

[5] I. Johnson, J. Abécassis, B. Charnomordic, S. Destercke and R. Thomopoulos, "Making Ontology-Based Knowledge and Decision Trees Interact: An Approach to Enrich Knowledge and Increase Expert Confidence in Data-Driven Models," in *Knowledge Science, Engineering and Management*, Springer Berlin Heidelberg, 2010.

[6] A. Bouza, G. Reif, A. Bernstein and H. Gall, "SemTree: Ontology-Based Decision Tree Algorithm for Recommender Systems," Karlshruhe, 2008.

[7] N. Fanizzi, C. d'Amato and F. Esposito, "Induction of Concepts in Web Ontologies through Terminological Decision Trees," Barcelona, 2010.

[8] A. Nenkova, L. Vanderwende and K. R. McKeown, "A Compositional Context Sensitive Multi-document Summarizer: Exploring the Factors That Influence Summarization," Seattle, Washington, 2006.

[9] J. M.E and C. Gunavathi, "Document Summarization and Classification sing Concept and Context Similarity Analysis," 2014.

[10] R. Witter, R. Krestel and S. Bergler, "Context-based Multi-Document Summarization Using Fuzzy Coreference Cluster Graphs".

[11] R. Barzilay, K. R. McKeown and M. Elhadad, "Information Fusion in the Context of Multi-Document Summarization".

[12] Z. Yang, C. Keke, J. Tang, Z. Li, S. Zhong, L. Jaunzi and Y. Zi, "Social Context Summarization," Beijing, China, 2011.

[13] Apache Spark, "Apache Spark," [Online]. Available: http://spark.apache.org. [Accessed 10 06 2016].

[14] T. Boujari, "Instance-Based Ontology Alignment Using Decision Trees," Institutionen för Datavetenskap, 2012.

[15] H. F. Witschel, "Using Decision Trees and Text Mining Techniques for Extending Taxonomies," in *In Proceedings of Learning and Extending Lexical Ontologies by Using Machine Learning Methods*, 2005.

[16] W. J. Wei Dai, "A MapReduce Implementation of C4.5 Decision Tree Algorithm," *International Journal of Database Theory and Application,* pp. 49-60, 2014.

[17] R. Mirambicka, A. R. Sulthana and G. Vadivu, Decision Tree Applied to Learning Relations Between Ontologies.

[18] D. Jeon and W. Kim, "Development of Semantic Decision Tree," in *Data Mining and Intelligent Information Technology Applications (ICMiA), 2011 3rd International Conference on*, 2011.

[19] M. Patil, S. Khomane, V. Saykar and K. Moholkar, "Web People Search Using Ontology Based Decision Trees," *International Journal of Data Mining & Knowledge Management Process,* 2012.

[20] B. Panda, J. S. Herbach, S. Basu and R. J. Bayardo, "PLANET: Massively Parallel Learning of Tree Ensembles," Google, 2009.

[21] J. Han, Y. Liu and X. Sun, "A Scalable Random Forest Algorithm Based on MapReduce," in *Software Engineering and Service Science (ICSESS), 2013 4th IEEE International Conference on*, Bejing, China, 2013.

[22] W. Yin, V. Simmhan and V. K. Prasanna, "Scalable Regression Tree Learning on Hadoop Using OpenPlanet," in *MapReduce '12 Proceedings of third international workshop on MapReduce and its Applications*, New York, New York, 2012.

[23] S. d. Río, V. López, J. M. Benítez and F. Herrera, "On The Use of MapReduce For Imbalanced Big Data Using Random Forest," *Information Sciences,* pp. 112-137, 2014.

[24] S. Tyree, K. Q. Weinberger and K. Agrawal, "Parallel Boosted Regression Trees for Web Search Ranking," in *WWW 2011 – Session: Ranking*, NY, NY, 2011.

[25] X. Zhanga, C. Liua, S. Nepalb, C. Yanga, W. Dou and J. Chen, "A Hybrid Approach For Scalable Sub-Tree Anonymization Over Big Data Using MapReduce On Cloud," *Journal of Computer and System Sciences,* pp. 1080-1020, 2014.

[26] J. Ye, J.-H. Chow, J. Chen and Z. Zheng, "Stochastic Gradient Boosted Distributed Decision Trees," in *CIKM '09 Proceedings of the 18th ACM Conference on Information and Knowledge Management*, New York, New York, 2009.

[27] A. Ghoting, P. Kambadur, E. Pednault and R. Kannan, "NIMBLE: a Toolkit For The Implementation of Parallel Data Mining and Machine," in *KDD '11 Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, New York, 2011.

[28] G.-Q. Wu, H.-G. Li, X.-G. Hu, Y.-J. Bi, J. Zhang and W. Xindong, "MReC4.5: C4.5 Ensemble Classification with MapReduce," in *2009 Fourth ChinaGrid Annual Conference*, Yantai, Shandong, 2009.

Using Ontologies to Quantify Attack Surfaces

Michael Atighetchi, Borislava Simidchieva, Fusun Yaman
Raytheon BBN Technologies
Cambridge, MA 02138 USA
{matighet | simidchieva | fusun}@bbn.com

Thomas Eskridge, Marco Carvalho
Florida Institute of Technology
Melbourne, FL 32901 USA
{teskridge | mcarvalho}@fit.edu

Captain Nicholas Paltzer
Air Force Research Laboratory
Rome, NY 13441 USA
nicholas.paltzer@us.af.mil

***Abstract*—Cyber defenders face the problem of selecting and configuring the most appropriate defenses to protect a given network of systems supporting a certain set of missions against cyber attacks. Cyber defenders have very little visibility into security/cost tradeoffs between individual defenses and a poor understanding of how multiple defenses interact, which, in turn, leads to systems that are insecure or too overloaded with security processing to provide necessary mission functionality. We have been developing a reasoning framework, called Attack Surface Reasoning (ASR), which enables cyber defenders to explore quantitative tradeoffs between security and cost of various compositions of cyber defense models. ASR automatically quantifies and compares cost and security metrics across multiple attack surfaces, covering both mission and system dimensions. In addition, ASR automatically identifies opportunities for minimizing attack surfaces, e.g., by removing interactions that are not required for successful mission execution. In this paper, we present the ontologies used for attack surface reasoning. In particular, this includes threat models describing important aspects of the target networked systems together with abstract definitions of adversarial activities. We also describe modeling of cyber defenses with a particular focus on Moving Target Defenses (MTDs), missions, and metrics. We demonstrate the usefulness and applicability of the ontologies by presenting instance models from a fictitious deployment, and show how the models support the overall functionality of attack surface reasoning.**

I. Introduction

Cyber security remains one of the most serious challenges to national security and the economy that we face today. Systems employing well known but static defenses are increasingly vulnerable to penetration from determined, diverse, and well resourced adversaries launching targeted attacks such as Advanced Persistent Threats (APTs).

Due to the heavy focus on cyber security technologies in both commercial and government environments over the last decade, an overwhelming array of cyber defense technologies have become available for cyber defenders to use. As the number and complexity of these defenses increase, cyber defenders face the problem of selecting, composing, and configuring them, a process which to date is performed manually and without a clear understanding of integration points and risks associated with each defense or combination of defenses.

As shown in Figure 1, the current state-of-the-art approach for selecting and configuring cyber defenses is manual in nature and is often done without a clear understanding of security metrics associated with attack surfaces. Due to the talent shortage in cyber security Subject Matter Experts (SMEs) [9], this introduces significant delays and cost.

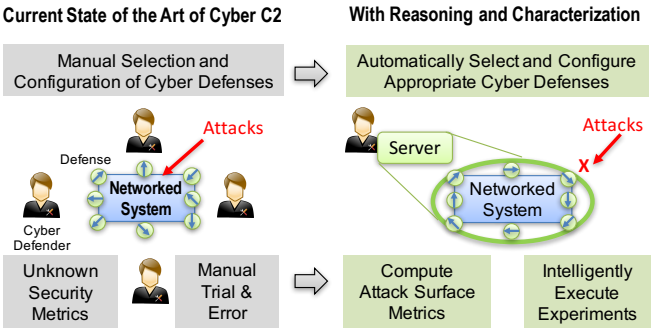


Fig. 1. The proposed approach computes attack surface metrics, provides structured support for deployment of (and experimentation with) wrapped defenses, and automates the defense selection and configuration process

The reasoning framework presented in this paper aims to significantly improve the level of rigor and automation associated with selection and configuration of cyber defenses. Using an ontologically grounded definition of an attack surface, the framework contains algorithms to find all applicable attack vectors and compute metrics for the security and cost impact of adding cyber defenses to target systems. Using models of key mission processes and their interactions, the analysis extends observations about system-level components to the resulting impact on execution of mission critical workflows. Finally, the framework combines measurement, modeling, and analysis with testing of software artifacts through the use of a virtualized test infrastructure [1]. Experimental validation of analysis results on real systems with real defense implementations establishes the usefulness and validity of the approach.

Figure 2 illustrates how the Attack Surface Reasoning (ASR) framework captures models of underlying systems, cyber defenses, and missions in the form of unified models. These models are augmented by other models that describe adversary constraints, potential attack steps, and definitions of security and cost metrics. ASR provides two categories of algorithms: attack surface characterization and minimization. The characterization algorithm constructs attack vectors and calculates security and cost metrics. The minimization algorithm uses system and mission information to identify opportunities for pruning unnecessary access paths to reduce

Distribution Statement "A" (Approved for Public Release, Distribution Unlimited). This material is based upon work supported by the Air Force Research Laboratory under Contract No. FA8750-14-C-0104.

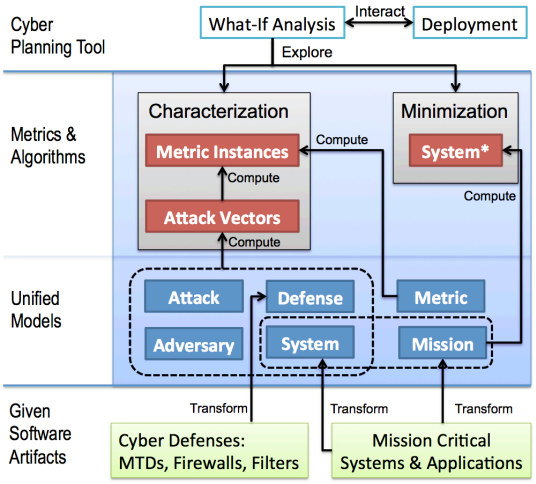


Fig. 2. The Attack Surface Reasoning (ASR) framework

the attack surface. Using the models, algorithms, and metrics, cyber defenders can compare various deployments of proactive cyber defenses in a quantitative manner and contrast tradeoffs between security benefits and performance overhead. As such, ASR provides a foundational capability in support of an envisioned cyber planning tool that automatically suggests and configures defenses given mission executions over systems.

This paper describes the ontologies used to model systems, cyber defenses, adversarial capabilities, and mission constraints. Validation of the approach focuses on a specific class of proactive cyber defenses, Moving Target Defenses (MTDs) [7], [11]. MTDs claim to make entry points into networks and systems harder to detect, thereby reducing vulnerabilities and making the exposure to those vulnerabilities that remain more transient. This introduced dynamism ought to render attacks against MTD-protected systems less effective, but few quantitative results are available to date, which makes MTDs a prime choice for quantification.

The rest of the paper is organized as follows. Section II describes related work in threat modeling and analysis. Section III describes the set of ontologies we developed to support attack surface reasoning. Section IV reports on the validation results of applying the ontologies to cyber defense operations of a small enterprise network. Section V concludes the paper.

II. Related Work

The ontologies presented in this paper relate to several approaches for modeling cyber security systems and observables.

A. Security Standards

A number of different taxonomies exist for describing cyber security related information. For threat information, this set of standard includes the Common Vulnerabilities Enumeration (CVE), Common Weakness Enumeration (CWE), Common Vulnerability Scoring System (CVSS), Malware Attribute Enumeration and Characterization (MAEC), Structured Threat Information eXpression (STIX), and Common Attack Pattern Enumeration and Configuration (CAPAC). Taxonomies in use for system modeling include the Cyber Observable eXpression (CybOX) and the Common Information Model (CIM). These standards focus on capturing detailed information about system observables, cyber security events, indicators of compromise, and vulnerabilities for the purposes of sharing specific threat information (to yield enhanced intrusion detection) and eliminating existing vulnerabilities (through continuous patching). In contrast, the ASR ontologies are expressed at a higher level of abstraction and focus on design-level assessments of attack surfaces. Another difference is that the ASR ontologies are expressed in OWL, while the community standards mentioned above are prescribed in XML. Finally, the above-mentioned standards focus on system and adversary modeling, but provide no structured means for representing cyber defense capabilities. In contrast, ASR contains a specific defense ontology describing the protection provided by defenses and the cost associated with various defense configurations.

B. Security Ontologies

A number of different ontologies exist for expressing security-related properties, including [6] and [4], as summarized in [13]. [5] applies semantic threat and defense modeling to identify proper firewall configurations. [14] develops an ontology for the HTTP protocol as well as attacks against web applications (using HTTP), and then uses a separate ontology for finding attack vectors. [10] focuses on a review of existing cyber security taxonomies and ontologies and points out several existing models. However, the review does not list any ontologies for cyber defenses. [15] describes an extensive ontology supporting forensic activities across disparate data sources. Finally, work on modeling cyber defense decision processes [3], [12] provides ontology support for learning and extracting cyber defense workflows and decision procedures.

The ASR ontologies are in large inspired by the STRIDE threat-modeling approach [16] used by Microsoft. One key difference to existing ontologies is the focus on abstract architectural concepts and high-level adversarial objectives.

III. Ontologies

The attack surface reasoning algorithms operate over a set of models that together describe the system under examination, its defenses, the assumed capabilities and starting point(s) of the adversary, and optionally a mission or set of missions which may operate over the defined system. In addition, the set of metrics to be computed is itself described in a model to allow for easy extension and modification by the user.

ASR models are defined in the WorldWideWeb Consortium (W3C) semantic Web Ontology Language (OWL). Using a semantic web substrate provides a number of benefits, including:

- Scalability: the OWL language and supporting tools allow for scaling to very large models;
- Inference: OWL ontologies encode meaning in a formal way, which enables inferring new facts from existing data;
- Cross-domain integration: OWL ontologies can connect disparate domains without contaminating the sources;

- Standards and community: OWL and associated languages such as Resource Description Framework (RDF) and SPARQL Protocol And RDF Query Language (SPARQL) provide interoperable libraries and tooling, and active practitioner communities; and
- Relative maturity: semantic web languages provide tested algorithms, established terminology, and relatively mature libraries. Tooling with predictable performance both within and beyond the laboratory setting is also available.

One of the key challenges of modeling distributed systems is to identify the level of abstraction most appropriate for the modelers who will create the models, the algorithms that will operate over them, and the results that are provided to stakeholders. Modeling at the extreme of precision allows exact answers to be derived, but creates models that are difficult to accurately create and to keep up to date, and leads to analysis outcomes that are brittle as the system changes. On the other hand, modeling at too coarse of a level of abstraction leads to easily created models, but models that can tell little to interested parties about questions of importance.

We took a middle road with ASR. A number of the concepts, and the level of granularity, were modeled after the Microsoft STRIDE [8] threat-classification framework and related modeling languages described in [16]. STRIDE expresses system concepts through abstract concepts including processes, data flows, boundaries, external entities, and data stores. We model the different aspects of an attack surface separately in order to facilitate modularity and extensibility. Table I lists the six ontological models used in ASR and summarizes their content.

TABLE I
ASR USES A COLLECTION OF MODELS TO QUANTIFY ATTACK SURFACES

Model	Concepts
System	System components and their relationships; e.g., computational entities, boundaries, and data flows
Attack	Generic attack logic as individual steps, vectors, and templates
Adversary	Adversarial starting position and goal
Mission	Mission relevant system elements and key performance metrics
Defense	Cyber defense capabilities in terms of protections provided plus associated costs
Metric	Metrics for security, cost, and mission impact

A. System Model

System models describe the business system against which attacks can be executed and within or around which defenses can be deployed. These models detail the hosts in the system, the networks that connect these hosts, and the processes that run on them. Data flows are modeled here at three different layers: process, network, and physical. The three layers are interconnected in the model such that one can determine for a given process-layer data flow that the described data is sent out through a given endpoint at the network layer, which in turn is bound to a particular network interface card (NIC) at the physical layer. Table II describes the main resource types associated with the system model ontology.

TABLE II
MAIN SYSTEM MODEL CONCEPTS

Resource	Description
Entity	General concept
Boundary	Trust realm for unrestricted access within a boundary
Vertical Boundary	subclassOf Boundary describing realm cross layers
Horizontal Boundary	subclassOf Boundary describing realm on a single layer
Host	subclassOf Vertical Boundary representing a computer system
WAN	subclassOf Horizontal Boundary representing a wide area network
VLAN	subclassOf Horizontal Boundary representing a wide area network
Layer	Logical layering of functionality into three main layers
NetworkLayer	subclassOf Layer describing network entities and interactions
PhysicalLayer	subclassOf Layer describing physical entities
ProcessLayer	subclassOf Layer describing application-level components and interactions
DataFlow	Flow of bits between two entities
DataStore	Persistent store of information
External	An entity that is external to the system
User	subclassOf External describing human actors
NetworkEndpoint	Sockets used in network connections
NIC	Network Interface Card
Process	Operating System process
Resource	Shared resource with certain capacity

The following properties have specific meaning:

- contains: expresses membership relationship between two Entities. For instance, a Host contains Processes and a VLAN contains NICs.
- connectsTo: expresses a data or control flow link between two Entities. For instance, a User connects to a Process, a Process connects to a NetworkEndpoint, and a NetworkEndpoint connects to a NIC.
- via: expresses a link between hierarchical data flows. For example, a process-layer flow is realized via a network-layer flow, which itself happens via a physical-layer flow.

B. Attack Model

The attack model describes the generic activities performed by adversaries as a collection of potential attack steps. Table III describes the main resource types associated with the attack model ontology. Each attack step definition comprises a number of attributes that specify an attack type (modeled via the six high-level types of attacks whose initials define STRIDE), the pre-conditions necessary for the attack step to execute, and the post-conditions that holds once the attack step executes successfully. Figure 3 shows an example of an attack step definition that represents network sniffing, and Table IV shows the set of attack step definitions that are currently modeled in ASR, using the STRIDE attack types from Table III.

C. Adversary Model

The adversary model contains the following information:

- Starting Position: A reference to an entity in the system model that describes the starting privilege an adversary has for the purpose of a specific assessment.
- Target Goal: Information about the type of attack and the intended target of the attack.

TABLE III
MAIN ATTACK MODEL CONCEPTS

Resource	Description
AttackStep	A specific instance of adversarial activity. Attack vectors consists of a collections of linked attack steps.
AttackStepDefinition	A reusable generic description of an adversarial activity. Attack steps are derived from definitions
AttackVectorElement	Ordering and context around an AttackStep to form an AttackVector
AttackVector	Ordered execution of AttackSteps
AttackTemplate	A templatized version of an attack vector
Attacker	Captures aspects of the expected adversary, including the starting position
SideEffect	As part of executing this attack, these specific facts are added to the model
AttackType	**The type of attack being executed**
Spoofing	subclassOf AttackType. Illegally accessing and then using another user's authentication information.
Tampering	subclassOf AttackType. Malicious modification of data
Repudiation	subclassOf AttackType. Deny performing an action without other parties having any way to prove otherwise
InformationDisclosure	subclassOf AttackType. Exposure of information to individuals who are not supposed to have access to it
DenialOfService	subclassOf AttackType. Deny service to valid users
ElevationOfPrivilege	subclassOf AttackType. An unprivileged user gains privileged access and thereby has sufficient access to compromise or destroy the entire system

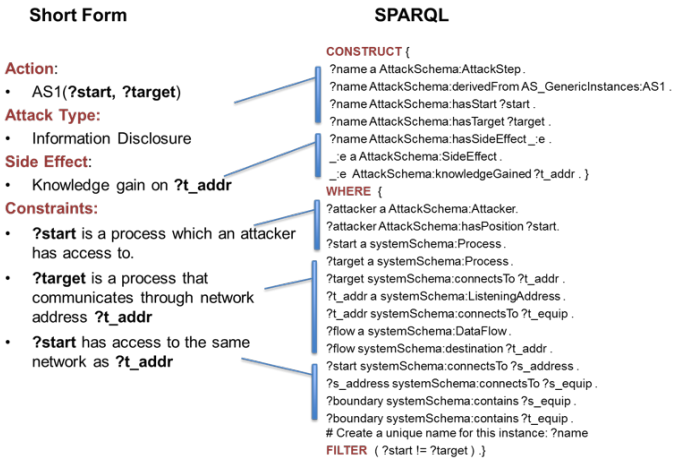


Fig. 3. Example of an attack step that performs a network sniffing action

TABLE IV
ATTACK STEPS CURRENTLY MODELED IN ASR

Name	Type	Pre-Condition	Post-Condition
Sniff	Information Disclosure	Access to network	Knowledge about observed network flows
PortScan	Information Disclosure	Network reachability	Knowledge about listening sockets
TCPConFlood	Denial of Service	Network reachability & Knowledge about the target endpoint	Depletes file descriptors at a given rate
OSFingerPrint	Information Disclosure	Knowledge on listening socket on a host	Knowledge about host OS specifics
GetRoot	Elevation of Privilege	Knowledge on host OS and listening socket	Root privilege on host
ShutDownServer	Denial of Service	Knowledge on host OS and listening socket Root privilege on host	Server unavailable

- Attack Vector Template: Preconceived structure of attack vectors specifying sequences of types of attack steps that have not been bound to specific instances.

Given these assumptions about the adversary, ASR will automatically identify all applicable attack vectors as a partially ordered sequence of bound attack steps.

D. Mission Model

Mission models describe mission-critical flows between actors and services at the application layer. The mission models are a strict subset of process-layer system entities and data flows contained in the system model. Table V shows the main concepts in the ASR mission models.Mission metrics evaluate the fitness of a specific mission within the context of a collection of other models. Like system metrics, mission metrics are evaluated along the two dimensions of cost and security, and mission-critical flows can specify requirements on the cost and security of information exchanges. Most mission metrics are rated on a *normal*, *degraded*, *fail* scale. To allow for quick and easy comparison of mission metrics among multiple configurations, we provide a mission aggregate cost index (ACI) and a mission aggregate security index (ASI), which return the minimum score along all cost or security concerns, respectively (i.e., if a single data flow fails a cost or security requirement, the mission aggregate cost or security index indicates a *fail* also). The individual metrics are provided for comparison purposes so that it is easy for the user to distinguish between a configuration that only has one or two poorly performing components for this mission, and an overall equally rated configuration whose every component is rated *degraded* or *fail* for this mission. Finally, the mission security and cost metrics are folded into an aggregate mission index (AMI), similar to the ACI and ASI. The value of the AMI

is *fail* if either the mission aggregate security or cost indices evaluates to *fail*, and equals the mission aggregate cost rating otherwise (this is because security is evaluated on a *pass/fail* scale, while cost follows the user-defined three-band ranking explained in detail below).

Mission performance is constrained through four threshold values, $p1_{latency}, p2_{latency}, p1_{throughput}, p2_{throughput}$, that describe lower and upper allowable thresholds for percentage overhead rates on latency and throughput. Not all mission-critical data flows must specify a lower and upper threshold, and, if there is no requirement on a data flow, user-configurable default threshold values will be used. These thresholds are used to define the following three bands:

- Normal ($p_{latency} < p1_{latency}$): The mission operates within normal parameters, i.e. the greatest latency penalty incurred is still less than the lower threshold.
- Degraded ($p1_{latency} <= p_{latency} < p2_{latency}$): The mission can continue, though with sub-optimal outcomes, i.e. the greatest latency penalty incurred is more than the lower threshold but less than the maximum allowable.
- Fail ($p2_{latency} < p_{latency}$): The mission cannot continue and misses objectives, i.e. the greatest latency penalty incurred exceeds the maximum allowed and the mission performance will be unacceptable.

For example, the user can specify that a latency penalty of up to 10% is acceptable if it allows for a more sophisticated defense to be deployed with a mission, but a latency penalty of 40% or more leads to unacceptable delays and jeopardizes the mission. In this case, if the cumulative latency along some mission-critical data flows does not exceed 110% of the normal value, these data flows are rated as *normal*; if the latency exceeds 110% but is below 140%, corresponding data flows are rated as *degraded*; and if the latency is over 140% of the original value, those data flows are rated as *fail*. The throughput calculations are analogous, with the exception that a penalty means a decrease, not an increase, in throughput.

Mission security requirements specify any required security attributes, which are delineated among confidentiality, integrity, and availability. Not all mission-critical data flows must specify a security requirement and if no requirement is specified, the data flow is not considered when evaluating mission security. Security metrics are evaluated on a binary scale where a data flow either meets its security requirement or violates it. A data flow is considered to violate a security requirement if an attack step can compromise that requirement.

For example, since all attack steps are categorized using STRIDE, if an attack step contributes to a denial of service on a data flow and that data flow has an availability requirement, the requirement is violated. If the same data flow also has confidentiality or integrity requirements, those are evaluated separately with respect to other attack steps that might compromise them. If at least one mission-critical data flow is found to violate a security-related requirement, that requirement is rated as *fail* for the entire mission. For example, if there are three data flows with integrity requirements and only one of them violates a requirement, then the mission still gets a *fail* score for integrity. If any of the individual percentages of data flows that fail for confidentiality, integrity or availability are greater than zero, the mission aggregate security index consequently evaluates to a *fail* score on security overall.

TABLE V
MAIN MISSION MODEL CONCEPTS

Resource	Description
Mission	Description of mission requirements over data flows
Requirement	Specifies thresholds for cost and minimum security requirements for a data flow
MetricType	Type of mission metrics
Integrity	⊂ MetricType. Security constraint
Availability	⊂ MetricType. Security constraint
Confidentiality	⊂ MetricType. Security constraint
Latency	⊂ MetricType. Cost constraint via performance impact
Throughput	⊂ MetricType. Cost constraint via performance impact

E. Defense Model

The defense models describe which static and dynamic defenses are in place, what elements of the system they protect, what types of coverage they provide, and what cost is incurred. A single defense model can incorporate multiple defenses. Table VI shows the main concepts associated with models of cyber defenses. Different defenses operate over different types of nodes and thus the coverage relationship from a defense has a range of type Entity, which in the ASR ontologies inheritance hierarchy is the parent of all system-level nodes (processes, hosts, NICs, etc.). In this way, MTDs from Address Space Layout Randomization (ASLR) to IP Hopping can all integrate with the system model in a uniform manner, despite the fact that they protect very different elements. Defenses can be modeled both abstractly, such as a generic definition for a firewall, and at the specific implementation level (e.g., IPTables).

Thanks to the ability of OWL to incorporate inheritance, we can reap the benefits of reuse. We can define a generic IP hopping MTD that describes the capabilities and requirements common to all IP hopping defenses, and extend this definition to minimize the effort needed to model any specific implementations of an IP hopping defense. We can even

TABLE VI
MAIN DEFENSE MODEL CONCEPTS

Resource	Description
Defense	Description of cyber defense mechanism
DefenseType	Categorization into different types of defenses
Cost	Characterization of the overhead defense incurred
Degradation	⊂ Cost. Reduction in metric.
Requirement	Prerequisite requirements for installing the defense
Setup	Description of the defense's configurable items
Protection	Security guarantees provided by the defense
Reconfiguration-Detail	Description of dynamic behavior associated with MTDs
ProtectionDetail	Description of target entities being covered by defense
Randomization-Detail	Description of the randomization space

analyze this generic instance without reference to a specific implementation to provide insight into how the entire class of defenses operates. In order to support the dynamic nature of MTDs, the defense model provides support for the proactive elements of a defense to be described. An IP hopping MTD may be configured to change IP addresses of the included NICs every 5 minutes, for example.

Our current approach divides MTDs into three main kinds, and Table VII shows the set of proactive defenses currently modeled in ASR that cover two of the three categories:

1) Time-bound observable information on targets. In this category, MTDs place limits on the useful life of information obtained in an execution step for use in a later execution step. IP Hopping in the context of TCP Connection flooding is an example of this.
2) Masquerade targets. MTDs in this category make a target look like another kind of target, causing an adversary to spend extra cycles. OS masquerading is an example of this effect.
3) Time-bound footholds. MTDs in this category reset the escalated privileges that an attacker has built up along the middle of an attack path. An example of this is the use of virtualization and watchdogs to proactively and continuously restart VMs to clear out corruption.

TABLE VII
DEFENSES CURRENTLY MODELED IN ASR

Name	Kind	Requires	Side Effect
IPHopping	Time-bound observable	Network Endpoints	IP changes at fixed intervals
OS Masquerading	Masquerade	Host OS image	Host OS image fake
OS Hopping	Time-bound observable	Multiple OSs compatible with applications	Host changes at fixed intervals

F. Metrics Model

The metrics model enumerates all ASR metrics and defines each metric's name, the domain over which it is executed, and the SPARQL query used to compute it. ASR computes a diverse set of both system- and mission-based metrics over a configuration. Most metrics are computed by querying other models (e.g., to count the total number of listening endpoints or of attack vectors found). Some metrics are post-processed to compute statistical attributes such as mean (e.g. to compute the average estimated duration of an attack vector) or maximum or minimum values (e.g., to find the shortest attack vector).

These metrics are meant to give the user an overview of how well a system is protected against a set of attacks executed by a modeled adversary, as well as what costs (in terms of latency and throughput) are incurred by the modeled defenses. To facilitate this cost-benefit analysis, ASR provides users with some index metrics that can be used to judge a configuration's fitness at a glance, and compare fitness between alternative solutions. Figure 4 illustrates how the

	Security	Cost
System	Aggregate Security Index summarizes security metrics	Aggregate Cost Index summarizes cost metrics
	Individual concerns such as length of attack vectors are easily viewed	Individual concerns such as latency penalty of defense can be examined
Mission	Missions are ranked as *pass*, *degraded*, or *fail*. Lowest score prevails for aggregate mission scores along security, cost, or both	
	Individual concerns such as confidentiality still accessible	Individual concerns such as percent of dataflows that fail latency viewable

Fig. 4. High-level ASR metrics

metrics are separated into security- and cost-related concerns along one axis, and along system- and mission-wide metrics along the other axis. Security and cost are frequently at odds, with higher security necessitating a more expensive defense. A single value may therefore be misleading to a user because it could either represent the ideal case of high security and low cost, or the clearly undesirable outcome of low security and high cost. For these reasons, ASR provides the user with a separate single-value index reflecting the cost of any deployed defenses (the Aggregate Cost Index, ACI) and another single-value index reflecting the security score of the current configuration (the Aggregate Security Index, ASI). If a mission model is specified, a third index reflecting the fitness of the configuration with respect to mission goals is also computed (the Aggregate Mission Index, AMI). The index metrics are composed of several lower-level metrics, as shown in Figure 5. The desired metrics are specified in an OWL ontology, which is user-extensible and customizable. The metric computation is done through SPARQL queries for both simple and aggregate metrics, and the Jena API is used to invoke the metric computation from the ASR server and store the results.

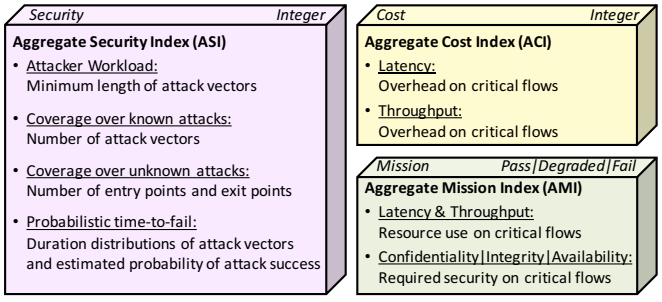


Fig. 5. The ASR index metrics take into account many submetrics

IV. EXEMPLAR APPLICATION OF THE ONTOLOGIES

To evaluate the modeling and reasoning performed by ASR, we developed an enterprise information sharing scenario involving several servers and both mobile and wired networks. Figure 6 shows the main actors participating in the scenario together with their interactions. An InformationProducer (e.g., a web camera) is sending videos and still images to a Website,

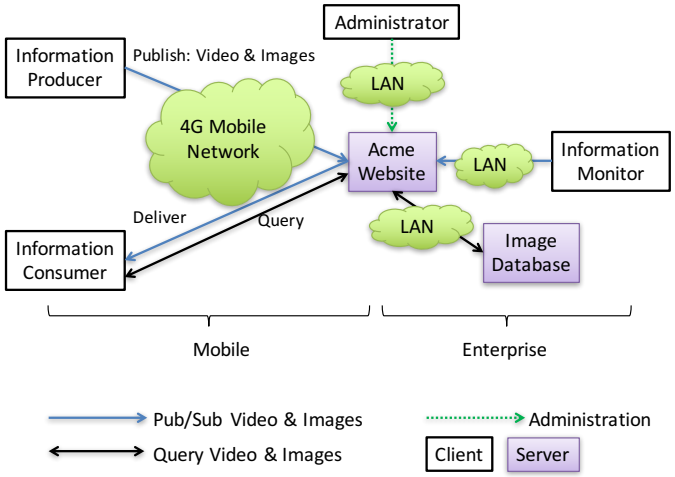


Fig. 6. Example information sharing scenario used to validate the approach

which in turn disseminates both video and images to two clients: an Information Consumer over a 4G mobile network and an Information Monitor over a Local Area Network. The Website is connected to an Image Database for persistence of images received. Finally, an Administrator can change settings on the Website through an administrative client.

A. Instance Models

Transcription of the components mentioned in the scenario involves creating instance models that are consistent with the ASR ontologies. To do this, we first define prefix shortcuts for name spaces as follows, using TURTLE:

```
@prefix demo1: <http://www.bbn.com/asr/demo1#> .
@prefix def: <http://www.bbn.com/asr/def#> .
@prefix sm: <http://www.bbn.com/asr/sm#> .
@prefix IPHop: <http://www.bbn.com/asr/iphop#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
```

The “Acme Website” host and its components can be expressed as:

```
% Acme Website from Figure 6
demo1:AcmeServer1
  rdf:type sm:Host ;
  rdf:type owl:Thing ;
  sm:contains demo1:Endpoint2 ;
  sm:contains demo1:ACME1 ;
  sm:hasImage demo1:OperatingSystem_1 .

 % Process running on the Acme Website Server
 demo1:ACME1
  rdf:type sm:Process ;
  rdf:type owl:Thing ;
  sm:connectsTo demo1:Endpoint2 .

 % NetworkEndpoint that ACME1 process connectsTo
 demo1:Endpoint2
  rdf:type sm:ListeningEndpoint ;
  rdf:type sm:NetworkEndpoint ;
  rdf:type owl:Thing ;
  sm:connectsTo demo1:MNE2 ;
  sm:hasResource sm:FileDescriptorPool_1 .

% Acme Website's MNE on the 4G Mobile Network
demo1:MNE2
  rdf:type sm:MNE ;
  rdf:type owl:Thing .
```

The MNE is plugged into a Mobile Network and there is a network flow coming in over that network that is expressed at three distinct layers that are linked through the “via” property.

```
% 4G Mobile Network from Figure 6
demo1:MobileNetwork1
  rdf:type sm:WAN ;
  rdf:type owl:Thing ;
  sm:contains demo1:MNE1 ; % Information Producer's MNE
  sm:contains demo1:MNE2 . % Acme Website's MNE

% Process-layer data flow from IP1 to ACME1
demo1:pDataFlow1
  rdf:type sm:DataFlow ;
  rdf:type owl:Thing ;

% Process on Acme Website defined above
  sm:destination demo1:ACME1 ;

% Process on Information Publisher from Figure 6
  sm:source demo1:IP1 ;
  sm:via demo1:nDataFlow1 .

% Underlying network-layer data flow
demo1:nDataFlow1
  rdf:type sm:DataFlow ;
  rdf:type owl:Thing ;
  sm:destination demo1:Endpoint2 ;
  sm:source demo1:Endpoint1 ;
  sm:via demo1:gDataFlow1 .

% Underlying physical-layer data flow
demo1:gDataFlow1
  rdf:type sm:DataFlow ;
  rdf:type owl:Thing ;
  sm:destination demo1:MNE2 ;
  sm:source demo1:MNE1 .
```

An Internet Protocol Address randomization (IP Hopping) defense is installed to cover the data flow between Endpoint 1 (the Information Producer) and Endpoint2, the Acme Website. The defense adds an additional data flow and processes for key synchronization. It also specifies necessary setup and configuration details and the incurred costs.

```
def:IPHopping1
  rdf:type def:Defense ;
  def:adds IPHop:DataFlow_pKeySharing ;
  def:adds IPHop:IPHoppingProcess_ACME ;
  def:adds IPHop:IPHoppingProcess_InfoProducer ;
  def:atCost IPHop:Cost_1 ;
  def:provides IPHop:Protection_1 ;
  def:requires IPHop:Setup_1 .

IPHop:Protection_1
  rdf:type def:Protection ;
  def:for demo1:Endpoint1 ;
  def:for demo1:Endpoint2 ;
  def:inSupportOf def:Confidentiality ;
  def:inSupportOf def:Discoverability ;
  def:through def:Randomization ;
  def:withSpecific IPHop:RandomizationDetail_1 .

IPHop:RandomizationDetail_1
  rdf:type def:RandomizationDetail ;
  def:disruptionLatency "5"^^xsd:float ;
  def:interval "10000"^^xsd:float ;
  def:space 6 .

IPHop:Setup_1
  rdf:type def:Setup ;
  def:includes demo1:Endpoint1 ;
  def:includes demo1:Endpoint2 .

IPHop:Cost_1
  rdf:type def:Cost ;
  def:impactOn IPHop:Latency_1 .

IPHop:Latency_1
  rdf:type def:MetricType ;
```

```
def:forProperty def:Latency ;
def:increase "0.3"^^xsd:float ;
def:on demo1:nDataFlow1 .
```

Further details and content for the remaining models, including attack steps, adversary, metrics, and mission, are included in the appendix to this paper and available at https://ds.bbn.com/projects/asr.html .

B. Quantification Results

To first step in quantifying an attack surface is creating a configuration containing the five model types and the metrics:

$C = (system, defense, attack, adversary, mission, metrics)$

The purpose of this evaluation was to study the impact of varying the hopping interval of one particular IP Hopping defense between slow and fast. To achieve this, we created three separate configurations where the only variable was the defense, as follows:

1) $C_{base} = (sm1, \varnothing, as_1, ap_1, mi_1, me_1)$
2) $C_{def1} = (sm_1, IPHopSlow, as_1, ap_1, mi_1, me_1)$
3) $C_{def2} = (sm_1, IPHopFast, as_1, ap_1, mi_1, me_1)$

Analyzing these three configurations using the ASR reasoning algorithms [2] yields the results shown in Table VIII. As a reminder, these index metrics are computed as weighted sums of several terms, as shown in Figure 5. Note that $IPHopSlow$ in C_{def1} and $IPHopFast$ in C_{def2} both add considerable cost compared to the base configuration, which contains no defense. This makes sense intuitively, since the latency penalty incurred by a defense with a shorter randomization interval (in this case, an IP Hopping defense that hops faster) is higher than the latency incurred by a defense with a longer randomization interval. The base configuration has no defenses deployed, so there is no latency penalty incurred and its ACI is therefore 0.

TABLE VIII
RESULTS OF ANALYSIS PERFORMED ON CONFIGURATIONS

Config	ASI	ACI	AMI
C_{base}	49.55	0	FAIL
C_{def1}	51.03	15.0	FAIL
C_{def2}	121.4	21.25	FAIL
C_{min}	MAX	21.25	DEGRADED

Also note that as $IPHopSlow$ in C_{def1} does not offer a significant security gain over the base configuration whereas $IPHopFast$ in C_{def2} doubles the ASI with respect to the base model. This is because in addition to submetrics that are computed over the base ontological models and do not change between the two configurations (such as the number of entry and exit points), the ASI also takes into account the probabilistic vector impact, which consists of vector duration distributions and their estimated probability of success. Intuitively, it makes sense that an IP Hopping defense that hops more frequently would provide better protection against a comparable adversary, since the adversary would have less time to complete a successful attack and would therefore be less likely to succeed. Figure 7 gives a primer on how the probability of success of attack steps and vectors is computed using the underlying ontologies.

For this example, suppose an attack step requires from 1 to 4 seconds to be successful (the duration distribution is part of the attack model) and we have a defense that hops every 1 to 3 seconds (this information is in the defense ontology). If the defense hops before the attack finishes, then the defense wins, else the attacker wins. Let us assume (for ease of computation) that both the attack step duration and the defense hopping interval are uniform random variables, which means that any number in the stated time range is equally likely and this will be captured in the sample data points. We also assume that these random variables are independent; intuitively this means that the attacker cannot detect when a hop has occurred and launch the attack immediately after the hop (which would give the attacker an unfair advantage). For this example, the probability density function for attack time needed will be

- $p_{attackDuration}(x) = \frac{1}{3}\ \forall x \mid 1 \leq x \leq 4$, and
- $p_{attackDuration}(x) = 0\ \forall x \mid x > 4$ or $x < 1$.

Similarly for defense we approximate

- $p_{defenseHoptime}(y) = \frac{1}{2}\ \forall y \mid 1 \leq y \leq 3$, and
- $p_{defenseHoptime}(y) = 0\ \forall y \mid y > 3$ or $y < 1$.

Lastly, the probability that the defense wins is computed as: $\sum(p_{attackDuration}(x) \times p_{defenseHoptime}(y)), \forall x, y \mid x > y$, which equals %66.7. Graphically, this is the normalized area to the right of the line $y = x$ in Figure 7, which represents the probability that the defense hops faster than attacker is able to successfully complete his attack.

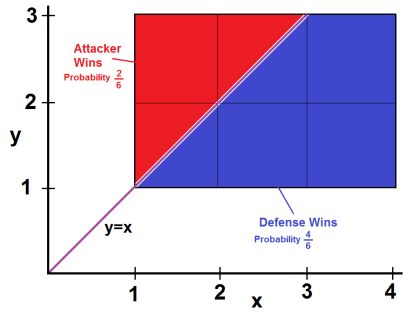


Fig. 7. A graphical representation of probability reasoning in ASR. The x axis represents the randomization interval of the defense. The y axis represents the duration distribution of an attack step that the defense is protecting against.

In addition to computing metrics, the ontologies are pivotal for another important innovation of ASR, its ability to semi-automatically minimize attack surfaces [2]. Minimization is supported through inspection and inference over all ontologies in a configuration. Two different modalities of attack surface minimization are supported:

- *System minimization* can find either entities that are not used within a system model (for instance an extraneous listening endpoint that no other endpoint connects to).
- *Mission minimization*, if a mission model is specified for a configuration, can find entities that are not defined to be mission-critical (e.g., an administrative interface that

is only used for the initial configuration of the system and never used during a mission).

Using the ontological models comprising a configuration and these two minimization modalities, ASR identifies all entities that can be safely removed and presents them to the user for selection. The user can select any or all of these entities to remove, and can save the minimized configuration for further inspection and analysis. Because removed entities may connect to other entities within the ontologies (e.g., an unused endpoint that is removed may result in an unnecessary process and its containing host, if they are not used for any other purposes), a second round of minimization may be necessary to remove all extraneous entities. The fourth configuration, C_{min}, in Table VIII is the fully minimized (i.e. with all extraneous and non-mission-critical entities removed) version of C_{def2}. Since the minimized configuration no longer contains all the entities that are not necessary (for instance, the Administrator host and associated processes, endpoints, and data flows), it has fewer entry points for an adversary to exploit and results in a higher security metric.

In all but the C_{min} configuration, the Aggregate Mission Index, AMI, is "FAIL." This is because none of them completely eliminate the attack vectors that threaten mission-critical resources. Only after minimization are all vectors are eliminated (thus the ASI score of "MAX"). The AMI is a single rating of mission health with respect to both security and cost and a single failing score on any requirement results in a failing score for the AMI. After minimization, the AMI improves from the initial "FAIL" score (initially the mission fails because of violated security requirements on mission-critical flows) to a "DEGRADED" score (the mission now passes all security requirements, but is "DEGRADED" on cost requirements). Intuitively, we have removed the security vulnerabilities that threatened the mission through deploying a faster defense and minimizing the attack surface. However, the improvement is only partial (the mission's rating is still "DEGRADED," not "PASS") due to the increased latency penalties incurred on mission-critical flows by an IP Hopping defense that hops more frequently.

We evaluated the runtime of the analysis algorithm with randomly generated models where the complexity of the models (i.e. number of hosts and other system entities and the number of available attack steps) vary in a controlled way. The points on the graph are averages of 5 runs for the same complexity configurations. The analysis time was measured on a MacBook Pro 2.8 GHz Intel Core i7 with 16 GB of RAM.

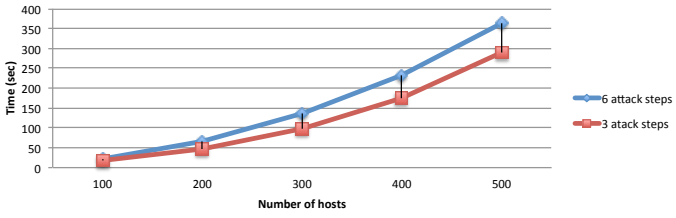


Fig. 8. ASR analysis runtime over system models of varying complexity

V. Conclusion

While it is common understanding that systems have attack surfaces and that those surfaces need to be minimized, the cyber security community has until now lacked a structured and generalizable approach for modeling attack surfaces and expressing associated security, cost, and mission impacts through concrete metrics. This paper presents ontologies including semantic models of attacks, systems, defenses, missions, and metrics, and supporting algorithms that quantify and minimize attack surfaces. An application of the ontologies on a concrete information-sharing demonstration scenario is also presented.

Next steps include extending coverage of the defense models beyond MTDs to include more traditional defenses, e.g., firewalls, VPNs, and host- and network-intrusion prevention systems. Furthermore, we plan to generate system models of realistic size systems, such as a model of the BBN network, which comprises hundreds of machines. Finally, we plan to improve the ontologies by including feedback provided by the cyber security research community.

References

[1] M. Atighetchi, B. Simidchieva, M. Carvalho, and D. Last. Experimentation support for cyber security evaluations. In *Proceedings of the 11th Annual Cyber and Information Security Research Conference*, page 5. ACM, 2016.

[2] 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 *The 27th Annual IEEE Software Technology Conference (STC)*, October 2015.

[3] N. Ben-Asher, A. Oltramari, R. F. Erbacher, and C. Gonzalez. Ontology-based adaptive systems of cyber defense. In *STIDS*, 2015.

[4] S. Fenz, T. Pruckner, and A. Manutscheri. Ontological mapping of information security best-practice guidelines. In *Business Information Systems*, pages 49–60. Springer, 2009.

[5] S. N. Foley and W. M. Fitzgerald. Management of security policy configuration using a semantic threat graph approach. *Journal of Computer Security*, 19(3):567–605, 2011.

[6] A. Herzog, N. Shahmehri, and C. Duma. An ontology of information security. *International Journal of Information Security and Privacy (IJISP)*, 1(4):1–23, 2007.

[7] S. Jajodia, A. K. Ghosh, V. Swarup, C. Wang, and X. S. Wang. *Moving target defense: creating asymmetric uncertainty for cyber threats*, volume 54. Springer Science & Business Media, 2011.

[8] L. Kohnfelder and G. Praerit. The Threats To Our Products, Apr. 1999.

[9] M. Loeb. Cybersecurity talent: Worse than a skills shortage, its a critical gap. *The Hill*, Apr. 2015.

[10] L. Obrst, P. Chase, and R. Markeloff. Developing an ontology of the cyber security domain. In *STIDS*, pages 49–56, 2012.

[11] H. Okhravi, M. Rabe, T. Mayberry, W. Leonard, T. Hobson, D. Bigelow, and W. Streilein. Survey of cyber moving targets. *Massachusetts Inst of Technology Lexington Lincoln Lab, No. MIT/LL-TR-1166*, 2013.

[12] A. Oltramari, L. Cranor, R. Walls, and P. McDaniel. Building an ontology of cyber security. In *9th Conference on Semantic Technologies for Defense, Intelligence and Security*. Citeseer, 2014.

[13] S. Ramanauskaitė, D. Olifer, N. Goranin, and A. Čenys. Security ontology for adaptive mapping of security standards. *International Journal of Computers, Communications & Control (IJCCC)*, 8(6):813–825, 2013.

[14] A. Razzaq, Z. Anwar, H. F. Ahmad, K. Latif, and F. Munir. Ontology for attack detection: An intelligent approach to web application security. *computers & security*, 45:124–146, 2014.

[15] M. B. Salem and C. Wacek. Enabling new technologies for cyber security defense with the icas cyber security ontology. In *STIDS*, 2015.

[16] A. Shostack. *Threat modeling: Designing for security*. John Wiley & Sons, 2014.

Developing an Ontology for Individual and Organizational Sociotechnical Indicators of Insider Threat Risk

Frank L. Greitzer[1], Muhammad Imran[2], Justin Purl[3], Elise T. Axelrad[4], Yung Mei Leong[5], D.E. (Sunny) Becker[3], Kathryn B. Laskey[2], and Paul J. Sticha[3]

[1]PsyberAnalytix, Richland WA, USA
[2]George Mason University, Fairfax, VA, USA
[3]Human Resources Research Organization, Alexandria, VA, USA
[4]Innovative Decisions, Inc., Vienna, VA, USA
[5]Independent Consultant, Hyattsville, MD, USA

Frank@PsyberAnalytix.com, mimran4@gmu.edu, JPurl@humrro.org, eaxelrad@innovativedecisions.com, y.leong03@gmail.com, sbecker@humrro.org, klaskey@gmu.edu, psticha@humrro.org

***Abstract*—Human behavioral factors are fundamental to understanding, detecting and mitigating insider threats, but to date insufficiently represented in a formal ontology. We report on the design and development of an ontology that emphasizes individual and organizational sociotechnical factors, and incorporates technical indicators from previous work. We compare our ontology with previous research and describe use cases to demonstrate how the ontology may be applied. Our work advances current efforts toward development of a comprehensive knowledge base to support advanced reasoning for insider threat mitigation.**

Keywords— insider threat; sociotechnical indicators ontology; domain knowledge representation; SME knowledge modeling; human behavioral modeling; domain knowledge sharing

I. INTRODUCTION

Government and corporate organizations alike recognize the serious threat posed by insiders who seek to destroy, steal or leak confidential information, or act in ways that expose the organization to outside attacks. A widely accepted definition of the insider threat is "a current or former employee, contractor, or other business partner who has or had authorized access to an organization's network, system, or data and who intentionally (or unintentionally) exceeds or misuses that access to negatively affect the confidentiality, integrity, or availability of the organization's information or information systems" [1]. More generally, the insider threat may be defined in terms of internal risks to physical and human assets as well as organizational information. In light of recent government initiatives, Executive Order 13587 [2], and the National Insider Threat Policy that specifies minimum standards for establishing an insider threat program, there is increasing acknowledgment of the need to develop formal frameworks to represent and analyze vast amounts of data that may be collected by insider threat monitoring and mitigation systems. There is a notable lack of standards within the insider threat domain to assist in developing, describing, testing, and sharing techniques and methods for detecting and preventing insider threats [3]. The present research is directed toward a systematic and comprehensive representation of concepts in the insider threat domain that will support reasoning and threat assessment models.

Research reported here was supported under IARPA contract 2016-16031400006. The content is solely the responsibility of the authors and does not necessarily represent the official views of the U.S. Government.

II. BACKGROUND

Research on insider threat has sought to develop models and tools to identify individuals who pose increased insider threat risk. Most mitigation approaches focus more narrowly on (a) detecting unauthorized user activity and anomalous activity that may be malicious; and (b) preventing data exfiltration. Typical approaches attempt to prevent unauthorized access through the use of firewalls, passwords, and encryption. That is, they are primarily based on the tools and technology used to thwart external attacks. Unfortunately, these security measures will not prevent authorized access by an insider.

Because a key element of insider threat is a "trusted" perpetrator with authorized access to organizational assets, monitoring and analysis approaches should not only address suspicious host/network activities (identifying so-called technical indicators) but also seek to identify broader aspects of human behavior, motivation, and intent that may characterize malicious insider threats. Thus, as noted in [4], approaches should seek to identify attack-related behaviors that include deliberate markers, preparatory behaviors, correlated usage patterns, and even verbal behavior and personality traits, all of which can be pieced together to detect potential insider threats. While a number of researchers [5-9] recommend including behavioral indicators that may be accessible to organizations prior to an attack, tools and methods that incorporate formal representations of these human behavioral factors are rare (exceptions are models described in [10-12]). The research and operational security communities require a comprehensive knowledge base of technical and behavioral indicators to stimulate the development of more effective insider threat mitigation systems. Existing ontologies include a knowledge

base for technical indicators of insider threat [3][13] and a human factors oriented ontology for cybersecurity risk [14]; our work extends [13] and complements [14] by further specifying individual human and organizational sociotechnical factors.

III. OBJECTIVES

The objective of this research is to develop a formal representation of our current understanding of factors underlying insider threats, particularly relating to individual behavioral and psychological indicators and constructs reflecting organizational factors. The work to date complements and extends extant insider threat ontology frameworks. First, it adds substantial detail (depth) to existing insider threat ontology frameworks that focus on cyber/technical constructs. Second, it defines formal ontological representations of individual and organizational sociotechnical constructs, which are insufficiently represented in current ontological frameworks. The use of a formal, standardized language (ontology) for expressing knowledge about the insider threat domain facilitates information sharing across the insider threat research community and supports model development. A longer term goal is to inform the development of ontology-based reasoning systems and models to support insider threat detection and mitigation. Adopting and using more comprehensive, formal ontological representations will also facilitate the systematic construction of scenarios that may be used in exercising and validating insider threat detection models.

IV. APPROACH

Our approach consisted of (a) developing a hierarchical taxonomy for insider threat risk that can be applied generally to all types of organizations; and (b) migrating the taxonomy into a formal ontology for insider threat risk. Care was taken to compare our representation with existing frameworks (particularly the ontology developed by Carnegie Mellon University's Computer Emergency Response Team CERT [13]) to maximize consistency and interoperability among formulations across the research community. Our approach to ontology development seeks to extend the ontological framework by incorporating probabilistic methods to express and reason with uncertainty, i.e., this work will inform the development of a probabilistic ontology to support reasoning about insider threat risk.

A. Taxonomy Development

A well-defined taxonomy provides an initial hierarchy of domain concepts as a starting point for our insider threat ontology. The taxonomy is based on a systematic review, analysis and synthesis of existing research, case studies and guidelines that have been produced by the insider threat research community. Continually being expanded at the leaf nodes, the current taxonomy is 6-7 levels deep. There are 262 unique factors (leaf nodes) defined across the entire taxonomy: a total of 223 constructs defined for the individual factors and 39 for the organizational factors. Our class structure overall contains more than 350 constructs.

At the highest level we distinguish individual human factors from organizational factors. Individual human factors reflect behaviors, attitudes, personal issues, sociocultural or ideological factors, and various biographical factors that may indicate increased risk. The individual level also differentiates psychological traits from dynamic states, consistent with findings that these two constructs are reliably distinct despite their admitted overlap (e.g., [15-16]) and with the diverse body of psychological research that hinges on (e.g., [17-19]) or capitalizes on (e.g., [20-21]) that distinction. This detailed branch of the taxonomy reflects a substantial body of work by a diverse set of researchers and practitioners focusing on psychosocial factors underlying insider threats (e.g., [5], [7-9], [22-33]). The constructs that comprise this branch are listed in Table I, which shows the main factors (or classes) in column 1 and sub-classes (in *italics*) in column 2. Column 2 also includes illustrative descriptions or instances that reflect lower-level constructs (not exhaustive). In column 1 we also indicate a count of the total number of constructs defined at the leaf node level for each class, to provide a sense of the extensiveness of the taxonomy.

TABLE I. CONSTRUCTS COMPRISING INDIVIDUAL HUMAN FACTORS

Class[(a)]	Sub-Class and Instances
Concerning Behaviors *(140)*	*Boundary Violation* -- Concerning work habits, attendance issues, blurred personal/professional boundaries, threatening/intimidating behaviors, boundary probing, social engineering, minor policy violations, travel policy violations, unauthorized travel, unauthorized foreign travel, change in pattern of foreign travel, security violations
	Job Performance – Cyberloafing, negative evaluation
	Technical/Cyber Violation – Concerns about: authentication/ authorization, data access patterns, network patterns, data transfer patterns, command usage, data deletion/modification, suspicious communications
Life Narrative *(34)*	*Criminal Record* – Court records
	Financial Concerns – Lifestyle incongruities (unexplained affluence, etc.), risky financial profile (bankruptcy, large expenses-to-income ratio, bounced/bad checks, credit problems)
	Personal History – Demographics, employment, education background, major life events, health status, marital history, U.S. Immigration/citizenship status
Ideology *(9)*	*Disloyalty* – Behaviors or expressions of disloyalty to the organization or to the U.S. government *[2, 6]*
	Radical Beliefs – Radical political beliefs, radical religious
	Unusual Contact with Foreign Entity –Unreported contact with foreign nationals
Dynamic State *(14)*	*Affect* – Excessive anger/hostility, disengagement, mood swings
	Attitude – Lack of motivation, overly competitive, expresses feelings of disgruntlement with job, overly critical, resentful, defensive
Static Trait *(25)*	*Personality Dimensions* – Neuroticism, disagreeableness, low conscientiousness, excitement seeking, honesty-humility on six-factor personality scale
	Other Personality Traits – Characteristics associated with maliciousness or vulnerability to exploitation (Machiavellianism, narcissism, psychopathy, sadism, authoritarianism, social dominance orientation)
	Temperament – Various temperament issues that may be observed/reported by coworkers – Big ego, callousness, lack of empathy, lack of remorse, manipulativeness, rebelliousness, poor time management, preoccupation with power/grandiosity

[(a)] In parentheses is the total # of sub-classes or instances populated to date within the class

Organizational factors focus on organizational and management practices, policies, and work setting characteristics that influence worker satisfaction, attitudes, safety, or protection/vulnerabilities of assets. These factors have received much attention by organizations that publish best practices—indicating situations or conditions that contribute to an increased likelihood of insider threats within an organization. Although they may play a role in triggering malicious or unintentional insider threats, these factors have not generally been identified in insider threat ontologies to date. This branch of our taxonomy was constructed by consulting the broad and diverse literature on industrial/organizational psychology and human error research, including [34-36] and relevant discussion of these factors in the context of workplace violence and insider threat (e.g., [37-38]). Table II lists classes and sub-classes defined to date for organizational factors.

TABLE II. CONSTRUCTS COMPRISING ORGANIZATIONAL FACTORS

Class[a]	Sub-Classes
Security Practices *(14)*	Communication/training
	Policy clarity
	Hiring
	Monitoring
	Organizational justice
	Implementation of Security Controls
Communication Issues *(2)*	Inadequate procedures/directions
	Poor communications
Work Setting (Management Systems) *(6)*	Distractions
	Insufficient resources
	Poor management systems
	Job instability
	Lack of career advancement
	Poor physical work conditions
	Organizational changes
Work Planning and Control *(13)*	Job pressure/job stress
	Time factors/unrealistic time constraints
	Task difficulty
	Change in routine
	Heavy or prolonged workload
	Insufficient workload
	Conflict of work roles
	Work role ambiguity
	Lack of autonomy
	Lack of decision-making power
	Irregular timing of work shifts
	Extended working hours
	Lack of breaks
Mitigating Factors *(4)*	Flexible work schedule
	Employee Assistance Plan
	Effective staff training and awareness
	Reporting mechanism

[a] In parentheses is the total # sub-classes or instances populated to date within the class

B. Ontology Development Approach

To date, insider threat ontology development has focused primarily on technical factors (e.g., [13]). In contrast, our approach is grounded in an extended problem space that includes methods, motivation, psychology, and circumstances of human behavior. As noted by previous authors (e.g., [13]), behavioral aspects of insider threat can be an extraordinarily complex domain to model. There are many overlapping concepts (e.g., state and trait anger), many providing little meaning in isolation (e.g., surfing the web vs. surfing the web instead of working). Our task has been to contextualize behaviors with related concepts (e.g., underlying motivations and personality traits) that allow the cataloging of information pertaining to both the insider threat incident and the insider. Through this catalogue of information, researchers and organizations can index cases and gain further insight into common attack vectors driven by human behavior. Our ontology extends previous work [3][13][14] in two ways: (a) adding more detail to the technical indicator branch of the ontology and (b) adding material focusing on individual behavioral and organizational factors.

Our approach is to migrate our taxonomy into a formal ontology expressed in the popular OWL-DL ontology language. OWL-DL balances expressiveness (ability to represent many kinds of domain entities and relationships), computational properties (conclusions are guaranteed to be computable in finite time), and functionality for drawing inferences from asserted facts. Enumeration of (potentially hundreds of) *Competency Questions* (*CQs*) for our ontology serves as a requirements specification as well as a means of testing the ontology implementation. An example of a simple *CQ* is "What are the components of class *Attitude*?" A more complex *CQ* is "What factors are associated with the observables *attendance problems*, *unauthorized personal use of work computer*, and *hostile*? The *CQs* may be evaluated using SPARQL queries. Our OWL-DL implementation will enable automated inferences about class relationships. For example, from the assertion that an individual belongs to class *Aggressive* and class *Manipulative*, the reasoning engine can infer that the individual fulfills the membership conditions of class *Threat*.

V. ONTOLOGY IMPLEMENTATION

A. Ontology Methods

Following widely recognized guidelines for ontology development [39], we used the Methontology ontology engineering methodology [40], which enables construction at the conceptual level and allows for development, re-use, or re-engineering of existing ontologies. In the *Specification* phase we defined the purpose of the ontology, its intended uses and its end users. In the *Conceptualization* phase we structured the domain knowledge into meaningful graphical models. In the *Formalization* phase we represented our conceptual models as a formal or semi-computable model. The *Implementation* phase supports the ontology development in the Web Ontology Language (OWL). Updates and corrections take place in the *Maintenance* phase. Our development also included supporting Methontology activities of *Knowledge Acquisition*, *Evaluation* (verification and validation that the ontology represents the domain), *Integration* (reuse of other available ontologies), *Documentation*, and *Configuration management*. We also adopted IDEF5 methods in conceptualization and formalization phases to acquire knowledge and develop graphical knowledge representation models. We implemented our taxonomy using an off-the-shelf ontology development tool (Protégé).

By default, the Protégé tool does not assume that classes are mutually exclusive. This is useful when concepts are most meaningful in combination. For example, high absenteeism, a weak indicator by itself, is made stronger in association with

other concerning factors [32], but the risk is mitigated when associated with documented illness, vacation or maternity leave. As another example, relaxation of the assumption of mutual exclusivity is especially useful when considering various correlated psychological or personality characteristics such as those defined in the Five Factor Model (FFM) of personality traits [41]. There are numerous well-supported relationships between dimensions of personality and various types of counterproductive work behavior [28].

B. Description of the Ontology Classes

We began by formalizing the hierarchy of concepts provided by the taxonomy discussed in Section IV-A, and translating the hierarchy into parent-child relationships of classes in our ontology. Classes represent objects with similar structure and properties Classes are arranged hierarchically; those without further subcategories are termed leaf nodes. Individuals in the ontology represent instances of classes. Class relationships other than parent-child are derived from the research literature, available material on insider threat cases, and the experience and judgment of subject-matter experts within the development team. As reuse of previous knowledge models is a key advantage of ontologies and an encouraged practice in ontology engineering, we included as much information from previous work as possible, especially the recent ontology developed by CERT [3][13]. In particular, the *Actor*, *Asset*, *Action*, *Event*, *Temporal Thing* and *Information* class structures are adopted in total. Selected classes from the Unified Cyber Security Ontology [42] were also incorporated into our ontology. For example the idea of "Consequence" class is adopted by our ontology but renamed to *Outcome* class since this terminology is more consistent with the insider threat cases scenario template used by CERT. The concepts of *Vulnerability* (e.g., [6]) and *Catalyst/Trigger* events (e.g., [43-44]) are also formalized as classes in our ontology. To capture the temporal information involved in insider threat cases, we imported the *Temporal Interval* class from the CERT ontology.

Figs. 1-3 show the hierarchy of classes in our ontology, as implemented in the Protégé tool. The ontology is derived from the extensive taxonomy described in Section IV-A. Due to space constraints we depict only selected classes with detail restricted to the 4th level of the hierarchy. A comparison of Tables I and II with Figs. 1-3, shows how the class hierarchy in the ontology represents the organization of domain concepts in the taxonomy. Fig. 1 shows how the ontology accounts for both malicious and non-malicious (unintentional) insider threats. Importantly, we distinguish between actions performed by employees (as insiders) and actions performed by organizations (which may, for example, include poor institutional policies and/or security practices as well as inadequate or exacerbating responses to potential threats). At the same root level we also include classes such as *Industry*, *Insider Threat Risk*, *Effect*, *Location* and *Outcome* as attributes of the organization. *Industry* may account for differences in organizational rules,

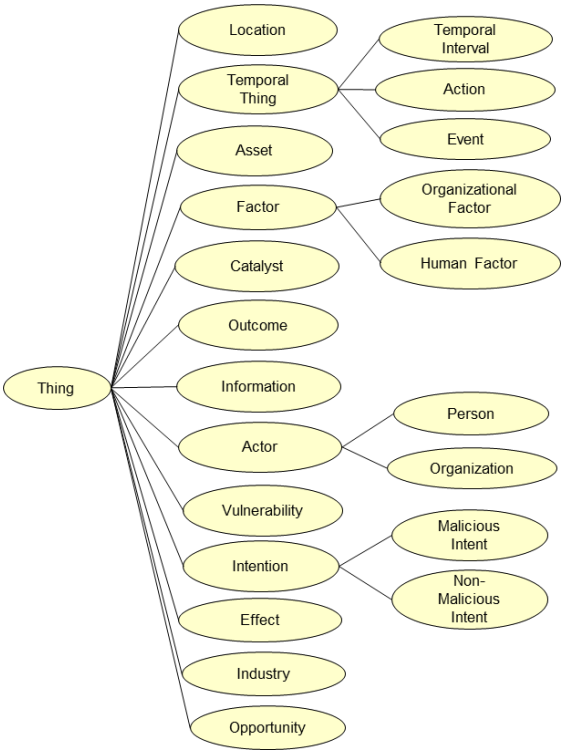


Fig. 1. Top-Level Classes

regulations and policies that differ across industry sectors. The *Effect* class captures information about the impact of the insider criminal activity on the organization(s), for example the action of injecting a virus into an enterprise network can induce a malfunction in other workstations on the network and/or a full network shutdown. The concept of the consequences of an attack is captured by the *Outcome* class, for example the shutdown of the network has an outcome of a halt of organization's operations and thousands of dollars of loss. The *Location* class encapsulates geographic information about the source of an attack. The *Insider Threat Risk* class captures the threat level that would be associated with the individuals of the *Actor* class based on the inference performed over the ontology.

Fig. 2 expands the *Human Factor* node of Fig. 1, and Fig. 3 expands the *Organizational Factor* node. Inspection of the human psychosocial factors in Fig. 2 reveal classes (and associated sub-classes) that correspond to elements of the taxonomy. Acknowledging the Capability-Motive-Opportunity (CMO) model (e.g., [4]), which postulates that the perpetrator of an attack must have the capability, motive, and opportunity to commit the attack, we include these constructs as classes in the ontology. Full implementation of CMO constructs is deferred for future efforts to define relationships among these classes.

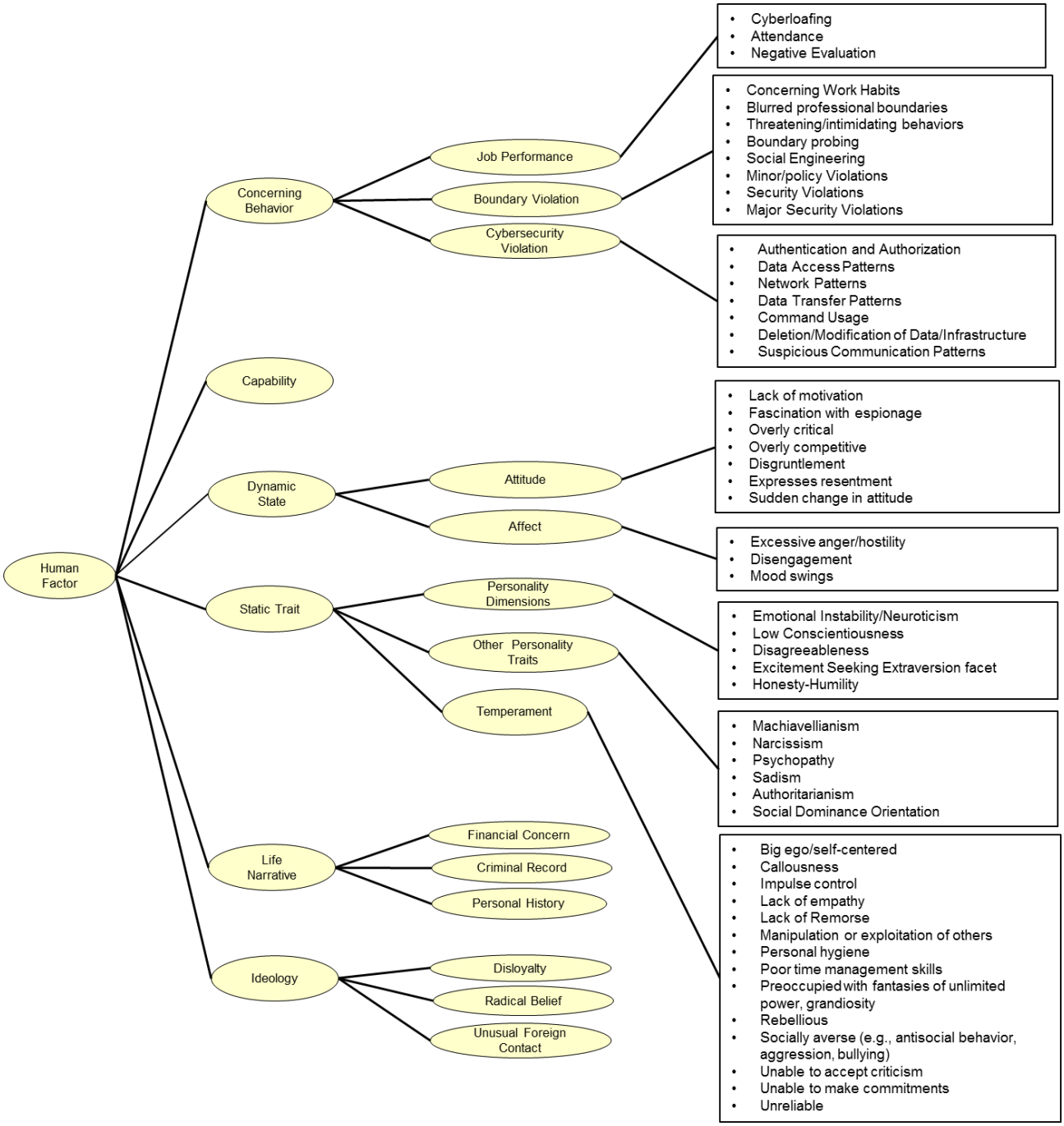


Fig. 2. *Human Factor* Classes (Lower level details for *Life Narrative* and *Ideology* classes are not shown due to space constraints)

The capability to conduct an attack is in part dependent on an individual's knowledge/skills/abilities that are represented in certain human behavioral factors (cf [14]), particularly the *Biographical Data* subclass within the *Life Narrative* factors class. Motive (or motivation) may be represented within the *Intention* class (and its *malicious* or *non-malicious* subclasses) in Fig 1; it is also related to psychological characteristics or predispositions such as *Static Traits*, *Dynamic States*, and *Life-Narrative* factors (e.g., financial or health problems that may act as stressors)—which are sub-classes of the *Human Factor* class (see Fig. 2)—as well as *Organizational Factors* (Fig. 3) that may act as stressors or triggers that can motivate an attack.

The sub-class *Concerning Behavior,* within the *Human Factor* class, contains a large set of individual actions that includes the subcategories *Job Performance, Boundary Violation*, and *Cyber Security Violation*. These in turn are broken down into more granular, lower-level constructs (shown in boxes); not shown are even lower levels of the hierarchy and individuals representing instances of the classes.

The initial structure of the ontology grew out of the detailed taxonomic structure that we developed based on subject-matter expertise and our analysis/synthesis of research literature and numerous case studies. A more robust and richer representation has been informed by exploring complex relationships among constructs (e.g., classes, sub-classes, instances) spread across multiple branches of the hierarchy. As a simple example, the ontology recognizes that different types of attack are identified from their relationships with certain aspects of the cyber/technical exploit (e.g., exfiltration requires certain actions performed on sensitive information, such as saving to external media, printing, emailing, uploading to the cloud, etc.). A more complex example may be considered in using the CMO model (mentioned above) to reason about insider risk. By incorporating knowledge of relationships among detected behaviors, individual behavioral factors, and organizational factors, the ontology allows reasoning about the risk associated

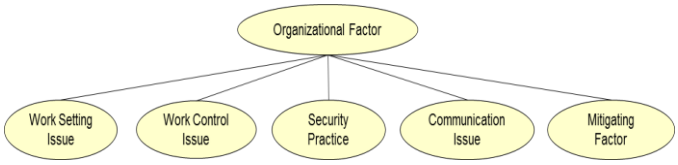


Fig. 3. *Organizational Factor* Class

with detected behaviors in the context of possible motives, capabilities, and opportunity. Relationships and gaps (missing elements in classes) were further identified by exercising the knowledge base using known or fictitious use cases.

C. Use Case and Application

Use cases help to verify the comprehensiveness of the knowledge representation and to identify missing or ill-defined classes and relationships. In this section, we demonstrate the application of the ontology to use cases that include human behavioral factors and organization factors as well as cyber/technical indicators. In the scenarios described, we use [brackets] to identify significant indicators with actions described in the scenario.

Use Case #1 (see small text box) describes a simple cyber-related insider threat incident. Use Case #2 (see large text box), which entirely subsumes the contextual and technical information regarding the insider threat incident described in the first use case, injects additional human behavioral factors.

Use Case #1

John **[PERSON: Insider X]** *is a long-time system administrator* **[LIFE NARRATIVE: PERS HISTORY] [CAPABILITY]** *with access to sensitive and classified information* **[OPPORTUNITY]** *in a company that performs government-sponsored R&D* **[ORGANIZATION: VICTIM ORGANIZATION]**.

John uses his personal web-based email account from his work computer to communicate with prospective employers **[DIGITAL ACTION: EMAIL ACTION]**. *Then he uses his administrative privileges to access some sensitive intellectual property information* **[BUSINESS INFORMATION: INTELLECTUAL PROPERTY]** *that will be of interest to a competitor. John saves these files to his computer* **[COMPUTER ASSET: WORK PC]** *and copies the files to a thumb drive* **[CONCERNING BEHAVIOR: TECH/CYBER VIOLATION–DIGITAL ACTION/COPY ACTION] [PHYSICAL ASSET: USB DRIVE]**, *which he then sneaks out of the office with the intention of using the information to leverage a job offer with a competitor* **[THEFT EVENT: DATA THEFT]**. *Subsequently John resigns and accepts a job offer from a competitor.*

It is evident that Use Case #1 lacks substantial contextual information described in Use Case #2 regarding possible contributing or mitigating factors, relevant personal predispositions, or concerning behaviors that may be associated with this individual's insider threat risk. Fig. 4 is a concept map depicting Use Case #2, showing all the behavioral and technical concepts and their associated relations. The dashed

Use Case #2

John **[PERSON: Insider X]** *is a long-time system administrator* **[LIFE NARRATIVE: PERS HISTORY] [CAPABILITY]** *with access to sensitive and classified information* **[OPPORTUNITY]** *in a company that performs government-sponsored R&D* **[ORGANIZATION: VICTIM ORGANIZATION]**. *The following input was recorded in his personnel file: (1) One colleague states that John discounts the opinions of colleagues and he becomes hostile when colleagues discuss and critique his ideas* **[STATIC TRAIT: TEMPERAMENT: RESISTS CRITICISM] [DYNAMIC STATE: AFFECT—HOSTILE].** *(2) A different colleague states that John seeks to control all aspects of a project and often insists on dominating the conversation about project tasks and approach* **[STATIC TRAIT: OTHER PERSONALITY DIMENSIONS—AUTHORITARIANISM]**. *(3) His manager corroborates these inputs and adds that John tends to become argumentative and irritated, and defensively cites his superior knowledge of industry best practices when others criticize his rigid protocols* **[DYNAMIC STATE: AFFECT–HOSTILE] [STATIC TRAIT: TEMPERAMENT—BIG EGO]**. *Staff development/performance review assessment includes criticism by colleagues that portions of his protocols are idiosyncratic with weak rationale, and that his rigid protocols have impacted company projects* **[CONCERNING BEHAVIORS: JOB PERF—NEGATIVE PERF EVALUATION].**

John was passed over for a promotion to manage a new, prestigious project **[LIFE NARRATIVE: PERS HISTORY: EMPLOYMENT–PASSED OVER FOR PROMOTION].** *He files a complaint with HR claiming unfair treatment and his manager, compelled to meet with him, comes away with the impression that John still harbors resentment over not being promoted. John's most recent evaluation cited a decline in performance* **[CONCERNING BEHAVIORS: JOB PERF—NEGATIVE PERF EVALUATION]**; *since being denied the promotion his attitude has been increasingly disgruntled* **[DYNAMIC STATE: ATTITUDE—DISGRUNTLEMENT]**; *and that there were multiple complaints from coworkers about frequent tardiness* **[CONCERNING BEHAVIORS: BOUNDARY VIOLATION—ATTENDANCE]**. The attendance problem led to *a formal, written warning* **[CONCERNING BEHAVIORS: BOUNDARY VIOLATION–POLICY VIOLATION].** *After getting the warning, John talks to his manager and loses his cool—storming out of the office* **[DYNAMIC STATE: AFFECT–HOSTILE].** *A colleague hears John's outburst and tells the manager about John's recent marital separation to provide some context to Johns behavior* **[LIFE NARRATIVE: PERS HISTORY—MAJOR LIFE EVENTS/RECENT CHANGE IN MARITAL STATUS (MARITAL SEPARATION)**]. *The incident prompts the manager to contact the company Security Office. The Security Office checks the local court records to learn that three weeks ago, John was arrested for allegedly driving under the influence (his first contact with the criminal justice system)* **[LIFE NARRATIVE: CRIMINAL RECORD—DUI]**.

Faced with these job and personal stressors, John begins to seek work with a competitor. John contacts a competitor to see if they are interested in him and in proprietary information he can provide. To avoid being noticed, John carries out email dialogue with the competitor by logging into his personal Yahoo web mail account from his work computer **[CONCERNING BEHAVIORS: JOB PERFORMANCE—CYBERLOAFING]**. *Next, John carries out the insider threat attack and resigns, as described in second paragraph of Use Case #1.*

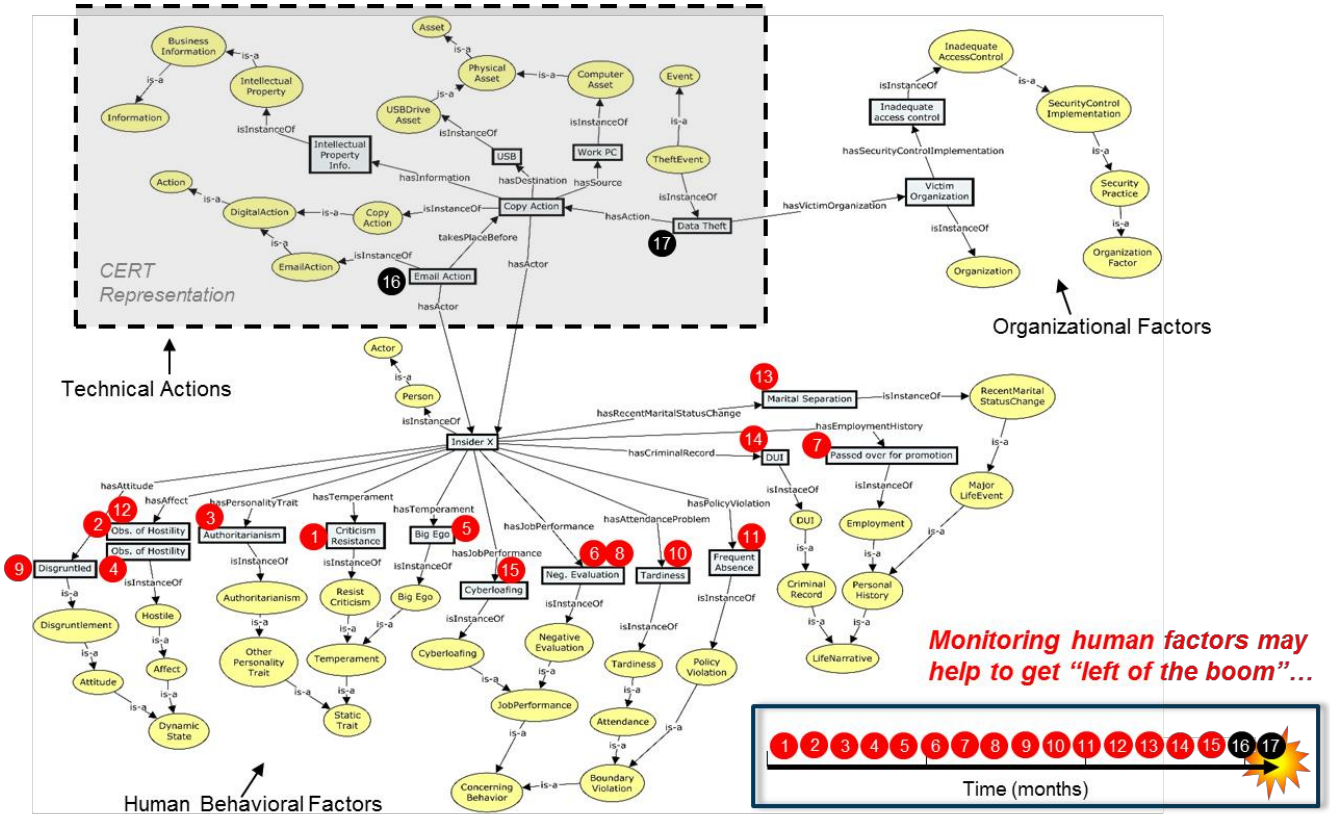


Fig. 4. Concept map representation of Use Case #2 (Case #1 is within dashed box). Not all concepts and relations are shown due to space limitations.

box in Fig. 4 represents Use Case #1 (due to space limitations, not all details are shown). In a real scenario, detecting concerning behaviors or other factors may require multiple factors to meet threshold requirements for alerts—these are not described or represented here due to space constraints. Events depicted in the use case scenarios are numbered chronologically. Shown in the lower right side of the figure is a timeline (spanning several months for illustrative purposes) suggesting that monitoring of sociotechnical factors may help achieve proactive mitigation goals (getting "left of the boom").

VI. Comparison with Related Work

The focus of our effort is to express and represent individual and organizational sociotechnical factors in an ontological characterization of insider threat risk (e.g., [13-14]). Our ontology provides a more robust, richer description of not only the nature of the attack but also possible contributing factors that more fully describe the insider threat to the organization. CERT [13] began with a database of insider threat case descriptions. The information framework underlying this database informed the vocabulary in the ontology. Namely, organizations grant access to persons that perpetrate events that harm the organization. Persons and Organizations are the actors in the CERT ontology, and their actions culminate in events (i.e., insider threat incidents). Instead of a focus on events, our ontology focuses on the insider. Our taxonomy and ontology are based on theories and models of insider threat in the literature that incorporate human behavioral as well as technical indicators of threat (e.g., [10-12]). While the current CERT ontology only describes technical/cyber events, our ontology also includes non-technical or sociotechnical constructs that reflect actions and psychosocial indicators of persons of interest. As a specific example, consider the class *Concerning Behaviors*. A concerning behavior such as "Leaving a classified security container unlocked and unattended" can be described using two concepts in the CERT ontology: an Asset (e.g., Classified file) and an Action (e.g., Unlock). However, this may not be the focal event, or a precipitating event, in a case description, and there may be other related contributing factors. For example, a previous condition (e.g., organizational reduction in force/layoffs) or individual predispositions (e.g., personality traits, personal stress) may lead to actions that reflect a lack of diligence or motivation in an actor who later commits an act of insider threat (these contributing factors are in part identified in the cybersecurity human factors ontology (HUFO) by [14]. The CERT ontology, in particular, does not connect these behavioral constructs to technical/cyber actions that comprise the actual exploit.

At a basic level, the *Factor* class, which contains much of the vocabulary in our ontology, can be placed alongside *Assets* in the CERT ontology. Both are non-temporal classes that a person can possess (i.e., *Things*). We integrated the two ontologies and eliminated duplications. All CERT ontology classes were incorporated in this way. There are, however, stark differences between the extent and scope of the CERT ontology and our ontology. The CERT ontology contains a standardized and well-defined vocabulary for describing the actions of insider threats. It contains 31 actions (e.g., Copy), along with six action modifiers (e.g., Suspicious), organized under four major

classes to describe digital, financial, and job-related insider threat behavior. These actions can be taken on 26 assets (e.g., USB drive) in three major categories (i.e., Physical, Financial, and Digital) and/or 16 types of information (e.g., Password) organized in seven major categories (i.e., National Security, Technology, Financial, Medical, Classified, Business, and Uniquely Identifiable). Eleven focal events are also captured as classes in the ontology (e.g., Theft), for a total of 125 constructs within their class structure. In contrast to the CERT ontology, our framework is broader and deeper. In addition to containing these constructs, our ontology represents a knowledge base that is six to seven layers deep, comprising a total of over 350 constructs. In sum, we have greatly expanded the CERT ontology by adding classes representing human behavioral and organizational factors of insider threat.

While not specifically addressing insider threat, the cybersecurity HUFO presented by [14], which focuses on trust, is similar to and largely compatible with our ontology; it defines roughly 48 human factors classes that address characteristics such as motivation, integrity, rationality, benevolence, personality, ideology, ethics, and risk posture, as well as knowledge, skills and abilities. In comparison, our ontology probes several levels deeper than the HUFO ontology. Further work is planned to integrate relevant features of these ontologies.

VII. Conclusions and Future Work

Our work addresses two major challenges. First, due to the large number of concepts and their complex interrelationships, the insider threat domain is cumbersome to model. Second, there is a need to establish a common terminology and shared understanding of the complex insider threat domain. We used an exhaustive approach that incorporates into our taxonomy most of the concepts we have encountered in the insider threat literature. We then developed a mapping that transforms the taxonomy into an ontology, and added relationships to the ontology to produce a formal representation of concepts and their interrelationships. By synthesizing the contributions of a diverse set of experts, we developed a knowledge representation that more fully characterizes insider threat indicators—from the perspective of human behavior as well as cyber/technical indicators—and that can be made available in a shareable knowledge base to facilitate reuse and collaboration.

Beyond its immediate use in providing a common, shareable knowledge base of insider threat problem space constructs, the present research will help to advance efforts to model and mitigate insider threats. Informed by extant research on human and organizational factors associated with insider threats, the constructs and indicators represented in the present ontology can be used to develop models to assess individual risk and organizational vulnerability, as well as to inform operational risk management practices. In addition, by specifying a more comprehensive knowledge base, our ontology facilitates the generation of diverse scenarios for use in red teaming and testing of more holistic insider threat models. Finally, the knowledge base provided here may have further operational impact by informing the structure of data to be captured by enterprises for effective insider threat monitoring and analysis.

A brief discussion of some limitations of the research reported here may be useful in interpreting progress to date as well as motivating future work. First, our choice to define a taxonomy as a foundation for the ontology meant that the initial structure only specified hierarchical parent-child relationships among constructs. Other relationships were then defined as part of the process of transforming the taxonomy into an ontology. Because our primary interest (and recognized need in modeling insider threats) was to incorporate sociotechnical factors that have been suggested in research literature, there was also an inherent limitation in the ability to specify robust axioms that reflect more complex relationships among constructs. Ultimately this more complete specification will be required to support inferences about classes and individuals. There is a tradeoff between implementing the asserted classes and individuals versus the inferred constructs. While some of the classes in our ontology are defined by certain inference rules and axioms (e.g., the class *Capability* categorizes instances based on specified rules), much more work is needed to more fully specify relationships that will ultimately be required to support inferences about insider threat risks. A second limitation is that, while the current ontology has captured salient constructs in the literature, there are certainly more constructs that can and should be added to the ontology. Research should continue the process of encapsulating the entirety of constructs related to insider threat. We are continually populating the individual and organizational classes of ontology with relevant instances (informed by use cases); we plan to further develop the *Capabilities* and *Opportunities* classes and associated relationships, building upon recent related work [14]. Future research should also focus on addressing the need to represent temporal relationships among constructs.

We use the present forum and others to share these results with the research community. We also plan to extend our ontology into a probabilistic ontology by incorporating information about uncertainty in the insider threat domain. The resulting probabilistic ontology will support reasoning under uncertainty [45]. Probabilistic ontologies combine semantically rich representations that support interoperability and automated reasoning with mathematically well-founded uncertainty management. Advancing research and development of probabilistic ontologies for insider threats will facilitate modeling and tool development. Our ontology provides a rich foundation for logical and probabilistic inferences necessary for protection against insider attacks.

References

[1] D. M. Cappelli, A. P. Moore, and R. F. Trzeciak, *The CERT guide to insider threats: How to prevent, detect, and respond to information technology crimes (theft, sabotage, fraud)*. Addison-Wesley, 2012.

[2] The White House. Executive Order 13587—Structural Reforms to Improve the Security of Classified Networks and the Responsible Sharing and Safeguarding of Classified Information, October 2011. http://www.whitehouse.gov/the-press-office/2011/10/07/executive-order-structural-reforms-improve-security-classified-networks-

[3] D. L. Costa, M. Collins, J. S. Perl, J. M. Albrethsen, J.G. Silowash, and D. Spooner. (2014). An Ontology for Insider Threat Indicators. In K. B. Laskey, I. Emmons and P C.G. Costa (Eds.), *Proceedings of the Ninth Conference on Semantic Technologies for Intelligence, Defense, and Security* (STIDS 2014), 2014, 48–53.

[4] E. E. Schultz, "A framework for understanding and predicting insider attacks." *Computers & Security*, 2002, vol. 21, 526–531.

[5] E. D. Shaw, J. M. Post, and K. G. Ruby, "Inside the mind of the insider." *Security Management*, 1999, vol 43 (12), 34-42.

[6] M. R. Randazzo, M. M. Keeney, E. F. Kowalski, D. M. Cappelli, and A. P. Moore. I*nsider threat study: illicit cyber activity in the banking and financial sector*. Carnegie-Mellon University. Software Engineering Institute. CMU/SEI-2004-TR-021, 2012.

[7] S. R. Band, D. M. Cappelli, L. F. Fischer, A. P. Moore, E. D. Shaw, and R. F. Trzeciak. *Comparing insider IT sabotage and espionage: a model-based analysis*. Carnegie-Mellon University. Software Engineering Institute. CERT Coordination Center. CMU/SEI-2006-TR-026, 2006.

[8] F. L. Greitzer, L. J. Kangas, C. F. Noonan, C. R. Brown, and T. Ferryman. Psychosocial modeling of insider threat risk based on behavioral and word use analysis. *e-Service Journal*, 2013, 9(1), 106-138. http://www.jstor.org/stable/10.2979/eservicej.9.1.106

[9] M. Maasberg, J. Warren, and N. L.Beebe. The dark side of the insider: Detecting the insider threat through examination of dark triad personality traits. IEEE. *48th Hawaii International Conference on System Sciences,* 2015*,* 3518-3526. DOI 10.1109/HICSS.2015.423

[10] F. L. Greitzer and R. E. Hohimer. "Modeling Human Behavior to Anticipate Insider Attacks." *Journal of Strategic Security,* 2011*, 4(2)*:25-48. http://scholarcommons.usf.edu/jss/vol4/iss2/

[11] R. E. Hohimer, F. L. Greitzer, C. F. Noonan, and J. D. Strasburg. "CHAMPION: Intelligent Hierarchical Reasoning Agents for Enhanced Decision Support." In *Semantic Technology for Intelligence, Defense, and Security (STIDS 2011*). 2011, 36-43

[12] E. T. Axelrad, P. J. Sticha, O. Brdiczka, and J. Shen, "A Bayesian network model for predicting insider threats." IEEE SPW *Workshop on Research for Insider Threat (WRIT), San Francisco, CA,* 2013, 82-89.

[13] D. L.Costa, M. J. Albrethsen, M. L. Collins, S. J. Perl, G. J. Silowash, and D. L. Spooner. *An Insider Threat Indicator Ontology*. TECHNICAL REPORT CMU/SEI-2016-TR-007. Pittsburgh, PA: SEI, 2016.

[14] A.Oltramari, D. H. Henshel, M. Cains, and B. Hoffman. "Towards a human factors ontology for cyber security." In *Semantic Technology for Intelligence, Defense, and Security (STIDS 2015*). 2015, 26-33.

[15] W. E. Chaplin, O. P. John, and L. R. Goldberg. "Conceptions of states and traits: Dimensional attributes with ideals as prototypes." *Journal of Personality and Social Psychology*, 1988, *54(4)*, 541-557.

[16] R. Steyer, A. Mayer, C. Geiser, and D. A. Cole. "A theory of states and traits—Revised." *Annual Review of Clinical Psychology*, 2015, *11*, 71-98.

[17] S. C. Roesch, A. A. Aldridge, S. N. Stocking, F. Villodas, Q. Leung, C. E. Bartley, and L. J. Black. "Multilevel factor analysis and structural equation modeling of daily diary coping data: Modeling trait and state variation. *Multivariate Behavioral Research,* 2010*, 45(5),* 767-789.

[18] L. Van Gelder and R. E. De Vries. "Traits and states at work: Lure, risk and personality as predictors of occupational crime." *Psychology, Crime & Law*, 2016, *22(7)*, 701-720. DOI 10.1080/1068316X.2016. 1174863

[19] D. F. Gro¨s, L. J. Simms, M. M. Antony, and R. E. McCabe. "Psychometric properties of the State–Trait Inventory for Cognitive and Somatic Anxiety (STICSA): Comparison to the State–Trait Anxiety Inventory (STAI)." *Psychological Assessment*. 2007, *19(4)*, 369–381.

[20] K. S. Douglas, S. D. Hart, C. D. Webster, and H. Belfrage. *HCR-20V3: Assessing risk of violence – User guide.* 2013. Burnaby, Canada: Mental Health, Law, and Policy Institute, Simon Fraser University.

[21] J. R. Meloy, S. G. White, and S. Hart. "Workplace assessment of targeted violence risk: The development and reliability of the WAVR-21". *Journal of Forensic Sciences*, 2013, *58(5)*, 1353-1358.

[22] E. D. Shaw and L. F. Fischer. Ten Tales of Betrayal: The Threat to Corporate Infrastructures by Information Technology Insiders. Report 1—Overview and General Observations. Technical Report 05-04, April 2005. Monterey, CA: Defense Personnel Security Research Center.

[23] M. Gelles, M. Exploring the mind of the spy. In Online *Employees' Guide to Security Responsibilities: Treason 101*. 2005. Retrieved from Texas A&M University Research Foundation website: http://www.dss.mil/search-dir/training/csg/security/Treason/Mind.htm

[24] J. L. Krofcheck and M. G. Gelles. *Behavioral Consultation in Personnel Security: Training and Reference Manual for Personnel Security Professionals.* Yarrow and Associates, 2005.

[25] D. Bulling, M. Scalora, R Borum, J Panuzio, and A Donica. *Behavioral science guidelines for assessing insider threats*. Publications of the University of Nebraska Public Policy Center. Paper 37. 2008. http://digitalcommons.unl.edu/publicpolicypublications/37

[26] D. B. Parker. *Fighting computer crime: A new framework for protecting information*. New York, NY: John Wiley & Sons, Inc., 1998.

[27] F. L. Greitzer, A. P. Moore, D. M. Cappelli, D. H. Andrews, L. A. Carroll, and T. D. Hull. Combating the insider threat. (2008). *IEEE Security & Privacy*, January/February 2008, 61-64.

[28] E. D. Shaw, L. F. Fischer, and A. E. Rose. *Insider risk evaluation and audit* (No. TR-09-02). Monterey, CA: Defense Personnel Security Research Center, 2009.

[29] B. Zadeh and F. L. Greitzer. "Motivation and Capability Modeling for Threat Anticipation." *OSD Human Social Culture Behavior (HSCB) Modeling Program Conference.* Chantilly, VA, 5-7 August 2009.

[30] F. L. Greitzer and D. A. Frincke. D.A. "Combining traditional cyber security audit data with psychosocial data: towards predictive modeling for insider threat," in *Insider Threats in Cyber Security*. vol. 49, C. W. Probst*, et al.*, Eds., Springer US, 2010, 85–114.

[31] F. L. Greitzer, L. J. Kangas, C. F. Noonan, and A. Dalton. *Identifying at-risk employees: A behavioral model for predicting potential insider threats.* PNNL-19665, Richland, WA: Pacific NW National Laboratory, 2010. http://www.pnl.gov/main/publications/external/technical_reports/PNNL-19665.pdf.

[32] F. L. Greitzer, L. J. Kangas, C. F. Noonan, A. Dalton, and R. E. Hohimer. "Identifying at-risk employees: a behavioral model for predicting potential insider threats." *Hawaii International Conference on System Sciences*. Maui, HI, Jan 4-7, 2012.

[33] Software Engineering Institute (SEI). *Analytic approaches to detect insider threats*. White Paper, SEI, December 9, 2015. http://resources.sei.cmu.edu/asset_files/WhitePaper/2015_019_001_451069.pdf

[34] S. Dekker. *The field guide to human error investigations*. Burlington, VT: Ashgate, 2002.

[35] D. J. Pond and K. R. Leifheit. "End of an error." *Security Management*, 2003, *47(5)*. 113–117.

[36] D. J. Pond and F. L. Greitzer. "Error-based accidents and security incidents in nuclear materials management." *Institute of Nuclear Materials Management 46th Annual Meeting*, Phoenix, AZ, 2005. http://www.osti.gov/scitech/biblio/966022

[37] R. Baron and J. Neuman. Workplace violence and workplace aggression: Evidence on their relative frequency and potential causes. *Aggressive Behavior*, 1996, vol. 22, no. 3, 161–173.

[38] F. L. Greitzer, J. Strozer, S. Cohen, J. Bergey, J. Cowley, A. Moore, and D. Mundie. "Unintentional insider threat: contributing factors, observables, and mitigation strategies." *47th Hawaii International Conference on Systems Sciences (HICSS-47)*, Big Island, Hawaii, 2014.

[39] N. F. Noy and D. L. McGuinness. *Ontology Development 101: A Guide to Creating Your First Ontology.* (SMI-2001-0880 (also available as KSL Technical Report KSL-01-05)) 2001

[40] M. Fernández-López, and A. Gómez-Pérez. Overview and analysis of methodologies for building ontologies. *The Knowledge Engineering Review, 2002*, *17(2)*, 129– 156.

[41] L. R. Goldberg, "The structure of phenotypic personality traits." *American Psychologist,* 1993, vol. 48, 26-34.

[42] Z. Syed., A. Padia., T. Finin, L. Mathews, and A. Joshi. *UCO: A Unified Cybersecurity Ontology* (Tech.). Baltimore, MD, 2016.

[43] Claycomb, W. R., Huth, C. L., Flynn, L., McIntire, D. M., Lewellen, T. B., & Center, C. I. T. (2012). Chronological Examination of Insider Threat Sabotage: Preliminary Observations. JoWUA, 3(4), 4-20.

[44] J. R. C. Nurse, O. Buckley, P.A. Legg, M. Goldsmith, S. Creese, G. R. T. Wright, and M. Whitty. *Understanding Insider Threat: A Framework for Characterising Attacks,* 2014.

[45] R. N. Carvalho, K. B. Laskey, and P. C. Costa. "Uncertainty modeling process for semantic technology." *PeerJ Computer Science, 2016,* 2:e77 https://doi.org/10.7717/peerj-cs.77

An Extended Maritime Domain Awareness Probabilistic Ontology Derived from Human-aided Multi-Entity Bayesian Networks Learning

Cheol Young Park, Kathryn Blackmond Laskey, Paulo C. G. Costa
The Sensor Fusion Lab & Center of Excellence in C4I
George Mason University, MS 4B5
Fairfax, VA 22030-4444 U.S.A.
cparkf@masonlive.gmu.edu, [klaskey, pcosta]@gmu.edu

Abstract— **Ontologies have been commonly associated with representing a domain using deterministic information. Probabilistic Ontologies extend this capability by incorporating formal probabilistic semantics. PR-OWL is a language that extends OWL with semantics based on Multi-Entity Bayesian Networks (MEBN), a Bayesian probabilistic logic. Developing probabilistic ontologies can be greatly facilitated by the use of a modeling framework such as the Uncertainty Modeling Process for Semantic Technology (UMP-ST). An example of using UMP-ST was the development of a probabilistic ontology to support PROGNOS (PRobabilistic OntoloGies for Net-Centric Operational Systems), a system that supports Maritime Domain Awareness (MDA). The PROGNOS probabilistic ontology provides semantically aware uncertainty management to support fusion of heterogeneous input and probabilistic assessment of situations to improve MDA. However, manually developing and maintaining a probabilistic ontology is a labor-intensive and insufficiently agile process. Greater automation through a combination of reference models and machine learning methods may enhance agility in probabilistic situation awareness (PSAW) systems. For this reason, a process for Human-aided MEBN Learning in PSAW (HMLP) was suggested. In previous work, we used UMP-ST to develop the PROGNOS probabilistic ontology. This paper presents an extended PROGNOS probabilistic ontology developed using HMLP. The contribution of this research is to introduce the extended PROGNOS probabilistic ontology and present a comparison between two processes (UMP-ST and HMLP).**

Keywords—Probabilistic Ontology; Maritime Domain Awareness; Predictive Situation Awareness; Bayesian Networks; Multi-Entity Bayesian Networks; Uncertainty Modeling Process for Semantic Technology; Human-aided Machine Learning

I. INTRODUCTION

In information science, integration of heterogeneous, distributed, and unstructured information is a difficult and complex challenge. A major issue is ensuring information compatibility, for which ontologies have become a standard solution [18]. Traditional ontologies are limited to deterministic knowledge. Probabilistic Ontologies (POs) move beyond this limitation by incorporating formal probabilistic semantics. Probabilistic OWL (PR-OWL) [19] is a probabilistic ontology language that extends OWL with semantics based on Multi-Entity Bayesian Networks (MEBN), a Bayesian probabilistic logic [1]. PR-OWL has been extended to PR-OWL 2 [14], which provides a tighter link between the deterministic and probabilistic aspects of the Ontologies. MEBN is flexible enough to represent a variety of complex and uncertain situations. MEBN has been applied to systems [2][3][4][5][6][7] for Predictive Situation Awareness (PSAW), providing the ability to estimate and predict aspects of a temporally evolving situation.

Developing probabilistic ontologies can be greatly facilitated by the use of a modeling framework such as the UMP-ST, a modeling process for constructing a probabilistic ontology [13]. The UMP-ST consists of four main disciplines: (1) *Requirement*, (2) *Analysis & Design*, (3) *Implementation*, and (4) *Test*. UMP-ST was used to develop a probabilistic ontology to support PROGNOS (PRobabilistic OntoloGies for Net-Centric Operational Systems), a system to support *Maritime Domain Awareness* (MDA). The existing system for MDA (e.g., US Navy's Net-Centric infrastructure, FORCENet) is used to fuse lower-level multi-sensor data, analyze the fused data by human analysts, and support decision-making for naval operations. However, the era of big data requires greater automation. The PROGNOS probabilistic ontology [7] supports ingestion of lower-level data, fusion of heterogeneous input, and probabilistic assessment of situations to improve MDA. PROGNOS is a prototype system that aims especially to identify threatening targets (e.g., terrorists and terrorist-ships).

Manually developing and maintaining a probabilistic ontology is a labor-intensive and insufficiently agile process. Furthermore, it is important to make use of data when available. Therefore, greater automation through a combination of reference models and machine learning methods has the potential to enhance agility and effectiveness in modeling a probabilistic ontology for PSAW. For this reason, a process for Human-aided MEBN Learning in PSAW (HMLP) has been suggested [20]. HMLP contains three supporting methodologies, MEBN-RM [10], a reference MEBN model for PSAW [8], and MEBN learning algorithms [9][10]. These component methodologies enable efficient and effective modeling. MEBN-RM and the reference model are introduced in Section 2 below.

In previous work, we used UMP-ST to develop the PROGNOS PO. This paper presents an extended PROGNOS PO developed using HMLP. In the following sections, the paper (1) provides background information, (2) introduces the original PROGNOS PO derived from UMP-ST, (3) presents the extended PROGNOS PO derived from HMLP, and (4) compares two processes.

II. BACKGROUND

This section introduces (1) MEBN, (2) MEBN-RM Mapping Model, (3) A Reference MEBN Model for PSAW, (4) Uncertainty Modeling Process for Semantic Technology (UMP-ST), and (5) Human-aided MEBN learning in PSAW (HMLP). HMLP assumes input data based on the relational model (RM) as its data schema. We choose RM because it is the most popular database model and has the necessary expressive power to represent entities and their relationships. It is necessary to define how to convert elements of RM to elements of MEBN, so a mapping rule between MEBN and RM, called MEBN-RM, was developed. Also, we introduce a reference MEBN model for PSAW which provides a set of basic templates to support the design of a MEBN model for PSAW. HMLP is a modification of UMP-ST, so UMP-ST is introduced in this section. Some of the following background summaries are taken from [20].

A. MULTI-ENTITY BAYESIAN NETWORKS

MEBN is a compact model combining Bayesian networks (BN) with First-order logic (FOL) to represent repeated structures in a joint distribution representing domain knowledge. MEBN is a highly expressive model for treating uncertainty and complex forms of data and information. A MEBN model, called an MTheory, is composed of fragments, called MFrags. An MFrag consists of a set of resident nodes, a set of context nodes, a set of input nodes, an acyclic directed graph for the nodes, and a set of class local distributions (CLD) for the nodes. A resident node is a random variable which is associated with a function or predicate of FOL and whose class local distribution is resident in an MFrag. A context node is derived from a resident node and determines conditions under which the class local distribution defined in the MFrag is valid. An input node has its distribution defined elsewhere and conditions the class local distribution defined in the MFrag. Nodes for an acyclic directed graph are associated with resident and input nodes. An FOL function or predicate of a resident node contains ordinary variables, which can be replaced with entity identifiers to generate multiple instances of the RVs. MFrags in an MTheory are used to generate instances of fragments of BN. The fragments of BN are combined to form a Bayesian network, called a situation-specific Bayesian Network (SSBN). An MTheory can be used to generate an unbounded number of different SSBNs. Further information about MEBN can be found in [1].

B. MEBN-RM Mapping Model

MEBN-RM [10] is a mapping model which provides a specification for how to convert relational databases [11][12] to MTheories [1]. The relational model (RM) is the most popular database model. MEBN-RM provides an entity mapping between a relation in RM and an entity in MEBN, a resident node mapping between an attribute in RM and a resident node in MEBN, an MFrag mapping between a relation in RM and an MFrag in MEBN, and an MTheory mapping between an RM and an MTheory. An MTheory can be constructed automatically from a relational database by using mapping rules in MEBN-RM. Therefore, MEBN-RM can support a MEBN learning algorithm, which develops an MTheory from a dataset, or an MTheory developer, who aims to develop an MTheory using domain knowledge and MEBN knowledge. HMLP exploits MEBN-RM for efficient development of an MTheory.

C. A Reference MEBN Model for PSAW

A reference model is an abstract framework to which a developer refers in order to develop a specific model. A reference MEBN model for PSAW is a reference model for a PSAW-MTheory which specifies references for MFrags, RVs, relationships of RVs, and entities. The reference MEBN model for PSAW can support the design of a PSAW-MTheory and improve the quality of the PSAW-MTheory. The references for entity are classified into five categories (Time entity *T*, Observer entity *OR*, Sensor entity *SR*, Target entity *TR*, and Reported target entity *RT*). Entities derived from these categories describe a situation in which an observer *OR* observes a target *TR* and interprets it as a reported target *RT* using a sensor *SR* at a certain time *T* [20]. The reference MEBN model for PSAW provides some referring random variables (RV), called PSAW-RVs. PSAW-RVs are classified into five categories (Observing condition RV, Reported object RV, Target object RV, Situation RV, and Context RV). These PSAW-RVs are defined in five types of MFrags (Observing condition MFrag, Report MFrag, Target MFrag, Situation MFrag, and Context MFrag). An observing condition RV defined in an observing condition MFrag represents probabilistic knowledge about conditions of a sensor (e.g., maintenance conditions for a sensor). A reported object RV defined in a report MFrag represents probabilistic knowledge about a relation or an attribute of observed targets (e.g., a reported target size). A target object RV defined in a target MFrag represents probabilistic knowledge about a relation or an attribute for actual targets (e.g., an actual target size). A situation RV defined in a situation MFrag represents probabilistic knowledge about situations of targets (e.g., a collaborating situation for targets). A context RV defined in a context MFrag represents probabilistic knowledge about conditions under which the class local distribution defined in the MFrag is valid. For example, an RV Predecessor(*pre_t*, *t*) can be a context RV. The context RV Predecessor(*pre_t*, *t*) means that the time interval *pre_t* occurs immediately before the time interval *t*. More specific information for the reference MEBN model for PSAW can be found in [20].

D. Uncertainty Modeling Process for Semantic Technology (UMP-ST)

UMP-ST is a framework to support the design of a probabilistic ontology [13]. The PROGNOS probabilistic ontology was developed using UMP-ST. UMP-ST provides processes for constructing a probabilistic ontology through four disciplines: (1) *Requirement*, (2) *Analysis & Design*, (3) *Implementation*, and (4) *Test*.

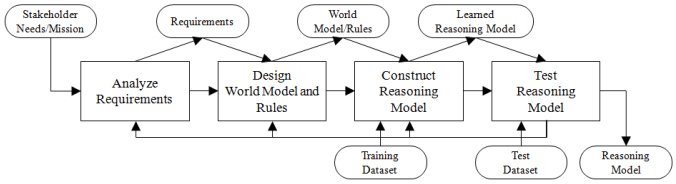
In the *Requirement* discipline, requirement statements are defined. The requirement statements can contain goals, queries, and evidence for a probabilistic ontology. Objectives to be achieved by reasoning with the probabilistic ontology are specified by statements for goals (e.g., detect a ship of interest). To achieve the objectives, specific query statements are specified in this discipline (e.g., what is the type of a ship?). To support the queries, evidence associating with the queries is determined in this discipline (e.g., an appearance of a ship). In the *Analysis & Design* discipline, entities, attributes, relationships, and probabilistic rules are defined. These are used to support the goals, queries, and evidence. For example, we are developing a probabilistic ontology, which aims to detect a ship of interest (the goals). The goal is achieved by identifying the type of a ship (the queries) given information about the appearance of the ship (the evidence). For this situation, a ship entity is required. Also, type and appearance attributes for the ship entity are required. Suppose that the appearance attribute may depend on the type attribute. This is specified by a probabilistic rule. In the *Implementation* discipline, a probabilistic ontology is developed using results from the previous disciplines. A probabilistic ontology based on MEBN is used to reason about uncertainty. Therefore, a probabilistic ontology contains OWL classes based on elements from MEBN such as an MFrag, an MTheory, a node, a probability distribution, and an entity. In this step, these OWL classes are defined. For example, the ship entity defined in the previous discipline is mapped to an entity type indicating a ship in the probabilistic ontology. The attributes ship appearance and ship type are mapped to random variables ship appearance and ship type, respectively. The probabilistic rule for the attributes ship appearance and ship type is converted to the joint probability for the random variables ship appearance and ship type. The random variables ship appearance and ship type may belong to an MFrag representing attributes of a ship. The MFrag ship and other MFrags related with a maritime domain may integrate into an MTheory representing a maritime situation. The *Test* discipline is used to assess the probabilistic ontology developed in the *Implementation* discipline. More specific information for UMP-ST can be found in [13].

E. Human-aided MEBN learning in PSAW (HMLP)

HMLP is a framework which aims the development of a probabilistic ontology in PSAW. HMLP provides specific development steps and supporting methods (MEBN-RM, the reference MEBN model for PSAW, and MEBN learning). HMLP improves MEBN learning by providing expert knowledge which is used to limit the search space of parameters, variables, and structures for a probabilistic ontology in PSAW.

Similar to the four disciplines of UMP-ST, HMLP contains four steps (Fig. 1): (1) *Analyze Requirements*, (2) *Design World Model and Rules*, (3) *Construct Reasoning Model*, and (4) *Test Reasoning Model*. (See a full discussion of HMLP in [20]). A summary of HMLP is presented below.

Fig. 1. Process for Human-Aided MEBN Learning (This figure was taken from [20] and was modified)



Stakeholders who request the development of a reasoning model or a probabilistic ontology provide needs and/or missions as inputs of HMLP. An output from the end of HMLP is a reasoning model (in our case, a probabilistic ontology for PSAW). The followings describe the four steps in HMLP. (1) In the *Analyze Requirements* step, requirements which contain goals to be achieved, queries to answer, and evidence to be used in answering queries are defined. Also, the requirements include performance criteria, which are used in the *Test Reasoning Model* step, to evaluate the probabilistic ontology. (2) In the *Design World Model and Rules* step, a world model and rules are developed using the requirements in the previous step. This step contains two sub-steps (*Design World Model* step and *Design Rules* step). The *Design World Model* step defines the world model which may include entities, attributes, and relations (e.g., RM) using the requirements, domain knowledge and/or existing data schemas. The world model is used to identify rules. In the *Design Rules* step, the rules or influencing relationships between attributes in the world model are defined. (3) In the *Construct Reasoning Model* step, a probabilistic ontology is constructed using a training dataset, the world model, and the rules. This step includes two sub-steps (*Map to Reasoning Model* step and *Learn Reasoning Model* step). The *Map to Reasoning* step maps the world model and rules to a candidate probabilistic ontology. The *Learn Reasoning Model* uses a MEBN learning method to learn the probabilistic ontology from a training dataset. (4) The *Test Reasoning Model* step evaluates the learned probabilistic ontology in the previous step to determine whether to accept it. The accepted probabilistic ontology is a final result from HMLP.

III. PROGNOS PO VIA UMP-ST

To develop the PROGNOS PO, three iterations of the four steps in UMP-ST (*Requirement*, *Analysis & Design*, *Implementation*, and *Test*) were performed [14]. The following sub-sections summarize the four steps in UMP-ST to develop the PROGNOS PO.

A. Requirements

The *Requirement* step identifies requirements containing goals, queries, and evidence for a probabilistic ontology. The requirements for the PROGNOS PO were developed gradually over the three iterations. In the first iteration, a simple requirement regarding a ship of interest was identified [7]. In the second iteration, requirements for two types of terrorist-ships were defined. In the third iteration, requirements for crew members in a ship of interest were specified. The following list shows part of the resulting requirements [14].

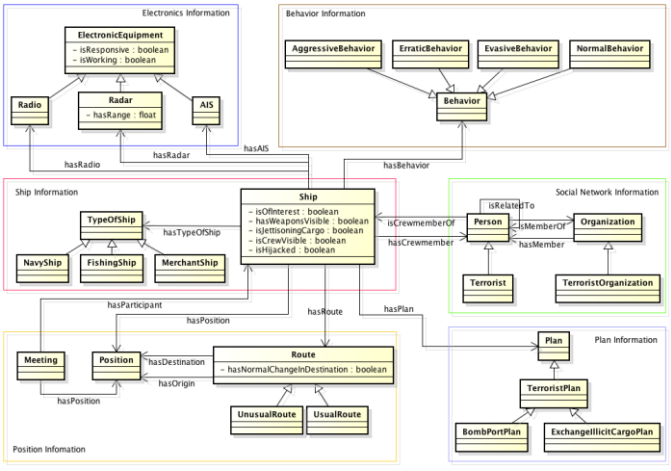
1. *Identify if a ship is of interest,*
1.1 *Is the ship being used to exchange illicit cargo?*
1.1.1 *Was the ship hijacked?*
1.1.2 *Does the ship have a terrorist crew member?*
1.1.2.1 *Is the crew member associated with any terrorist organization?*
…
1.2 *Is the ship being used as a suicide ship to bomb a port?*
…

The main goal was to identify a ship of interest (i.e., a terrorist-ship). In this requirement, we assumed the ship of interest may exchange illicit cargo and/or be used as a suicide ship to bomb a port. To support this goal, we needed to identify the type of a crew member of a ship. If the type of a crew member is a terrorist, the ship is highly likely to be a terrorist-ship. To identify whether a crew member is a terrorist, we can check whether the crew member is associated with any terrorist organization.

B. Analysis & Design

This step defines the types of entities, their properties and relationships, and the rules that apply to them, i.e., the semantics of the domain model. The Unified Modeling Language (UML) diagrams can provide a convenient and understandable visualization of the classes and relationships for the model semantics. The requirements defined in the previous step are used to develop the model semantics. Thus, entities, attributes for the entities, and relationships between the entities were identified. For example, from Requirement 1, an entity was derived (i.e., a ship) and an attribute of the entity was derived (i.e., the type of a ship). From Requirement 1.1.2, a new entity was derived (i.e., a (terrorist) person) and a relationship between the entities was derived (i.e., a ship has a crew (terrorist) member). In the second iteration, Carvalho [14] developed the model represented by UML as shown in Fig. 2.

Fig. 2. Entities, their attributes, and relations for the MDA model after the second iteration (This figure provided by permission of Carvalho [14])



The classes and relationships form six natural groups (i.e., *Electronics*, *Behavior*, *Ship*, *Position*, *Plan*, and *Social Network*). The ship types are *NavyShip*, *FishingShip*, and *MerchantShip*. Ship routes are *UnusualRoute* and *UsualRoute*. Two ships can meet each other at a position. A ship can use electronic devices such as *Radio*, *Radar*, and *AIS* (Automatic Identification System). A ship can show *behavior* such as *Aggressive*, *Erratic*, *Evasive*, and *Normal*. A ship can have a (terrorist) crewmember who may belong to a (terrorist) organization. A ship can have a terrorist plan such as *BombPort* and *ExchangeIllicitCargo*.

After developing the model semantics, conditional rules were identified. There were three iterations of this process. The following list shows a few of the conditional rules from [14].

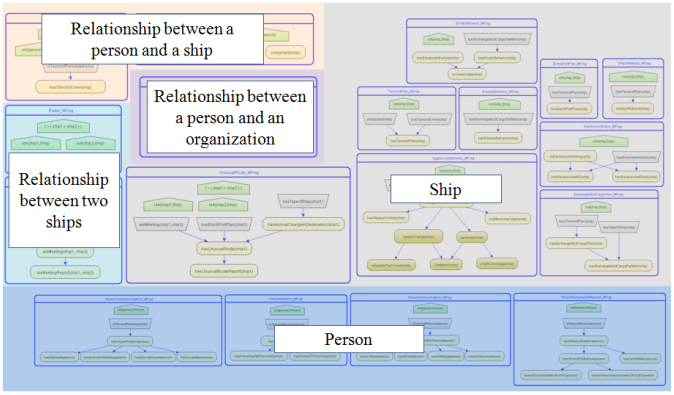
1.(a) *If a crew member is a member of a terrorist organization, then it is more likely that he is a terrorist.*
1.(b) *If an organization has a terrorist member, it is more likely that it is a terrorist organization.*
…
4.(a) *Research shows that if a crew member has a relationship with terrorists, there is a 68% chance that he has a friend who is a terrorist.*
…

These conditional rules were derived from extensive research about terrorism [16] and from the knowledge provided by a domain expert. These rules were used to develop the PROGNOS PO.

C. Implementation

In the *Implementation* step, the PROGNOS PO was designed. The PROGNOS PO can be found in [14][15]. Fig. 3 shows the PROGNOS PO containing five groups of MFrags.

Fig. 3. Original PROGNOS probabilistic ontology



The first set of MFrags is for a ship of interest. It includes nine MFrags *Aggressive Behavior*, *Terrorist Plan*, *Evasive Behavior*, *Erratic Behavior*, *Unusual Route*, *Bomb Port Plan*, *Ship Of Interest*, *Electronics Status*, and *Exchange Illicit Cargo Plan*. These MFrags are used to reason about properties of a ship (e.g., unusual behavior and an illegal plan). The second set of MFrags is for a person of interest. It includes four MFrags *Person Communications*, *Person Background Influences*, *Person Cluster Associations*, and *Person Relations*. These MFrags are used to identify a person who may communicate with a terrorist, has a suspicious background and history, and has a relationship with a terrorist. The third set of MFrags is for information of relationships between two ships. It includes two MFrags, *Radar* and *Meeting*. These MFrags are used to identify whether one ship is within radar range of another ship

and whether two ships are meeting. The fourth set of MFrags is for information about the relationship between a person and an organization. It includes one MFrag *Terrorist Person* in which a person who belongs to an organization is identified. The last set of MFrags is for information about a relationship between a person and a ship. It includes two MFrags *Has Terrorist Crew* and *Ship Characteristics*. These MFrags are used to link a person and a ship, and to identify whether a ship has a terrorist crew member.

The following list shows part of a partial PROGNOS PO containing information about MFrags (*F*), context nodes (*C*), resident nodes (*R*), resident parent nodes (*RP*), and input parent nodes (*IP*). Note that a partial probabilistic ontology doesn't contain a class local distribution and domain information for a random variable.

PO 1: Original PROGNOS probabilistic ontology

```
[F: ErraticBehavior_MFrag
        [C: isA(ship,Ship)]
        [R: hasErraticBehavior(ship) [IP: hasExchangeIllicitCargoPartition(ship)]]
        [R: hasEquipmentFailure(ship)]
        [R: isCrewVisible(ship)[RP: hasErraticBehavior(ship)][RP: hasEquipmentFailure(ship)]]
]
[F: TerroristPerson_MFrag
        [C: isA(person,Person), isA(org,Organization)]
        [R: isTerroristOrganization(org)[RP: isTerroristPerson(person), isMemberOfOrganization(person, org)]]
        [R: isTerroristPerson(person)][R: isMemberOfOrganization(person, org)]
]
[F: ShipCharacteristics_MFrag
        [C: isA(ship,Ship), isA(person,Person)]
        [R: hasCrewMember(ship, person)] [R: hasTypeOfShip(ship)][R: isHijacked(ship)]
]
[F: EvasiveBehavior_MFrag
        [C: isA(ship,Ship)]
        [R: hasEvasiveBehavior(ship)[IP: hasExchangeIllicitCargoPartition(ship)]]
]
[F: PersonCommunications_MFrag
        [C: isA(person,Person)]
        [R: communicatesWithTerrorist(person)[IP: isTerroristPerson(person)] ]
        [R: usesChatroom(person) [RP: communicatesWithTerrorist(person)]]
        [R: usesEmail(person) [RP: communicatesWithTerrorist(person)]]
        [R: usesCellular(person) [RP: communicatesWithTerrorist(person)]]
        [R: usesWeblog(person) [RP: communicatesWithTerrorist(person)]]
]
[F: PersonBackgroundInfluences_MFrag
        [C: isA(person,Person)]
        [R: hasInfluencePartition(person) [IP: isTerroristPerson(person)]]
        [R: knowsPersonImprisionedInOIForOEF(person) RP: hasOIForOEFInfluence(person)]]
        [R: hasFamilyStatus(person) [RP: hasInfluencePartition(person)]]
        [R: hasOIForOEFInfluence(person) [RP: hasInfluencePartition(person)]]
        [R: knowsPersonKilledInOIForOEF(person) [RP: hasOIForOEFInfluence(person)]]
]
[F: AggressiveBehavior_MFrag
        [C: isA(ship,Ship)]
        [R: hasAggressiveBehavior(ship) [IP: hasBombPortPlan(ship), hasExchangeIllicitCargoPartition(ship)]]
        [R: hasWeaponVisible(ship) [RP: hasAggressiveBehavior(ship)]]
        [R: isJettisoningCargo(ship) [RP: hasAggressiveBehavior(ship)]]
        [R: speedChange(ship) [RP: hasAggressiveBehavior(ship)]]
        [R: turnRate(ship) [RP: hasAggressiveBehavior(ship)]]
        [R: propellerTurnCount(ship) [RP: speedChange(ship)]]
        [R: cavitation(ship) [RP: speedChange(ship)][RP: turnRate(ship)]]
        [R: shipRCSchange(ship)  [RP: turnRate(ship)]]
]
[F: ShipOfInterest_MFrag
        [C: isA(ship,Ship)] [R: isShipOfInterest(ship) [IP: hasTerroristPlan(ship)]]
]
[F: ExchangeIllicitCargoPlan_MFrag
        [C: isA(ship,Ship)]
        [R: hasExchangeIllicitCargoPlan(ship) [IP: hasTerroristPlan(ship)]]
        [R: hasExchangeIllicitCargoPartition(ship)
                [IP: hasTypeOfShip(ship)][RP: hasExchangeIllicitCargoPlan(ship)]]
]
[F: PersonRelations_MFrag
        [C: isA(person,Person)]
        [R: hasKinshipToTerrorist(person) [RP: hasTerroristBeliefs(person)]]
        [R: hasFriendshipWithTerrorist(person) [RP: hasTerroristBeliefs(person)]]
        [R: hasTerroristBeliefs(person) [IP: isTerroristPerson(person)]]
]
[F: Meeting_MFrag
        [C: isA(ship1,Ship), isA(ship2,Ship)]
        [C: ( ¬ ( ship1 = ship2 ) )]
        [R: areMeeting(ship1, ship2) [IP: hasExchangeIllicitCargoPartition(ship1)]]
        [R: areMeetingReport(ship1, ship2) [RP: areMeeting(ship1, ship2)]]
]
[F: BombPortPlan_MFrag
        [C: isA(ship,Ship)] [R: hasBombPortPlan(ship) [IP: hasTerroristPlan(ship)]]
]
[F: HasTerroristCrew_MFrag
        [C: isA(ship,Ship), isA(person,Person)]
        [C: hasCrewMember(ship,person)]
        [R: hasTerroristCrew(ship) [IP: isTerroristPerson(person)]]
]
[F: UnusualRoute_MFrag
        [C: isA(ship2,Ship), isA(ship1,Ship)]
        [C: ( ¬ ( ship1 = ship2 ) )]
        [R: hasUnusualRoute(ship1)
                [RP: hasNormalChangeInDestination(ship1)]
                [IP: hasBombPortPlan(ship1)][IP: areMeeting(ship1,ship2)]]
        [R: hasUnusualRouteReport(ship1) [RP: hasUnusualRoute(ship1)]]
        [R: hasNormalChangeInDestination(ship1) [IP: hasTypeOfShip(ship1)]]
]
[F: TerroristPlan_MFrag
        [C: isA(ship,Ship)]
        [R: hasTerroristPlan(ship) [IP: hasTerroristCrew(ship)][IP: isHijacked(ship)]]
]
[F: ElectronicsStatus_MFrag
        [C: isA(ship,Ship)]
        [R: isElectronicsWorking(ship)]
        [R: hasResponsiveRadio(ship)
                [IP: hasEvasiveBehavior(ship)][RP: isElectronicsWorking(ship)]]
        [R: hasResponsiveAIS(ship)
                [IP: hasEvasiveBehavior(ship)][RP: isElectronicsWorking(ship)]]
]
[F: Radar_MFrag
        [C: isA(ship1,Ship), isA(ship2,Ship)] [C: ( ¬ ( ship1 = ship2 ) )]
        [R: isWithinRadarRange(ship1, ship2)]
]
[F: PersonClusterAssociations_MFrag
        [C: isA(person,Person)]
        [R: hasOccupation(person) [RP: hasClusterPartition(person)]]
        [R: hasEducationLevel(person) [RP: hasClusterPartition(person)]]
        [R: hasClusterPartition(person) [IP: isTerroristPerson(person)]]
        [R: hasEconomicStanding(person) [RP: hasClusterPartition(person)]]
        [R: hasNationality(person) [RP: hasClusterPartition(person)]]
]
```

PO 1 shows the context nodes and the resident nodes in the MFrags, and the relationship between the resident nodes. For example, the MFrag *ErraticBehavior_MFrag* (Line 1~6) contains an *isA* context node and three resident nodes *hasErraticBehavior*, *hasEquipmentFailure*, and *isCrewVisible*. The resident node *hasErraticBehavior* is influenced by an input node *hasExchangeIllicitCargoPartition*. The resident node *isCrewVisible* is influenced by the resident nodes *hasErraticBehavior* and *hasEquipmentFailure*. This PROGNOS PO was tested in the next step.

D. Test

In this step, the PROGNOS PO was evaluated to determine whether to accept it. To do this, the case-based evaluation, in which various scenarios were defined and used to examine the reasoning implications of the probabilistic ontology, was used. For example, given a scenario which was developed by a subject matter expert (SME), some information (e.g., history of a target) from the scenario for a target was used as evidence for inference of the PROGNOS PO to identify some properties (e.g., whether the target is a terrorist) of the target. If the result of inference coincided exactly with the scenario from SME, we could think that the probabilistic ontology was reasonable. For this test, three qualitatively different scenarios were used [14].

After three iterations for UMP-ST, an overall test for the PROGNOS PO was performed using a simulation. In the real world situation, it is very difficult to acquire a real dataset to develop such a probabilistic ontology which contains rare events. For this reason, the simulation was used to produce a test dataset given different scenarios generated randomly. Carvalho [14] and Costa et al [15] introduced some results for this test. In such a test, it is important that knowledge used to develop a probabilistic ontology and knowledge used to develop a simulation for testing the probabilistic ontology should not be same. If they are same, the test is meaningless, because the probabilistic ontology and the simulation are same models, but just in different forms.

IV. PROGNOS PO VIA HMLP

In this section, we introduce an extended PROGNOS PO derived from the HMLP process. The following shows how the development operates.

A. Analyze Requirements

This step is not much different from the requirement step in UMP-ST. Therefore, we can reuse requirements developed

from the PROGNOS project. The full requirements can be found in [14]. However, the reference MEBN model for PSAW can provide more items by which a PSAW modeler can consider predefined entities, RVs, and MFrags for PSAW. Recall the four MFrag groups from the reference model: *Reported Object*, *Observing Conditions*, *Target Object*, and *Situation*. The last of these, *Situation*, is of special note. In PSAW, understanding a situation in which targets operate for their own purposes is one of the important issues. Identifying just the type of a target is an insufficient task for PSAW. The meaning of *awareness* is not to perceive and estimate actual properties of a target but is to understand, interpret, and explain the relationships between targets. Kokar et al [17] stated: "*The main part of being aware is to be able to answer the question of "what's going on?"*". Awareness of a situation is subjective according to an observer, who is aware of the situation. The modeler, who is developing a probabilistic ontology to support PSAW, should define what situation will be considered and explained through all observation from the world. For the awareness of the PROGNOS situation, we add the following new requirement.

New Goal 1: Recognize emergency situation at sea

Query 1.1: How high is the potential terrorist threat?

Evidence 1.1.1: Ship(s) of interest

Evidence 1.1.2: Crew member(s) of interest

The new goal aims to alert a response team when the threat reaches a certain level. This will be accomplished by estimating potential terrorist attacks in the field given estimation of terrorist ships and terrorist crew members.

In HMLP, a requirement can contain a performance criterion specifying a measure of accuracy (e.g., the mean squared error or the Brier score [26]). For example, we might require that the mean squared error between ground truth and estimated results from the probabilistic ontology shall be less than a given threshold (e.g., a mean squared error < 0.1).

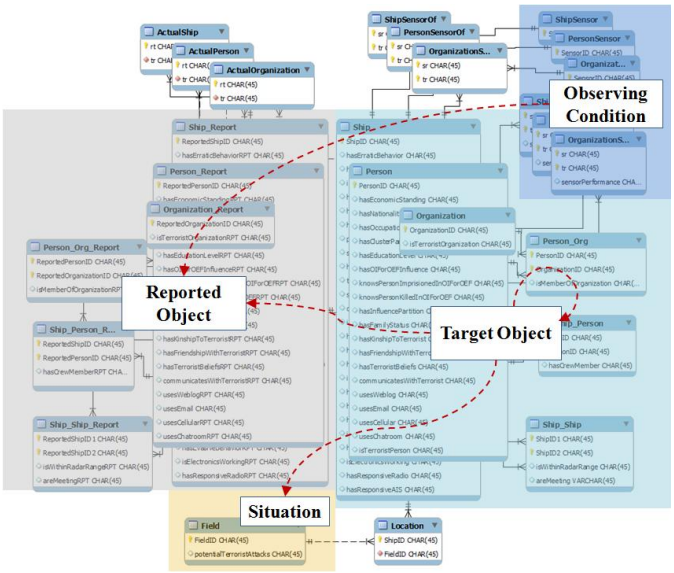
B. Design World Model and Rules

This step performs two sub-steps (*Design World Model* and *Design Rules*). The *Design World Model* step is to define a world model for PROGNOS from the requirements defined in the previous step.

In this step, the reference MEBN model for PSAW can be used to identify possible entities, random variables, and relationships between the random variables. Fig. 4 shows a PROGNOS world model represented in an EER (enhanced entity–relationship) model. We develop the PROGNOS world model using the requirements and the reference model.

The reference model suggests four groups: (1) *Reported Object*, (2) *Observing Condition*, (3) *Target Object*, and (4) *Situation*. A world model for the original PROGNOS PO included the seven relations (e.g., *Target*, *Ship*, *Person*, *Organization*, *Person_Org*, *Ship_Person*, and *Ship_Ship*). The original PROGNOS PO treated only the target object group. In other words, it did not emphasize sensing. We would expect evidence (e.g., reported objects) to be reported to estimate actual targets (e.g., target objects), so relations (i.e., *Ship_Report*, *Person_Report*, *Organization_Report*, *Ship_Ship_Report*, *Person_Org_Report*, *Ship_Person_Report*, and *ReportedTarget*) for the reported object group are added in the world model for the extended PROGNOS PO. Observations may contain observation errors influenced by observing conditions (e.g., weather). The observing condition group contains two relations *Sensor* and *SensorProperty*. In the previous step, a requirement for the awareness for a situation was added. Therefore, we added a relation *Field* for the situation group in Fig. 4. Relations (i.e., *Location*, *SensorOf*, and *ActualTarget*) which are not classified in these groups are supporting relations used to join the relations in the four groups.

Fig. 4. Part of EER Model for a PROGNOS world model



The reference model provides some rules or relationships between these groups as shown in the arrows (Fig. 4). The observing conditions group and the target object group can influence the reported object group. For example, the attribute *sensorPerformance* in the relation *SensorProperty* influenced the report attributes in the report relations *Ship_Report*, *Person_Report*, *Organization_Report*, *Ship_Ship_Report*, *Person_Org_Report*, and *Ship_Person_Report*. The arrows in Fig. 4 indicate these relationships. The following shows a few of these rules.

Rule 1: *causal* ({*hasErraticBehavior*, *sensorPerformance*}, *hasErraticBehaviorRPT*)

Rule 2: *causal* ({*isShipOfInterest*, *isTerroristPerson*}, *PotentialTerroristAttacks*)

...

Rule 1 means that two attributes *hasErraticBehavior* and *sensorPerformance* cause the attribute *hasErraticBehaviorRPT*. Rule 2 means that two attributes *isShipOfInterest* and *isTerroristPerson* cause the attribute *PotentialTerroristAttacks*.

C. Construct Reasoning Model

This step performs two sub-steps (*Map to Reasoning Model* and *Learn Reasoning Model*) to construct the PROGNOS PO. MEBN-RM provides a converting rule from RM to a probabilistic ontology. Entity relations which contain only one attribute for the primary key of the relation (e.g., *ship* and *person*) can be defined as entity types in the probabilistic ontology. Each of the attributes in the relations could be mapped to a resident node in the probabilistic ontology using MEBN-RM. For example, the attribute *hasErraticBehavior* of the relation *Ship* became the resident node *hasErraticBehavior*(*ship*).

Rules which are defined in the previous step are used to develop relationships between resident nodes in the probabilistic ontology. For example, from Rule 1, we had a conditional dependence P(*hasErraticBehaviorRPT*(*ship_report*) | *hasErraticBehavior*(*ship*), *sensorPerformance*(*shipSensor*, *ship*)). From Rule 2, we had a conditional dependence P(*PotentialTerroristAttacks*(*field*) | *isShipOfInterest*(*ship*), *isTerroristPerson*(*person*)).

We could model the extended PROGNOS PO as shown in Fig. 5 using the resident nodes, the relationships between the resident nodes, and the MFrag groups.

Fig. 5. Extended PROGNOS probabilistic ontology

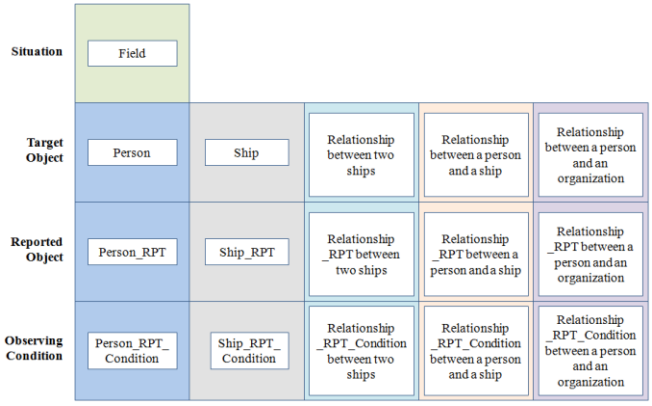


Fig. 5 shows a set of MFrags in the extended PROGNOS PO. The list on the left indicates the four MFrag groups. Each group is decomposed into sub-groups. For example, the target object group contains five sets of MFrags (Person MFrags, Ship MFrags, MFrags for the relationship between two ships, MFrags for the relationship between a person and a ship, and MFrags for the relationship between a person and an organization). The following list (PO 2) shows part of new MFrags added into the extended PROGNOS PO.

PO 2: Part of New MFrags added into the original PROGNOS probabilistic ontology

```
1     [F: Orgainzation_Report_MFrag
2           [C: isA(sr,SENSOR), isA(tr ,ORGANIZATION), isA(rt,REPORTEDTARGET)]
3           [C: SensorOf(sr, tr), tr = ReportedTarget(rt)]
4           [R: isTerroristOrganizationRPT(rt)
5                 [IP: isTerroristOrganization(tr)]
6                 [IP: performance(sr, tr)]
7           ]
8     ]
9     [F: Situation_MFrag
10          [C: isA(ship,SHIP), isA(person,PERSON), isA(field,FIELD)]
11          [C: field = Location(ship)]
12          [C: hasCrewMember(ship, person)]
13          [R: PotentialTerroristAttacks(field) [IP: isShipOfInterest(ship), isTerroristPerson(person)]]
14    ]
15    ...
```

In PO 2, we added the ship report MFrag which can be used to reason about Rule 1. Also, we added the situation MFrag which can be used to reason about Rule 2.

In the *Learn Reasoning Model* step, the extended PROGNOS PO can be refined using a MEBN learning algorithm. The goal of MEBN learning is to learn an MTheory from a training dataset. A basic MEBN learning method for relational datasets was suggested [9][10]. This approach assumes that the training dataset is stored in a relational database based on RM. MEBN learning searches parameters, variables, and structures to find an MTheory that provides a good fit to the training dataset. In our case, the structures are given by the above steps as suggested in the PSAW reference model. Therefore, only parameter learning is required. The goal of parameter learning is to estimate the parameters θ^* of a class local distribution L given a training dataset $\boldsymbol{D}$ and the type of distribution being learned, which fit well the training dataset $\boldsymbol{D}$.

For a discrete random variable case, Dirichlet distribution is commonly used because it is conjugate to the multinomial distribution. With a Dirichlet prior distribution, the posterior predictive distribution has a simple form [21][22]. For continuous random variables, multiple regression can be used. Park et al [9] introduced a basic MEBN parameter learning and structure learning for a conditional Gaussian hybrid model in which no discrete random variable may have a continuous parent random variable.

For example, parameters for a conditional Gaussian distribution can be estimated using multiple regression. The following class local distribution (CLD) is an illustrative example of a conditional linear Gaussian CLD for the node *Speed_RPT*(*rt*, *tr*), which means a speed report *rt* for a target *tr*. The CLD of the node is a continuous CLD with hybrid parents (*Sensor_Condition* and *Speed*). In this case, we assume that the discrete parent node *Sensor_Condition*(*sr*, *tr*), which means a condition of a sensor *sr* for a target *tr*, has two states (*Good* and *Bad*) and the node *Speed* (*tr*), which means an actual speed of a target *tr*, is continuous.

CLD 1 [Conditional Linear Gaussian]: *Speed_RPT*(*rt*, *tr*)

```
1     if some sr.tr have (Sensor_Condition = Good) [
2           Θ1.0 + Θ1.1*Speed + NormalDist(0, Θ1.2)
3     ] else [
4           Θ2.0 + Θ2.1*Speed + NormalDist(0, Θ2.2)
5     ]
```

Parameter learning for this CLD estimates the parameters ($\Theta_{1.0}$, $\Theta_{1.1}$, and $\Theta_{1.2}$) in Line 2 and the parameters ($\Theta_{2.0}$, $\Theta_{2.1}$, and $\Theta_{2.2}$) in Line 4 using multiple regression.

D. Test Reasoning Model

This step performs two sub-steps (*Experiment Reasoning Model* and *Evaluate Experimental Results*) to evaluate the extended PROGNOS PO from the test dataset. In the *Experiment Reasoning Model* step, the performance of estimation and prediction for the extended PROGNOS PO can be assessed using a performance measure (e.g., the mean squared error or the Brier score). Each experiment consists of the following five steps. (1) The test dataset provides entity

information (e.g., ship1, person1, and field1) and ground truth information (e.g., isShipOfInterest_ship1 = true, isTerroristPerson_person1 = true) to the extended PROGNOS PO. (2) Given these, the extended PROGNOS PO is used to compute a marginal probability distribution (e.g., P(PotentialTerroristAttacks_field1 | isShipOfInterest_ship1 = true, isTerroristPerson_person1 = true) in response to a query. (3) The test dataset provides ground truth data (e.g., PotentialTerroristAttacks_field1 = High). (4) Steps 1-3 are repeated for all test cases. (5) Finally, for results for all cases, the measures are calculated.

In the *Evaluate Experimental Results* step, we evaluate the measures using the performance criteria in the requirements defined in the *Analyze Requirement* step (e.g., a mean squared error < 0.1). If the evaluation is not satisfied (e.g., a mean squared error >= 0.1), we can return to the previous steps to improve the performance of the extended PROGNOS PO. We can investigate the extended PROGNOS PO in the *Construct Reasoning Model* step. Unsatisfactory performance can be caused by a training database of insufficient size. In this case, we may find more datasets and apply them to the learning process. Also, it is possible that the MEBN learning algorithm which we use is ineffective. In this case, the application of a more effective MEBN learning algorithm is required. The world model in the *Construct Reasoning Model* step can be incorrect. For this, we may need to conduct a further field investigation and research to develop a more accurate world model. The requirements in the *Analyze Requirements* step can be impracticable or requires a too high standard to address it. In this case, readjustments for the requirements can be performed by the stakeholders.

V. COMPARING UMP-ST AND HMLP

HMLP is a modification of UMP-ST that specifies some detailed sub-steps and uses two reference models (the reference MEBN model for PSAW and MEBN-RM). These reference models can support efficient modeling for a probabilistic ontology in PSAW. The first steps (*Requirement*) for both processes are same. In the case of HMLP, the reference MEBN model for PSAW provides some guidance on groups of entities to be defined (i.e., Reported Object, Observing Condition, Target Object, and Situation). In the second step of HMLP, the reference model also supports developing a world model in terms of PSAW by providing candidate entities (i.e., *T*, *OR*, *SR*, *TR*, and *RT*), attributes, and relationships. In the third step of HMLP, MEBN-RM supports the development of entities, random variables, and MFrags from a relational schema. HMLP also makes use of MEBN learning algorithms, so given a training dataset, a probabilistic ontology can be efficiently constructed. The second and third steps are mainly different with UMP-ST. These steps in HMLP can accelerate the modeling for probabilistic ontologies in PSAW and produce more comprehensive models.

Table 1 shows feature comparison between the original PROGNOS PO and the extended PROGNOS PO. Each number in the table means the number of the features (entities, random variables, relationships between random variables, and MFrags). For example, the number of entities in the original model is three (*Ship*, *Person*, and *Organization*), while the number of entities in the extended model is ten (*Field*, *Ship*, *Person*, *Organization*, *ShipSensor*, *PersonSensor*, *OrganizationSensor*, *ReportedShip*, *ReportedPerson*, and *ReportedOrganization*). Table 1 shows that the feature of the extended PROGNOS PO is more comprehensive than the feature of the original PROGNOS PO. The original PROGNOS PO contains 51 RVs, while the extended PROGNOS PO contains 115 RVs. This means that the extended PROGNOS PO can answer more various questions. For example, a reasoning about potential terrorist attacks in a field can be performed using the extended PROGNOS PO, but the original PROGNOS PO can't. Also, the extended PROGNOS PO contains observing conditions for sensors, so this may enable us to perform more accurate reasoning.

TABLE 1. Comparison between the original PROGNOS probabilistic ontology and the extended PROGNOS probabilistic ontology

	Entities	Random Variables	Relationships	MFrags
Original	3	51	53	18
Extended	10	116	147	36

If we assume that there is a training dataset for MEBN learning, the development period for the PROGNOS PO can be reduced. Usually, to develop an RV and its parameter, we study literature related to the RV and find possible parameters for the RV. Another way for the development of such an RV is to use domain expert knowledge. A subject matter expert (SME) may provide values and parameters for the RV, and relationships between RVs. In the PROGNOS project, to develop one RV, we used the following steps: (1) an SME in the maritime domain explained domain knowledge to an RV developer, (2) the RV developer developed the RV using the MEBN/PR-OWL software [27], and (3) the RV in the MEBN/PR-OWL software was evaluated by the SME. These steps were implemented with at least one day per RV. If we assume that for each RV, one day may be required to develop it by one RV developer and one SME, then the original PROGNOS PO requires around 51 days. On the contrary, if we assume that all datasets are available, the development with MEBN learning may require around one day for setting the datasets and learning a PO using a MEBN learning algorithm.

VI. CONCLUSION

UMP-ST was applied for construction of a probabilistic ontology to support PROGNOS including the PROGNOS PO. The PROGNOS PO played an important role in the operation of PROGNOS. However, manually developing and maintaining a probabilistic ontology is a labor-intensive and insufficiently agile process. Therefore, HMLP containing the reference models and machine learning methods was introduced. In the previous work for PROGNOS, UMP-ST was applied to develop the PROGNOS PO. This paper applied HMLP to develop the extended PROGNOS PO which was more comprehensive than the original model and was developed more quickly.

The following summarizes future research issues. (1) HMLP in this research was not fully applied with MEBN learning from a training dataset. Evaluation of effectiveness (i.e., reasoning accuracy) of reasoning models learned from

MEBN learning is required. (2) A probabilistic ontology can contain MFrags, context nodes, resident (or inputs) nodes, graphs, FOL formula for nodes, and class local distributions for nodes. These elements can be subject to MEBN learning. Especially, FOL formula learning in a probabilistic ontology is a difficult topic relative to the others. In our approach, a dataset for learning is given from a relational database. Because we rely on MEBN-RM, we do not need to perform the complicated task of FOL formula learning from text data. FOL formula learning in a probabilistic ontology can be supported by Inductive Logic Programming [23][24] and Statistical Natural Language Processing [25]. (3) Also, future steps for the extended PROGNOS PO are to apply it to a realistic reasoning system for Maritime Domain Awareness.

ACKNOWLEDGMENTS

We appreciate Dr. K. C. Chang, Dr. W. Sun, Dr. R. Carvalho, Dr. R. Haberlin, Mr. S. Matsumoto, and Mr. A. Mugali for their efforts in the previous PROGNOS research. The research was partially supported by the Office of Naval Research (ONR), under Contract#: N00173-09-C-4008.

REFERENCES

[1] Laskey, K. B. (2008). MEBN: A Language for First-Order Bayesian Knowledge Bases. Artificial Intelligence, 172(2-3).

[2] Laskey, K. B., D'Ambrosio, B., Levitt, T. S., & Mahoney, S. M. (2000). Limited Rationality in Action: Decision Support for Military Situation Assessment. Minds and Machines, 10(1), 53-77.

[3] Wright, E., Mahoney, S. M., Laskey, K. B., Takikawa, M. & Levitt, T. (2002). Multi-Entity Bayesian NetworksforSituation Assessment. Proceedings of the Fifth International Conference on Information Fusion.

[4] Costa, P. C. G., Laskey, K. B., Takikawa, M., Pool, M., Fung, F., & Wright, E. J. (2005). MEBN Logic: A Key Enabler for Network Centric Warfare. Proceedings of the 10th ICCRTS.

[5] Suzic, R. (2005). A generic model of tactical plan recognition for threat assessment. In Defense and Security (pp. 105-116). International Society for Optics and Photonics.

[6] Costa, P. C. G., Laskey, K. B., & Chang, K. C. (2009). PROGNOS: Applying Probabilistic Ontologies To Distributed Predictive Situation Assessment In Naval Operations. Proceedings of the 14th ICCRTS.

[7] Carvalho, R. N., Costa, P. C. G., Laskey, K. B., & Chang, K. C. (2010). PROGNOS: predictive situational awareness with probabilistic ontologies. In Proceedings of the 13th International Conference on Information Fusion.

[8] Park, C. Y., Laskey, K. B., Costa, P. C. G., & Matsumoto, S. (2014). Predictive Situation Awareness Reference Model using Multi-Entity Bayesian Networks. Proceedings of the 17th International Conference on Information Fusion.

[9] Park, C. Y., Laskey, K. B., Costa, P. C. G., & Matsumoto, S. (2013). Multi-Entity Bayesian Networks Learning For Hybrid Variables In Situation Awareness. Proceedings of the 16th International Conference on Information Fusion (Fusion 2013).

[10] Park, C. Y., Laskey, K. B., Costa, P. C. G., & Matsumoto, S. (2013). Multi-Entity Bayesian Networks Learning In Predictive Situation Awareness. Proceedings of the 18th International Command and Control Technology and Research Symposium (ICCRTS 2013).

[11] Codd, E. F. (1970). A Relational Model of Data for Large Shared Data Banks. Communications of the ACM.

[12] Codd, E. F. (1969). Derivability, Redundancy, and Consistency of Relations Stored in Large Data Banks. IBM Research Report.

[13] Carvalho, R. N., Laskey, K. B., & Da Costa, P. C. (2016). Uncertainty modeling process for semantic technology (No. e2045v1). PeerJ Preprints.

[14] Carvalho, R. N. (2011). Probabilistic Ontology: Representation and Modeling Methodology. PhD Dissertation. George Mason University.

[15] Costa, Paulo. C. G., Laskey, Kathryn B., Chang, Kuo-Chu, Sun, Wei, Park, Cheol Y., & Matsumoto, Shou. (2012). High-Level Information Fusion with Bayesian Semantics. Proceedings of the Nineth Bayesian Modelling Applications Workshop.

[16] Sageman, M. (2004). Understanding terror networks. University of Pennsylvania Press.

[17] Kokar, M. M., Matheus, C. J., & Baclawski, K. (2009). Ontology-based situation awareness. Information fusion, 10(1), 83-98.

[18] Smith, B. (2003). Ontology. Retrieved September 2, 2016from http://ontology.buffalo.edu/smith/articles/ontology_pic.pdf.

[19] Paulo C. G Costa, Bayesian Semantics for the Semantic Web. PhD Dissertation, George Mason University, July 2005. Brazilian Air Force.

[20] Park, C. Y., Laskey, K. B., Costa, P. C., & Matsumoto, S. (2016). A process for human-aided Multi-Entity Bayesian Networks learning in Predictive Situation Awareness. In 19th International Conference on Information Fusion (FUSION).

[21] Heckerman, D., Geiger, D., & Chickering, D. M. (1995). Learning Bayesian networks: The combination of knowledge and statistical data. Machine Learning, 20:197–243.

[22] Koller, D., & Friedman, N. (2009). Probabilistic Graphical Models: Principles and Techniques. The MIT Press, 1 edition.

[23] Lavrac, N. & Dzeroski, S. (1994). Inductive Logic Programming: Techniques and Applications. Ellis Horwood, New York.

[24] Muggleton, S., & De Raedt, L. (1994). Inductive logic programming: Theory and methods. The Journal of Logic Programming, 19, 629-679.

[25] Manning, C. D., & Schütze, H. (1999). Foundations of statistical natural language processing (Vol. 999). Cambridge: MIT press.

[26] Brier, G. W. (1950). Verification of forecasts expressed in terms of probability. Monthly weather review, 78(1), 1-3.

[27] Costa, P., Ladeira, M., Carvalho, R. N., Laskey, K., Santos, L., & Matsumoto,S. (2008). 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), pp. 631–636, AAAI Press.

PR-OWL Decision: Toward Reusable Ontology Language for Decision Making under Uncertainty

Shou Matsumoto, Kathryn B. Laskey, Paulo C. G. Costa
Department of Systems Engineering and Operations Research
George Mason University
Fairfax, VA
[smatsum2, klaskey, pcosta]@gmu.edu

***Abstract*—Decision making is a big topic in Intelligence, Defense, and Security fields. However, very little work can be found in the literature about ontology languages that simultaneously support decision making under uncertainty, abstractions/generalizations with first-order expressiveness, and forward/backward compatibility with OWL—a standard language for ontologies. This work proposes PR-OWL Decision, a language which extends PR-OWL—an extension of OWL to support uncertainty—to support first-order expressiveness, decision making under uncertainty, and backward/forward compatibility with OWL and PR-OWL.**

Keywords—ontology, decision making, uncertainty, OWL

I. INTRODUCTION

Ontologies are engineering artifacts which consist of formal vocabularies of terms, usually describing specific domain knowledge and accessed by persons or computers sharing a common view or domain application. Various interdisciplinary works addressing the engineering aspects of this field have been held in the recent years by the information systems—in a broader sense—community [1, 2, 3, 4, 5]. The Web Ontology Language (OWL) is a standard ontology language which represents classes, properties, and individuals in Semantic Web documents [6]. In 2005, Probabilistic Web Ontology Language (PR-OWL) [7] was formulated to address OWL's lack of support for uncertainty—a ubiquitous factor in complex real-world problems. As a continuing effort, version 2 of PR-OWL [8] was formulated in order to address some backward compatibility issues with its predecessor OWL.

Nevertheless, continuous efforts have been performed in the field of decision support, especially with models supporting uncertainty [9, 10, 11, 12, 13, 14, 15]. Decision making is the process of selecting a course of action among several possibilities, based on values or preferences of some decision maker. Values and preferences play a very important role here, because they represent the desirability of an outcome, in a manner that is different from the likelihood or probability that the outcome will happen.

For example, one's probabilistic model may state that the probability of failing some exam is 20% if you do not study. The decision maker may consider this is an acceptable probability for choosing not to study, given that the impact of failing is nothing more than minor embarrassment. However, if the decision maker may lose his/her job as a consequence of failing the exam, the decision maker would definitely study hard. This well illustrates how difficult it would be for someone to make decisions based *only* on metrics of uncertainty (*e.g.* probabilities or likelihoods of events), and how important values and preferences are in actually taking some action. Consequently, ontologies for decision making need to support both uncertainty and values (or preferences of decision makers). Unfortunately, current ontology tools and languages often do not have standardized constructs for representing preferences.

On the other hand, there are models that were not originally designed for ontologies, but can be used for decision making under uncertainty with explicit representation of values. For instance, classic probabilistic decision models like Influence Diagrams (ID) [16] can be enough to just represent and solve decision-making problems—with representation of actions and values or preferences of a decision maker—with support for uncertainty. However, IDs perform probabilistic reasoning about propositional (as in propositional logic) statements, which is not expressive enough to capture many important situations; thus we would like to have first-order expressiveness (as in First-Order Logic), with functions, predicates, and quantification.

OWL direct semantics [6, 17]—mainstream in ontology languages—offer first-order expressiveness, but they do not natively support uncertainty and decisions (*i.e.* support for efficiently representing and treating actions, values and preferences of decision makers). PR-OWL, being an extension of OWL, also offers first-order expressiveness, and it also offers support for uncertainty, but it lacks support for decisions. It was already stated that IDs offer support for decision and uncertainty, but have only propositional expressiveness. It thus becomes of interest to extend the results we have for the propositional cases to the first-order case. **Therefore, there is a need to extend the syntax of PR-OWL and its underlying logic—Multi-Entity Bayesian Network (MEBN)** [18]**—to include elements of IDs. *PR-OWL Decision*, the extension proposed in this work, addresses this issue.**

Reuse receives special attention, because it is a common, yet powerful way to drastically reduce the development effort. This is why special care is taken for backward and forward compatibility (with OWL). Backward compatibility can be achieved by designing the new language so that systems meant for the new language will automatically function with the older language, due to syntactical similarities. This offers incentives

for legacy system users to migrate to new solutions. Forward compatibility can be achieved by composing the new language's syntax with valid constructions of the older language. Legacy systems may not be able to handle the new portions perfectly, but it ought to be guaranteed that the new construction will not cause legacy systems to fail catastrophically. This increases the practical usefulness of a new solution, because part of new models can be built on well tested legacy systems.

Examples of kinds of decision problems (and related tasks) that could particularly benefit from the new solution are:

- Those which the number of decisions and available actions (choices) are not known in advance. For instance, we can have decisions that repeat over time and the number of choices may increase/decrease for each decision. Other types of repetitions (in probabilistic dependency, or on utility functions) can also be treated by PR-OWL Decision.
- Those using abstractions/concretizations from OWL class hierarchy. For instance, an OWL ontology may indicate that a "Tablet" is a subclass of "Computer", thus a decision making model developed for a "Computer" might work well with a "Tablet" (*e.g.* decision models about information theft involving computers/tablets). PR-OWL Decision handles such inheritance natively.
- When the process involving decision making itself is performed or aided by multiple software systems, interoperability plays a major role. OWL has strong support for interoperability, so does PR-OWL Decision.
- Iterative/incremental model development process may benefit from PR-OWL Decision, due to its aim in reuse. A PR-OWL Decision ontology can be developed incrementally, starting from a well-tested deterministic ontology, then creating a PR-OWL ontology which imports the deterministic ontology (so that the original ontology is kept unchanged), and finally a PR-OWL Decision ontology can import the PR-OWL ontology. Cost of verification and validation is reduced, because previously tested artifacts are reused in "as-is" basis. An example in Software Product Line domain is discussed in the following sub-section.

A. *Software Product Line (SPL) Domain*

Examples presented throughout this paper are based on a Software Product Line (SPL) ontology, which was developed as a Proof of Concept for PR-OWL Decision [19]. SPL is a "family" of software-intensive systems that share a common set of characteristics satisfying specific needs of a particular domain, and are developed from a common set of software assets [20]. The engineering process of SPL is often divided into two phases: *domain engineering* (the process of analyzing, architecting and developing reusable components among the family) and *application engineering* (process of producing a single product by integrating and/or customizing reusable components). Proper SPL practices enable fast production and customization.

Quickly developing a series of configurable/customizable software systems is important not only because software is ubiquitous in any current intelligence, defense or security system, but also because such systems are becoming increasingly complex and competitive, both in terms of pricing and available functionalities. Problems in intelligence, defense, and security are diverse, thus it's natural to think that not all clients will use of the entire set of available system features. Quickly—and automatically—offering a proper set of features to the client, given their particular needs, would help in establishing a competitive price, and also to avoid unnecessary use of computational resources caused by unused features (the latter may become rather critical in embedded systems). Our Proof of Concept model mainly addresses this issue.

The following list summarizes some important concepts of SPL that are referenced throughout this paper:

- ***Features*** are common and variant characteristics among a set of software systems. These are related to (or originated from) a set of domain requirements, and can be mapped to a set of software assets, so it can be thought as an abstraction that maps requirements to reusable components.
- ***Configuration*** can be thought as a set of features which jointly satisfies constraints of consistency (*e.g.* dependency and compatibility). We can move from a configuration to another by adding, removing, or substituting features, of course, without breaking consistency rules.
- ***Domain requirements*** are requirements identified and treated in the domain engineering process (*i.e.* "inter-system" requirements that will derive features and related reusable components).
- ***Application requirements*** are requirements treated in the application engineering process (*i.e.* emerging requirements that will result in a single product). A "requirement" in SPL can be either a domain or application requirement.

The Proof of Concept ontology was developed in a iterative/evolving manner, starting from a simple, deterministic OWL ontology, which captured the features and their constraints. Then, a PR-OWL ontology which encodes some probabilistic relationships between the features, requirements, and assets was developed by reusing (importing) the original ontology. Finally, a PR-OWL Decision ontology was developed in order to represent the costs and profits (with associated risks) of incorporating new features to some configuration given emerging requirements. The resulting ontology is able to solve, for example, a decision problem of choosing the set of features to (re)use during application engineering, under maximum expected profit (or minimum expected cost) criteria.

II. PR-OWL

Traditional ontologies have no built-in mechanism for representing or drawing inferences under uncertainty. The *Probabilistic Web Ontology Language* (PR-OWL) consists of a set of classes and properties (relationships) that collectively form a framework for building and reasoning with probabilistic ontologies, yet keeping syntactical compatibility with OWL. The purpose of a probabilistic ontology is to describe knowledge about a domain and its associated uncertainty in a principled, structured, and sharable way, so that it can be applied to support semantic applications working in complex open-world environments. PR-OWL 2 is an extension of OWL 2 with

enhanced meta-level [1] support for specifying probability distributions of OWL properties [8]. Constructs of PR-OWL basically follow an abstraction inherent from *Multi-Entity Bayesian network*, which is explained in next sub-section.

A. *Multi-Entity Bayesian Network*

Multi-Entity Bayesian Network (MEBN) [18] is the underlying logic of PR-OWL (and its version 2). For this reason, a PR-OWL specification can be informally seen as a scheme for describing a MEBN model in OWL.

MEBN extends BN [21] by combining the expressiveness of First-Order Logic and the inference power of BN. MEBN represents the world as a collection of inter-related entities, their respective attributes, and relations among them. Knowledge about attributes of entities and their relationships is represented as a collection of repeatable patterns, known as MEBN Fragments (MFrags). A set of well-defined MFrags that collectively satisfies first-order logical constraints ensuring a unique joint probability distribution is a MEBN Theory (MTheory). The probabilistic portion of a consistent PR-OWL 2 ontology represents an MTheory.

An MFrag represents uncertain knowledge about a collection of related random variables (RVs). RVs, also known as "nodes" of an MFrag, represent the attributes and properties of a set of entities. A directed graph represents dependencies among the RVs. Since an MFrag is in fact a template that can be repeatedly instantiated to form *Situation-Specific Bayesian Networks* (SSBNs), their RVs usually contain as arguments one or more ordinary variables, which are variables that are substituted by instances of entities during the instantiation process. SSBNs are regular BNs that are formed, usually in response to a query, to address a particular situation that may occur in the domain. Since a SSBN is just a regular BN, traditional BN algorithms, like junction tree algorithm [22], can be applied to it with no special adaptations. Usually, a SSBN would look like a collection of "similar" nodes, differing only by their arguments' values.

MEBN provides a compact way to represent repeated structures in a Bayesian Network. An important advantage of MEBN is that there is no fixed limit on the number of random variable instances, which can be dynamically instantiated as needed. Some may see MFrags as tiny "chunks of knowledge" of a given domain. Since a MTheory is a consistent composition of such "chunks", MEBN (as a formalism) is suitable for use cases addressing reuse of information. This property is used in this work in order to achieve efficient reuse of ontology.

Finally, MEBN categorizes random variables into three different types. See Figure Fig 1 for a graphical representation. Directed arrows going from parent to child variables represent dependencies. The list of arguments in parenthesis are replaced by unique individuals when the SSBN instantiation process is triggered. The following list describes the elements presented in Fig 1:

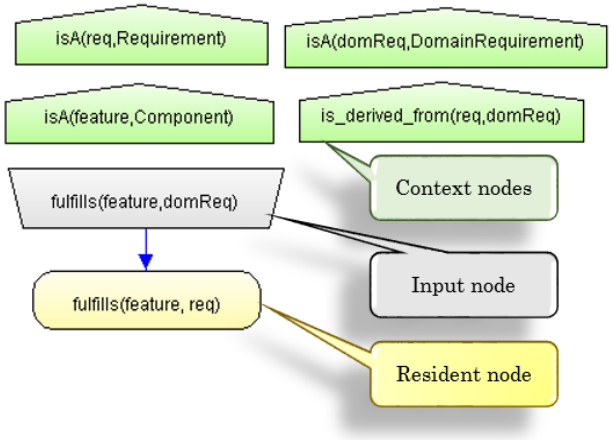


Fig 1. Structure of MEBN Fragment.

- *Resident nodes* (rounded yellow rectangles) are predicates (as in First-Order Logic) which represent the actual random variables that form the core subject of an MFrag. MEBN logic requires that the local probabilistic distribution of each resident node should be uniquely and explicitly defined in its home MFrag. The possible values of a resident node can be instances of entities (*e.g.* individuals of an OWL class). In this example, the resident node "*fulfills*" represents a relationship between a feature and a set of requirements (of any type) that the feature satisfies/fulfills.
- *Context nodes* (green pentagons) are Boolean (*i.e.* logical datatype) random variables representing conditions that must be satisfied to make a distribution in an MFrag valid. First-Order Logic formula (which may reference predicates in other MFrags) can be used in order to express complex conditions. For instance, the context node *is_derived_from(req,domReq)* indicates that the MFrag is only valid if *req* (a requirement) is derived from *domReq* (a domain requirement). Any combination of *req* and *domReq* not satisfying the context node will cause the instances of the nodes in that MFrag to be marked as invalid and thus some default probability distribution (instead of the distribution specified in the MFrag) will be applied.
- *Input nodes* (grey trapezoids) are basically "pointers" referencing to some resident node. Input nodes also provide a mechanism to allow resident nodes' re-usage between MFrags. In the example, the input node *fulfills(feature, domReq)* is a reference to the resident node *fulfills* in the same MFrag. The arc from *fulfills* input node to *fulfills* resident node (*i.e.* the recursive dependency) indicates that whether a feature fulfills or not some requirement depends on whether the feature fulfills or not a domain requirement which derived the requirement in question.
- *Ordinary variables* appear as arguments of nodes in the example (see labels *feature*, *req*, and *domReq*). They are "non-random" variables that can be replaced with instances

[1] The language offers means for specifying or extending information or rules about other elements in the ontology.

of entities in order to fill the arguments of nodes. Constraints about the type of ordinary variables are declared in "*isA*" context nodes, whose first argument is an ordinary variable and the second argument is a name of some entity (*e.g.* some OWL class).

III. Multi-Entity Decision Graph

Multi-Entity Decision Graph (MEDG) provides a framework for modeling and solving decision problems which require both first-order expressiveness and handling of uncertainty; and it forms the semantics, mathematical formalism, and a graphical abstraction of documents written in PR-OWL Decision. Consequently, in a technical view, PR-OWL Decision documents can be seen as a computer-readable representation of MEDG models that can be persisted in storage media or streamed to a network.

MEDG extends MEBN by combining the expressiveness of a probabilistic First-Order Logic—MEBN—with the ability to represent decisions and values (utilities) and to perform decision making under uncertainty, with maximum expected utility criterion, of Influence Diagrams (ID) [16]. IDs are a generalization of Bayesian Networks (BN) [21] which consist of a directed acyclic graph of *probabilistic nodes* (just like nodes in BN, it corresponds to random variables), *decision nodes* (they correspond to decisions to be made, and represent available actions), *utility nodes* (corresponds to utility functions, which quantifies values or preferences of a decision maker), *conditional arcs* (arcs that points to a probabilistic node and represent probabilistic dependence), *information arcs* (arcs that points to decision nodes and represent information that have to be available at the time of the decision), and *functional arcs* (arcs that points to utility nodes and represent inputs for the utility function). The main idea of MEDG is, therefore, to augment MEBN with decision nodes, utility nodes, information arcs and functional arcs.

Following the convention of MEBN, the world is represented in MEDG as a collection of inter-related entities, their respective attributes, and relations among them. Knowledge about attributes of entities and their relationships is represented as a collection of network fragments that represent repeatable patterns, known as *MFrags* (now, this name stands for *MEDG Fragments* instead of *MEBN Fragments*). A set of well-defined MFrags that collectively satisfies logical constraints is called *MTheory* (similarly, this name now stands for *MEDG Theory*). A consistent PR-OWL Decision ontology represents an MTheory. Fig 2 shows the components of a MEDG Fragment, and the following list is a description of such components:

- **Decision resident node**: this orange rectangular node is a new type of node in MEDG and it represents the class of decision nodes. It can be used in input nodes or context nodes, and just like resident nodes it needs to be uniquely and explicitly defined in some home MFrag. As in IDs, arcs pointing to these nodes are information arcs that represent information that are assumed to be known at the time of taking the action. In the example, *incorporateFeature* represents the decision of whether to add or not some feature "*feat*" to the current configuration "*config*", given

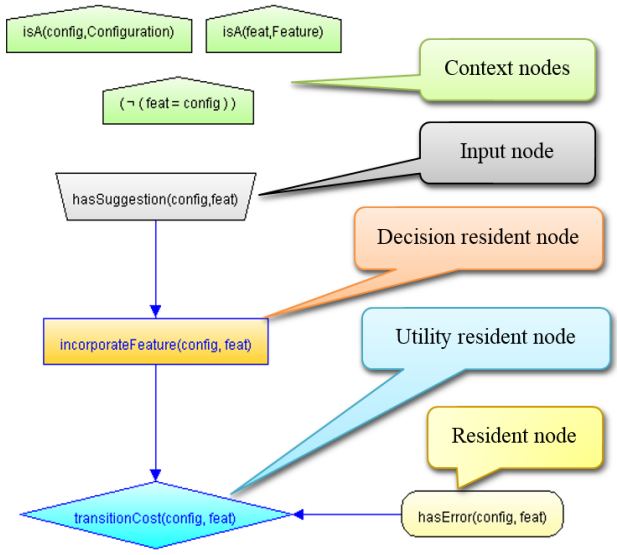


Fig 2. Structure of MEDG Fragment.

information of *hasSuggestion* (whether such feature can be suggested to the configuration or not).

- **Utility resident node**: this blue diamond node is a new type of node in MEDG which represents the class of utility nodes. MEDG logic requires that the utility function of a utility resident node must be uniquely and explicitly defined in some home MFrag. Utility resident nodes cannot be parents of resident nodes or decision resident nodes, and cannot be used in context nodes. Arcs pointing to these nodes are functional arcs and represent inputs of the utility function. Under the multi-attribute utility criteria, we can represent the "global" utility function as a combination of sub-functions (*i.e.* the utility function can be decomposed to multiple sub-functions involving only a smaller subset of variables, and each of such sub-functions can be represented by utility resident nodes). In such context, when some utility resident node is a child of utility resident nodes, it represents the combining function over the parents. If no such combining function is specified, then the unweighted additive function (*i.e.* a simple sum over the sub-functions) is implicitly assumed by default. In Fig 2, *transitionCost* represents the cost of adding the feature "*feat*" to the current configuration "*config*" (given the decision about whether to actually add or not such feature).

- **Resident node** (or "probabilistic" resident node), **input node**, **context node**, and **ordinary variables**: these elements play the same role as in MEBN. However, input and context nodes can now have references to Decision resident nodes. The three context nodes in Fig 2 are declaring that the type of the ordinary variable *config* and *feat* are respectively the *Configuration* and *Feature* entities, and the values of these ordinary variables must not be equal. The input node *hasSuggestion* is a reference to a resident node in another MFrag (not shown in the figure, though). The resident node *hasError* is the probability of the new feature *feat* to cause error to current configuration *config*, and it has direct impact on the utility.

From a semantic viewpoint, backward compatibility (*i.e.* tools that support MEDG should also support MEBN) is only possible if MEDG models without presence of decision and utility nodes are equivalent to the respective MEBN model. This explains why components of MEBN (*e.g.* resident nodes, input nodes, context nodes) are fully reused in MEDG. It is worth noting that these approaches for backward compatibility are directly applicable to PR-OWL Decision as well, because PR-OWL Decision ontologies semantically represent MEDG models, and they share the same abstractions (*i.e.* nodes, entities, states, *etc*.).

On the other hand, forward compatibility (*i.e.* tools that support MEBN should be able to open MEDG models) is not directly guaranteed at the logic level, obviously because MEBN semantics cannot handle decision and utility nodes. Instead, forward compatibility is achieved at the syntactical level in PR-OWL Decision by asserting that decision resident nodes and utility resident nodes in PR-OWL Decision are subclasses of resident nodes of PR-OWL. This shall enable tools compatible with PR-OWL to open PR-OWL Decision ontologies, and allow decision and utility nodes to be displayed and edited as if they were just resident nodes. This is why decision resident nodes and utility resident nodes in PR-OWL Decision are defined respectively as resident nodes with no probability distribution, and single-valued resident nodes in PR-OWL Decision.

A. Entailments of PR-OWL Decision: MEDG Inference

Entailments of PR-OWL Decision are information that can be inferred from a PR-OWL Decision ontology document, based on its underlying semantics—MEDG. This includes anything that can be deterministically inferred (by First-Order Logic or its subsets), anything that can be inferred by first-order probabilistic reasoning (which requires combination of First-Order Logic and probabilistic inference), and anything that can be inferred by combining the previous inference with decisions and utility functions. The former two can be achieved with MEBN and PR-OWL (actually, the first one can even be achieved with OWL direct semantics and description logic reasoning), so they are not important in the context of this document. The last one is our focus, because it requires inference in MEDG semantics.

Namely, the tasks of calculating expected utility, and to find optimal policy under maximum expected utility criterion are important entailments of PR-OWL Decision that will be considered in this research. We propose an algorithm (described in Listing 1) adapted from [23] for grounding a MEDG Theory based on entity information and evidence currently available in the knowledge/data base (in the context of PR-OWL Decision, the knowledge/data base is the ontology itself, or it can be a separate ontology, but consistent with PR-OWL Decision) to generate a Situation-Specific Influence Diagram (SSID) in order to solve the above tasks.

Fig 3 illustrates grounded inference of MEDG in the context of PR-OWL Decision. In the figure, data/evidences retrieved without probabilistic inference (*e.g.* OWL individuals or OWL property assertions) will be combined with elements of MEDG in order to instantiate the SSID. Once SSIDs are generated, they are equal to ordinary IDs, so any algorithm for solving (*e.g.* calculate expected utility or find optimal policy) IDs can be used to solve SSIDs.

Inputs:
- Queries: a list of nodes (instances of decision or resident nodes) that will be guaranteed to be present in SSID.
- Instances of entities: collection of all known instances of entities. These can be OWL individuals in PR-OWL Decision.
- Evidence: list of all random variables and decision nodes with known values (and their respective values as well).

1 Include all nodes in evidence and queries in SSID.
2 Include all possible instantiations of utility nodes (by instantiating all possible values of arguments of utility resident nodes) to SSID.
3 Mark all nodes in SSID as "unfinished".
4 For each "unfinished" node "n" in SSID, do:
  4.1 Find the resident node (or decision/utility resident node) "res" whose "n" is its instance.
  4.2 If the MFrag of "res" is marked as "unsatisfiable", set "n" to use default distribution, mark "n" as "finished", and continue at line 4.
  4.3 For each context node "cx" in the same MFrag
    4.3.1 If "cx" is unsatisfiable (*i.e.* 100% false), then mark the MFrag as "unsatisfiable", set "n" to use default distribution, mark "n" as "finished", and continue at line 4.
    4.3.2 Else if "cx" is unknown (*i.e.* neither 100% true or 100% false), then:
      4.3.2.1 Virtually transform the context "cx" to input node.
      4.3.2.2 Create arcs from new input node to all resident nodes (and decision nodes) in same MFrag.
  4.4 For each parent "p_res" of "res", do:
    4.4.1 Instantiate arguments (ordinary variables) of "p_res" that match the formulae in context nodes in the same MFrag.
    4.4.2 Instantiate "p_res" with the combination of arguments found in previous step.
    4.4.3 For each instance "p_n" of "p_res", do:
      4.4.3.1 Mark "p_n" as "unfinished", and add it to SSID (if not already there).
      4.4.3.2 Add arcs from "p_n" to "n" in the SSID.
  4.5 Mark "n" as "finished".
5 Prune (remove) from SSID all nodes that are d-separated or disconnected from queries and utility nodes.
6 Compile the LPD/utility scripts of all probabilistic and utility nodes, so that the scripts are translated to actual probability distributions/tables or actual utility functions/tables.
7 Return (output) SSID.

Listing 1: pseudocode for generating SSID.

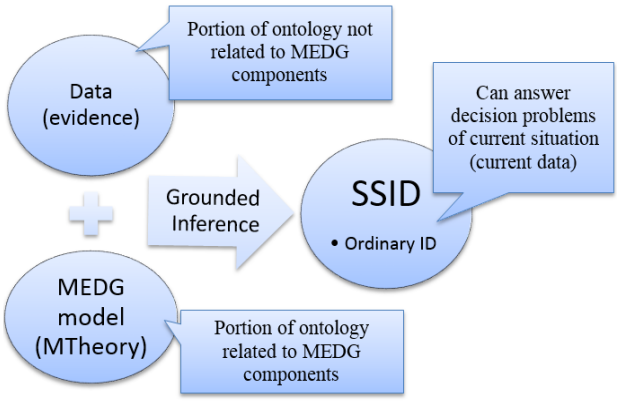


Fig 3. Grounded inference of MEDG.

B. A Script Language for Utility and Probability Distribution

A resident node in MEDG specifies a *Local Probability Distribution* (LPD), a generic specification of conditional probabilities of random variables that can be instantiated from that resident node, given their parents. However, since MEDG represents generalizations, LPDs cannot be specified in a "propositional" manner, like a table of conditional probabilities for all possible combinations of parents' states. Similarly, utility functions of utility resident nodes also cannot be specified in a "propositional" manner.

We propose a scripting language for specifying LPDs and utility functions in MEDG in a uniform and "non-propositional" manner, by extending the scripting language of [23, pp. 17-18] with more support for first-order syntax, such as support for ordinary variables in conditions, support for arguments in nodes, more support for nodes with states dynamically instantiated, and support for non-normalized values (for utilities, which do not necessarily sum up to 1). Special care was taken for backward compatibility, so that old scripts are also valid in the new grammar.

Listing 2 shows a tentative version of the new grammar in Backus–Naur Form [24] for a script for specifying utility and LPD. Listing 3 is an example of LPD script that complies with the proposed LPD grammar (it specifies the probability distribution of node *fulfills* of Fig 1).

Table I is an example of a conditional probability table that can be generated from Listing 3, when SSID is instantiated. In this example, the ordinary variable "*feature*" was substituted by an entity instance called "*F1*", and the ordinary variable "*domReq*" (*i.e.* a domain requirement) was substituted by entity instances "R1" and "R2". We can see in the table that if at least one parent is true, then the probabilities are set to true = 0.7, and false = 0.3. When no parent is true, but at least one parent is false, then the probabilities are set to true = 0.1, and false = 0.9. Otherwise, the probability of absurd is set to 1. This complies with Listing 3.

Scripts for specifying LPDs are not formally part of PR-OWL, so such scripts are directly stored as literal data properties. We will follow the same approach and store scripts in the new grammar as literal (text) data properties in PR-OWL Decision as well. Consequently, the new LPD scripting language is not formally a part of the specification of PR-OWL Decision.

```
<distribution> ::= <statement> | <if_statement>
<if_statement> ::=
            "if" <allop> <varsetname>
            "have" "(" <b_expression> ")" <statement>
            "else" <else_statement>
<allop> ::= "any" | "all"
<varsetname> ::= <ident>[["."|","]<ident>]*
<b_expression> ::= <b_term> [ "|" <b_term> ]*
<b_term> ::= <not_factor> [ "&" <not_factor> ]*
<not_factor> ::= [ "~" ] <b_factor>
<b_factor> ::= "(" <b_expression> ")"
            | <ident> ["(" <arguments> ")"]
            "=" <ident> ["(" <arguments> ")"]
<arguments> ::= <ident>[["."|","]<ident>]*
<else_statement> ::= <statement> | <if_statement>
<statement> ::= "[" <assignment_or_if> "]"
<assignment_or_if> ::= <assignment> | <if_statement>
<assignment> ::= <ident> "=" <expression> [ "," <assignment> ]*
<expression> ::= <term> [ <addop> <term> ]*
<term> ::= <signed_factor> [ <mulop> <signed_factor> ]*
<signed_factor> ::= [ <addop> ] <factor>
<factor> ::= <number> | <function> | "(" <expression> ")"
<function> ::= <possibleVal>
            | "CARDINALITY" "(" [<varsetname>] ")"
            | "MIN" "(" <expression> ";" <expression>")"
            | "MAX" "(" <expression>";" <expression>")"
            | <external_function>
<possibleVal> ::= <ident>
<addop> ::= "+" | "-"
<mulop> ::= "*" | "/"
<ident> ::= <letter> [ <letter> | <digit> ]*
```

Listing 2: BNF grammar of LPD/utility script.

```
if any feature,domReq have ( fulfills(feature,domReq) = true ) [
  true = .7, false = .3
] else if any feature,domReq have ( fulfills = false ) [
  true = 0.1, false = 0.9
] else [ absurd = 1 ]
```

Listing 3: Example of LPD script.

IV. PR-OWL Decision

PR-OWL Decision, the language proposed in this research, extends PR-OWL in order to support decision variables (*i.e.* actions that a decision maker can take) and utility variables (*i.e.* values and preferences) in probabilistic ontologies.

The new language provides definitions of special classes and properties (relationships) that collectively form a framework for building and reasoning with decision problems expressed as probabilistic ontologies. These new components are defined in terms of existing PR-OWL and OWL elements, so that syntactical compatibility with PR-OWL (and OWL) is achieved. In this chapter we define such new components and how they relate to PR-OWL and OWL.

We primarily extend PR-OWL version 2 (PR-OWL 2), because it offers enhanced meta-level features—not present in version 1—that allows us to represent probability distributions

TABLE I. EXAMPLE OF CONDITIONAL PROBABILITY TABLE THAT CAN BE OBTAINED FROM SCRIPT IN LISTING 3.

Parents	fulfills (F1, R1)	true			false			absurd		
	fulfills (F1, R2)	true	false	absurd	true	false	absurd	true	false	absurd
Child's states	true	0.7	0.7	0.7	0.7	0.1	0.1	0.7	0.1	0
	false	0.3	0.3	0.3	0.3	0.9	0.9	0.3	0.9	0
	absurd	0	0	0	0	0	0	0	0	1

of existing OWL properties [8]. These features are necessary conditions for semantic-level compatibility with OWL, because they enable entailments of OWL ontologies to be also contained in the entailments of PR-OWL 2. We also offer an alternative extension of PR-OWL version 1 (PR-OWL 1) for decision support in ontologies originally written in this older version as well. However, this is only kept for backward compatibility, and is superseded by the extension of PR-OWL 2. The version of PR-OWL Decision which extends PR-OWL 2 is called PR-OWL 2 Decision, and the version that extends PR-OWL 1 is called PR-OWL 1 Decision; but for simplicity, in this document We'll simply use "PR-OWL Decision" to refer to the one that extends PR-OWL 2.

A. PR-OWL Decision Schema Vocabulary

Just like any OWL and PR-OWL document, a PR-OWL Decision document needs to be built by combining a set of pre-defined building blocks. A PR-OWL Decision document is said to be syntactically valid if the document is validated against a schema vocabulary. A schema vocabulary is a document that partially defines another document's structure with a list of legal elements, attributes, built-in classes and properties.

Fig 4 illustrates how the PR-OWL Decision schema vocabulary relates to other vocabularies. The vocabulary (schema) files of PR-OWL 1 Decision and PR-OWL 2 Decision reuses constructs from PR-OWL 1 and PR-OWL 2 respectively. While the vocabularies of PR-OWL are valid ontologies in OWL direct semantics (thus, we can use the OWL "import" mechanism to reuse the entire document), the OWL RDF/XML syntax vocabulary file/document has some constructs that are not defined in OWL direct semantics, so only a subset of OWL vocabulary document is used in PR-OWL vocabulary. Finally, as the name implies, the OWL RDF/XML syntax document combines syntaxes from XML (and XML Schema) and Resource Description Framework (RDF) and its schema (RDFS) [25].

From the foundation of OWL, any ontology component is identified by an Internationalized Resource Identifier (IRI), a standard defined by the Internet Engineering Task Force to extend the Uniform Resource Identifier (URI) scheme. URIs and IRIs are both text identifiers that resemble web addresses, but URIs are limited to ASCII characters, while IRIs allow Unicode characters to be used. The stereotype <<*owl:imports*>> in arcs represents a property that is used for importing other OWL ontologies entirely. The World Wide Web Consortium (W3C) recommends not to import the OWL schema vocabulary directly to ontologies using direct semantics of OWL, because it will break some compatibility with Description Logic. Therefore, the stereotype <<*uses*>> indicates that only a subset of features are referenced. The stereotype <<*Definition*>> is used instead of <<*Vocabulary*>> in XML Schema Definition (XSD) simply because the word "definition" is part of its official name.

In the syntax viewpoint, backward compatibility with PR-OWL is forced because we explicitly import the PR-OWL schema vocabulary into the new schema (thus, tools compatible with PR-OWL Decision are forced to handle PR-OWL schema as well). Forward compatibility (*i.e.* tools compatible with OWL or PR-OWL will be able to open PR-OWL Decision documents—but not necessarily execute some reasoning process) is achieved because PR-OWL Decision schema vocabulary only uses building blocks of OWL and PR-OWL, and the PR-OWL schema vocabulary only uses building blocks compatible with OWL's RDF/XML syntax and vocabulary—thus the entire import closure is forward compatible.

B. Syntactical Differences with PR-OWL

PR-OWL Decision introduces the concept of decision nodes and utility nodes to PR-OWL. No changes will be made to existing syntactical blocks of PR-OWL, which will be fully reused—imported—by the PR-OWL Decision. Fig 5 illustrates the classes of PR-OWL 2 Decision and their relationships to PR-OWL 2 classes. Fig 6 illustrates the classes of PR-OWL 1 Decision and their relationships to PR-OWL 1 classes. The remaining paragraphs of this section basically discusses about the contents of the figures.

The prefixes of IRIs of classes in PR-OWL 2 Decision are the IRIs of its schema vocabulary (*i.e.* IRIs of these classes starts with the IRI of the schema vocabulary of PR-OWL 2 Decision, and the IRI fragment—suffix after "#"—is the name of the class). Similarly, prefixes of IRIs of classes in PR-OWL 1 Decision are the IRIs of the schema vocabulary of PR-OWL 1 Decision. For example, the IRI of class *DomainDecisionNode* of PR-OWL 2 Decision is <*http://www.pr-owl.org/pr-owl2-*

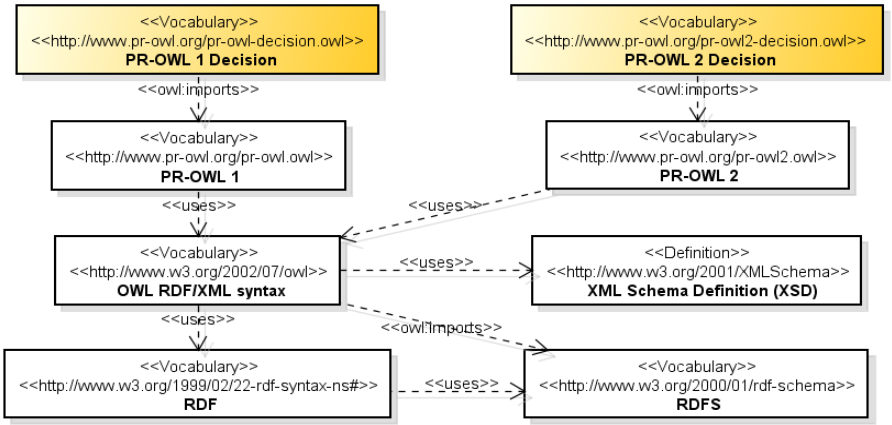


Fig 4. IRI/URI of vocabularies.

decision.owl#DomainDecisionNode>. The IRIs of the other classes follow the same pattern, so IRIs are omitted in the figures for sake of visibility. These classes can be mapped to components of its underlying logic—Multi-Entity Decision Graph (MEDG)—which is presented in later section.

The following list describes the main elements of PR-OWL 2 Decision in Fig 5—again, please refer to the section about Multi-Entity Decision Graph for the semantics of these elements:

- *DomainDecisionNode*: this class represents decision resident nodes of MEDG (see next section for descriptions about MEDG). It extends *DomainResidentNode*, a class which represents resident nodes in PR-OWL, because all properties that are valid for the *DomainResidentNode* (for instance, it should be associated with possible values, can have parents and children, and can be used as arguments of other nodes) are also valid for *DomainDecisionNode*, except for the fact that LPDs are not used in *DomainDecisionNode*.
- *DomainUtilityNode*: this class represents utility functions (utility resident nodes in MEDG). This is represented as subclass of *DomainResidentNode* for forward compatibility, so that tools compatible with PR-OWL can open utility nodes as if they were resident nodes with a single possible value (the *utility* instance).
- *UtilityMExpression*: this is an extension of *MExpression* of PR-OWL 2 for *DomainUtilityNode*. The *MExpression* connects *Node* to its arguments, types, or possible values, and *UtilityMExpression* specifies some restrictions that force *DomainUtilityNode* not to be used in arguments of context nodes (this is achieved by "*isTypeOfArgumentIn exactly 0 MExpressionArgument*" restriction), and by forcing the type of *DomainUtilityNode* to be always *UtilityVariable*.
- *UtilityVariable*: this is an extension of *RandomVariable*, a class which describes the type of *MExpression*. *UtilityVariable* is used to force *DomainUtilityNode* to be associated with only a single possible value: the *utility*. This asserts that tools compatible with PR-OWL will see instances of *DomainUtilityNode* as being resident nodes with a single value.
- *utility*: this OWL individual is a possible state of *DomainUtilityNode* created for compatibility with PR-OWL (thus, this OWL individual does not actually represent the "concept" of utility), because constraints in PR-OWL forces any node to have at least one possible state. Please, notice that numerical values of utilities in PR-OWL Decision are represented in terms of utility functions, not by some OWL individual or literal called "utility". This is similar to the approach in PR-OWL for probabilities, because such values are represented as probability distributions, not by some individual or literal called "probability".

The following list describes the elements of PR-OWL 1 Decision (Fig 6) and compares them with PR-OWL 2 Decision:

- *Domain_Decision*: same of *DomainDecisionNode* of PR-OWL 2 Decision.
- *Domain_Utility*: same of *DomainUtilityNode* of PR-OWL 2 Decision. The "*isArgTermIn exactly 0 ArgRelationship*" forces *Domain_Utility* not to be used as arguments in context

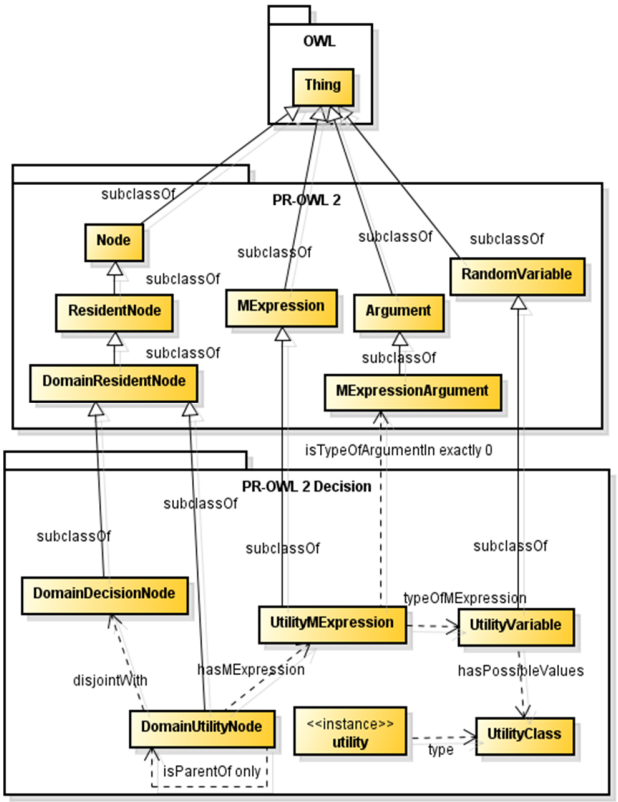


Fig 5. PR-OWL 2 Decision classes and relations to PR-OWL 2.

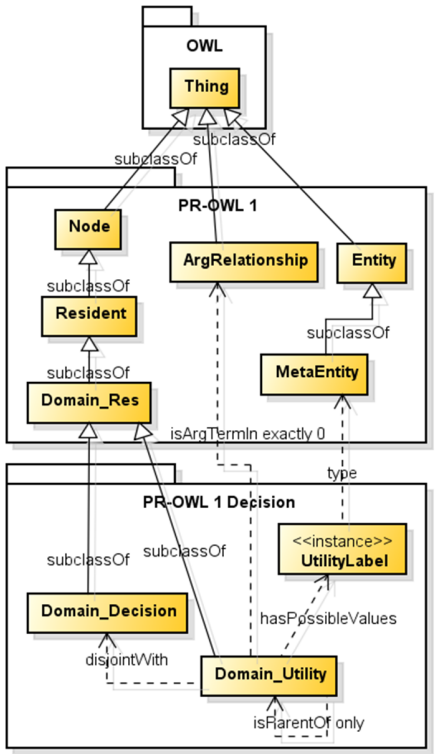


Fig 6. PR-OWL 1 Decision classes and relations to PR-OWL 1.

nodes; and the restriction "*hasPossibleValues utility*" enables tools compatible with PR-OWL 1 to see *Domain_Utility* as a resident node with single value.

- *UtilityLabel*: this is the same of *utility* in PR-OWL 2 Decision. The difference is that a constraint in PR-OWL 1 forces a possible state of a node to be an individual of *Entity*, while in PR-OWL 2 this constraint is relaxed. For this reason, utility in PR-OWL 1 Decision is an individual of *MetaEntity*—subclass of *Entity*.

V. Conclusion and Future Work

PR-OWL Decision was formulated as an extension to PR-OWL in order to support decision making under uncertainty. Backward and forward compatibility was ensured by reusing both syntax and semantic elements from PR-OWL. MEDG, the underlying logic of PR-OWL Decision, augments MEBN with decision and utility variables, so that entailments of PR-OWL Decision can be obtained with MEDG inference. An example of grounded inference/solving algorithm and a script for specifying probabilities and utilities in MEDG was described in this document. This work is part of an ongoing Ph.D. research, thus further details on MEDG, related algorithms, and software implementations will be coming in future works.

Acknowledgment

We thank UnBBayes development team of Universidade de Brasília, especially professor Marcelo Ladeira, M.S. student Laécio Santos, undergraduate students Guilherme Torres, Diego Marques, Rafael Martins, and Pedro Abreu for their insights and assistance with software development.

References

[1] R. N. Carvalho, P. C. G. Costa, K. B. Laskey and K. C. Chang, "PROGNOS: predictive situational awareness with probabilistic ontologies," in *In Proceedings of the 13th International Conference on Information Fusion*, Edinburgh, UK, July 2010.

[2] P. Mitra, N. F. Noy and A. R. Jaiswal, "Omen: A probabilistic ontology mapping tool," in *In The Semantic Web–ISWC 2005*, 2005.

[3] T. Tudorache, "Employing ontologies for an improved development process in collaborative engineering," 2006.

[4] H. H. Wang, Y. F. Li, J. Sun, H. Zhang and J. Pan, "Verifying feature models using OWL," *In Journal of Web Semantics,* vol. 5, no. 2, p. 117–129, June 2007.

[5] O. Udrea, D. Yu, E. Hung and V. S. Subrahmanian, "Probabilistic ontologies and relational databases," in *On the Move to Meaningful Internet Systems*, 2005.

[6] J. Carroll, I. Herman and P. F. Patel-Schneider, "OWL 2 Web Ontology Language (Second Edition)," 11 December 2012. [Online]. Available: https://www.w3.org/TR/owl2-rdf-based-semantics/. [Accessed 20 July 2016].

[7] P. C. G. Costa, "Bayesian Semantics for the Semantic Web," 2005.

[8] R. N. Carvalho, K. B. Laskey and P. C. G. Costa, "PR-OWL 2.0-bridging the gap to OWL semantics," in *In Proceedings of the 6th International Conference on Uncertainty Reasoning for the Semantic Web*, November 2010.

[9] E. Acar, C. Thorne and H. Stuckenschmidt, "Towards Decision Making via Expressive Probabilistic Ontologies," in *In Proceedings of the 4th International Conference, ADT 2015*, Lexington, KY, USA, 2015.

[10] C. Guestrin, D. Koller, C. Gearhart and N. Kanodia, "Generalizing plans to new environments in relational MDPs," in *In Proceedings of the 18th international joint conference on Artificial intelligence*, 2003.

[11] S. Joshi, K. Kersting and R. Khardon, "Generalized First Order Decision Diagrams for First Order Markov Decision Processes," in *In International Joint Conference on Artificial Intelligence*, 2009.

[12] S. Sanner, "Relational dynamic influence diagram language (RDDL): Language description," Australian National University, 2010.

[13] C. Wang, S. Joshi and R. Khardon, "First order decision diagrams for relational MDPs," in *Journal of Artificial Intelligence Research*, 2008.

[14] H. L. Younes and M. L. Littman, "PPDDL1. 0: An extension to PDDL for expressing planning domains with probabilistic effects," Technical report. CMU-CS-04-162, 2004.

[15] D. Poole, "The independent choice logic for modelling multiple agents under uncertainty," *Artificial Intelligence,* vol. 94, no. 1, pp. 7-56, 1997.

[16] R. A. Howard and J. E. Matheson, "Influence diagrams," in *In Readings on the Principles and Applications of Decision Analysis II*, 1984/2005.

[17] I. Horrocks, B. Parsia and U. Sattler, "OWL 2 Web Ontology Language Direct Semantics (Second Edition)," 11 December 2012. [Online]. Available: https://www.w3.org/TR/owl2-direct-semantics/. [Accessed 20 July 2016].

[18] K. B. Laskey, "MEBN: A language for first-order Bayesian knowledge bases," *Artificial intelligence,* vol. 172, no. 2, pp. 140-178, 2008.

[19] S. Matsumoto, K. B. Laskey and P. C. G. Costa, *Probabilistic Ontologies in Domain Engineering,* Washington DC: presented at the Systems Engineering in DC Conference (SEDC), 2016.

[20] K. Pohl, G. Böckle and F. J. van Der Linden, Software product line engineering: foundations, principles and techniques, Springer Science & Business Media, 2005.

[21] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Revised second printing. Morgan Kaufmann, 1988/2014.

[22] F. Jensen, F. V. Jensen and S. L. Dittmer, "From influence diagrams to junction trees," in *In Proceedings of the Tenth international conference on Uncertainty in artificial intelligence*, 1994.

[23] S. Matsumoto, R. N. Carvalho, P. C. Costa, K. B. Laskey, L. L. Santos and M. Ladeira, "There's No More Need to be a Night OWL: on the PR-OWL for a MEBN Tool Before Nightfall," in *Introduction to the Semantic Web: Concepts, Technologies and Applications*, G. P. C. Fung, Ed., iConceptPress, 2011, pp. 267-290.

[24] J. Backus, F. Bauer, J. Green, C. Katz, J. McCarthy, P. Naur, A. J. Perlis, H. Rutishauser, K. Sameison, B. Vauquois, J. H. Wegstein, A. van Wijngaarden and M. Woodger, "Revised report on the algorithmic language Algol 60," *The Computer Journal,* vol. 5, no. 4, pp. 349-367, 1963.

[25] P. Hayes and B. McBride, "RDF Semantics.," 10 February 2004. [Online]. Available: https://www.w3.org/TR/2004/REC-rdf-mt-20040210/. [Accessed 13 May 2016].

[26] D. McDermott, M. Ghallab, A. Howe, C. Knoblock, A. Ram, M. Veloso, D. Weld and D. Wilkins, "PDDL-the planning domain definition language.," Technical report. Yale Center for Computational Vision and Control, 1998.

Sharing Data under Genetic Privacy Laws

Michael Reep*, Bo Yu*, Duminda Wijesekera*, Paulo Costa †
* Department of Computer Science, George Mason University, Fairfax, VA, USA
mreep@gmu.edu, byu3@gmu.edu, dwijesek@gmu.edu
† Department of Systems Engineering and Operations Research, George Mason University, Fairfax, VA, USA
pcosta@gmu.edu

***Abstract*— Clinical medical practice and biomedical research utilize genetic information for specific purposes. Irrespective of the purpose of obtaining genetic material, methodologies for protecting the privacy of patients/donors in both clinical and research settings have not kept pace with rapid genetic advances. When the usage of genetic information is not predicated on the latest laws and policies, the result places all-important patient/donor privacy at risk. Some methodologies err on the side of overly stringent policies that may inhibit research and open-ended diagnostic activity, whereas an opposite approach advocates a high-degree of openness that can jeopardize patient privacy, identifying patient relatives and erode the doctor-patient privilege. As a solution, we present a unique approach that is based on the premise that acceptable clinical treatment regimens are captured in workflows used by caregivers and researchers and therefore their associated purpose can be extracted from these workflows. We combine these purposes with applicable consents (derived from applicable laws) to ascertain the releasability of genetic information. Given that federal, state and institutional laws govern the use, retention and sharing of genetic information, we create a three-level rule hierarchy to apply the laws to a request and auto-generate consents prior to releasing. We prototype our system using open source tools, while ensuring that the results can be added to existing Electronic Medical Records (EMR) systems.**

Keywords—genetic privacy, electronic medical records, ontology, health care, genomic medicine, SWRL

I. INTRODUCTION

Genetic studies match genotypic and phenotypic data to associate genetic markers with onset of diseases [1]. Studies have shown that preventive care costs significantly less than treatment upon disease onset and diagnosis [2, 3]. Furthermore, rapid advancement of genetic research continues to lengthen the list of predictable diseases. Examples include genetic mutations causing some breast cancers (BRC-1 and BRC-2), ovarian cancer, sickle cell anemia, β-thalassemia, left ventricular noncompaction cardiomyopathy and Alzheimer's disease. However, both research and clinical use of genetic information entail privacy challenges that differ from usage of other medical data in following ways:

* Ethics - Privacy of genetic data differs from traditional medical information privacy. For example, protecting patients' private information (e.g., Protected Health Information - PHI) is an important medical ethics and legal obligation. Data for genotype-phenotype matching can be used to stigmatize or discriminate against genetic relatives of a donor, so the dangers of its exposure must be carefully weighed against the benefits of its use [1, 4, 5]. There is an ongoing ethical debate between the two different schools of thought, one in which the donor gives open consent for using his/her data vs. the other that advocates explicit purpose-based consent [6].

* Legal Issues - Due to the unusual situation of being able to expose relative's genetic composition, genetic privacy has been proposed as categorical privacy that differs from traditional individual-centered concepts of privacy in literature [7]. Federal (HIPAA and GINA) [8, 9], state laws and institutional polices provide the legal framework for the sharing of genetic information. Furthermore, genetic privacy laws vary from state-to-state and may be inconsistent with, or more or less stringent than, federal regulations.

* Social Implications - Societal views are often reflected in law and/or organizational policies, so their implications are likely inextricably intertwined with laws and policy governing genetic privacy and what constitutes informed consent.

As a solution, we provide an encompassing framework consisting of workflow-enforced genetic privacy as well as biomedical consent management, consistent with state and federal genetic privacy laws such as statute, regulation and precedent. Following this Introduction, Section 2 addresses related work; Section 3 reviews the prototype design and ontology,

Section 4 describes the implementation of our genetic services workflow that enforces appropriate informed consent based on applicable law to achieve genetic privacy; and, finally, Section 5 presents conclusions.

II. Related Work

Many researchers have suggested adopting traditional information protecting methodologies to protect patients' confidentiality. Yet, this might not be effective due to the uniqueness of being traceable to an individual or group of individuals [10, 11]. After all, some genetic information of an individual may not only precisely identify him/her as high risk of certain hereditary disease(s), but also indicate that his/her relatives have the same risks due to a heritable gene.

Prince et. al. describe three practical genetic counseling cases that illustrate genetic discrimination [12]. The fundamental covenant of protecting patient privacy is embodied in patient-doctor privilege. Conversely, many scholars believe *genetic information is essentially familial in nature and is referred to as the Genetic Information is Familial Thesis (GIFT)* [13], since sharing such information will benefit related groups of individuals. Some countries have regulations to enforce sharing such information among family members [14, 15]. However, many publications discuss and debate the familial approach, with their authors advocating the view that humans possess the rights of privacy and to protect those that *do not want to know* [13, 16]. Conversely, rapid innovations in genetic research require wide accessibility to many genetic databases. The idea of open access in the field of genomic research is expressed in the Bermuda Principles and the Fort Lauderdale Agreement, which has been applied in North America and in the UK for funded research [17]. Genetic research typically requires additional metadata with genetic data sets, such as demographic details family relationships, medical history, etc. These metadata elements can be exploited for tracing an individual's identity.

In general medicine, an informed consent, especially informed privacy consent, provides the proper opportunity and knowledge for patients and research participants to understand and decide how the medical community can use and share their identifiable medical information. Analogously, informed consent tailored for genetic research, clinical usage and counseling constitutes a strong basis for ensuring appropriate genetic privacy. Some genetic medical practices and biomedical research are performed without obtaining appropriate informed consent such as enticing participants in a study without obtaining the proper informed consent. To address this issue, some researchers advocate different methodologies such as using highly-stringent policies to maintain patient confidentiality, but this approach potentially risks limiting scientific innovation [18]. Yet, other researchers have proposed a new, open-consent model for medical and scientific genetic research [7] or open-access policies for genetic data sharing [19]. As the underlying predicate for us undertaking this effort, we proposed a prototype system capable of automatically generating or obtaining appropriate informed consent forms for genetic data sharing under various situations.

EMRs play a vital role of sharing medical information among participating actors based on their usage scenarios. Using EMRs for genetic services present a unique set of challenges [20]. Belmont et al. highlighted the privacy, ethical and legal issues of handling genetic data in EMRs [21]. Scheuner et al. conducted a case study to validate if current EMR systems meet genetic information needs [22]. This study shows an overall lack of support for functionality, structure, and tools for clinical genetic practice. A more recent study of the state of EMRs supporting genomics for personalized medicine identifies structure of data as a challenge [23]. Therefore, it is necessary to implement an informed consent management system in current EMRs.

Some researchers suggested that the legislation for generating and using genetic information properly is pivotal to improving genetic privacy [24]. In 2013, the Health Insurance Portability and Accountability Act of 1996 (HIPAA) [8] Omnibus Rule included genetic information as PHI to be regulated under the privacy portion of HIPAA.

Nonetheless, states may have different definition of genetic information. The combination of Federal privacy laws along with the various state laws form a fragmented regulatory and statutory landscape for permissible information sharing and consent management. To be valid, informed consents for genetic privacy must comply with these laws and regulations. Indeed, significant regulatory gaps create additional burdens in providing automated ways to obtain and generate information consent in EMRs.

III. System Design

We developed a functioning prototype that addresses the various aspects for an automated and integrated informed genetic information consent system. The prototype brings together the data gathered during interactions with the medical provider with the applicable laws, regulations and policies to address the privacy issues specific to genetic information. There are three components of the prototype as shown in Fig. 1:

- Workflow to gather the information, display the outcome and obtain acceptance from the user of the results and any pre/post conditions for using the data.
- A ontological rule-base that takes the data from the workflow, evaluates the applicable laws, determines prerequisites (such as consents and obligations), and decides on the releasability of genetic data.
- A consent service that interacts with the workflow engine and ontology to pass data back and forth. The service includes the Rule Hierarchy Algorithm which combines the outcomes from the three levels (Federal, State and Organization) and provides a final result for permitting or denying access. The outcome includes the consolidated list of conditions for all three levels. For example, the list of consent clauses required by both the Federal regulations and organizational policies.

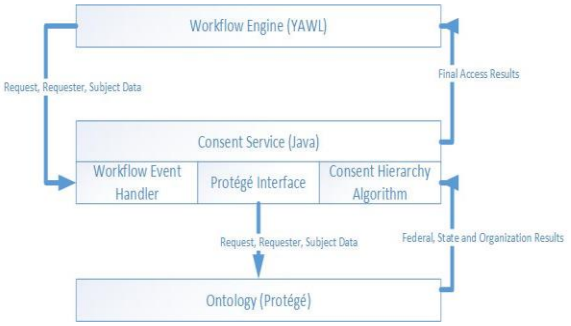


Fig.1. Prototype Components

The first component of implementing the genetic privacy enforcement is to gather the required information through the workflow. As the usage scenario is executed (under the workflow engine) the meta-data required to determine the releasability of data is gathered and passed to the consent service. The consent service then creates the objects and relationships in the ontology for evaluation by the reasoner. Next the service retrieves the results and calls our 3-level rule hierarchical algorithm. The service determines if access is permitted and passes the access results back to the workflow engine. The acknowledgment steps in the workflow display the results along with the decision source (specific law or regulation referenced), the consent clauses, obligations to be enforced for information released, and the specific rules used in the ontology to generate the answer.

To support the consent service, we developed an ontology to capture the various aspects of enforcing privacy laws and policies. As seen in the Fig. 2 the prototype requires four related data items.

- Requester: the person making the request to access the medical information including their role, associations with a specific organization, and information about this organization,
- Request: details on the purpose for requesting the information, and where the information will be used. The four purposes applicable to genetic information are disclosure, research, testing and treatment. The prototype currently implements the information disclosure component with the applicable specific instances for Self-Request by the Patient, Law Enforcement, etc.
- Response: the results of the reasoner applying the appropriate rules along with a list of any obligations that must be enforced by the EMR and specific consent clauses that are needed for the associated approvals. (A subclass for

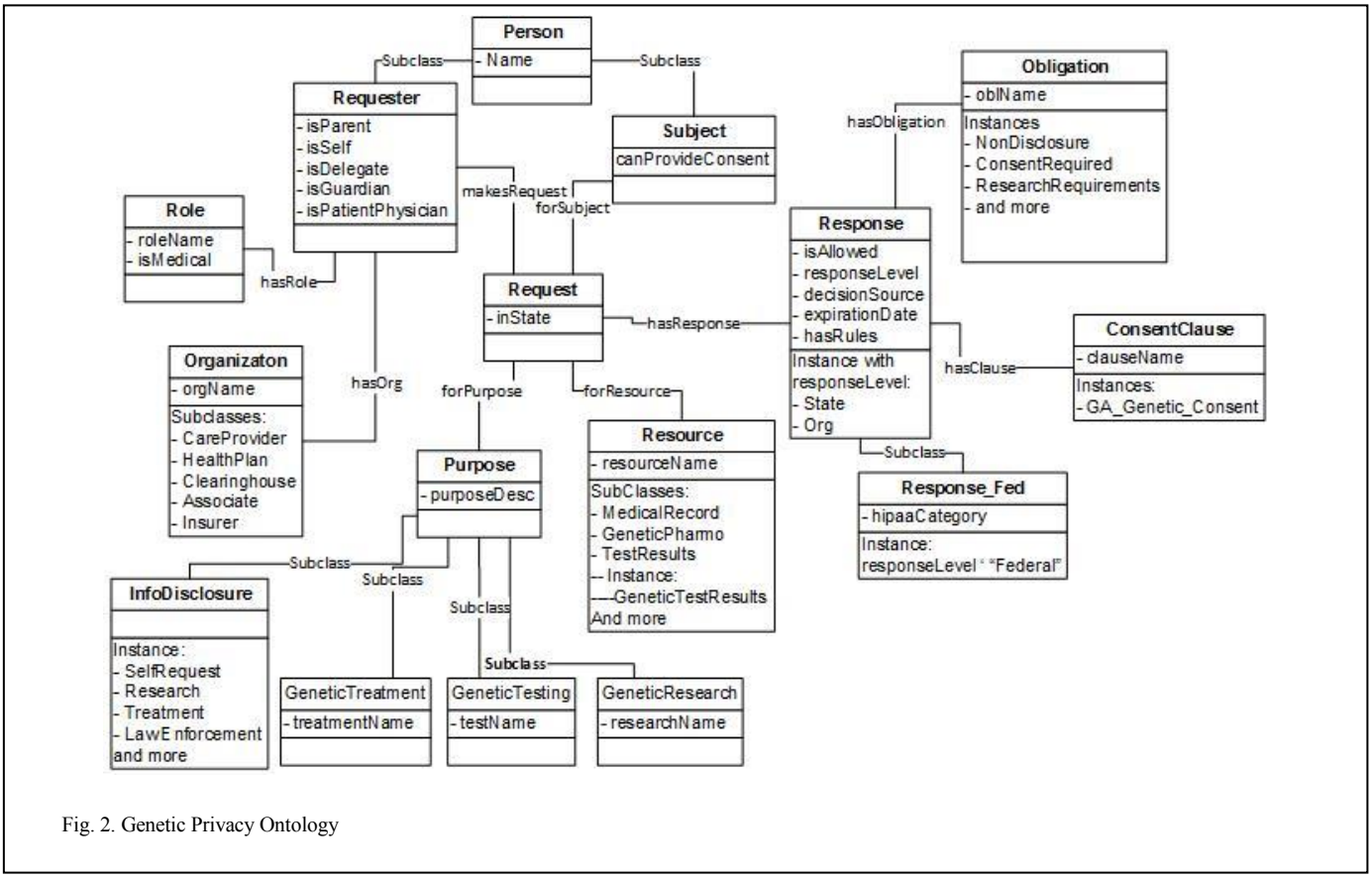


Fig. 2. Genetic Privacy Ontology

Federal Responses allows information about HIPAA-specific requirements to be gathered.)

- Resource: the part of the electronic medical record being requested along with information about the subject (or patient). The Resource instances can be used to categorize detailed levels of rules such as enforcing restrictions to specific parts of the genome that can be used to identify individuals or grant permission to components used in genomic medicine.

The ontology does not need to contain all the information from the EMR because the current focus is on rules implementation. Many entities in the ontology provide reference information such as the organizational meta-data or a list of specific Consent Clauses that are not described presently.

The Rule Hierarchy Algorithm evaluates the interactions between Federal and State laws, regulations and institutional policies. The access evaluation is done at each level (Federal, State and Organization) in the hierarchy that is applicable for the specific access request. By definition, Federal laws are at the top of the hierarchy, followed by State laws, and then organizational policies. The hierarchy algorithm dictates how conflicts between laws and policies can be resolved based the decisions made at each level.

In order to address these potential conflicts, Federal and State laws have an override flag associated with them in the ontology to indicate whether lower level rules can change the answer. If two levels come to the same conclusion (both permit access), the supplemental clauses and obligations are combined into one complete response. For example, HIPAA permits access to medical records for treatment. In Georgia, there are additional obligations and consent requirements when the resource being accessed is from genetic testing.

The Response structure allows both sets of answers to be passed back to the EMR for evaluation and execution. However, if the results were different, the previous answers are discarded in favor of the lower level requirements in order to resolve the inconsistency. For example, if Federal law permitted access and allowed an override to the Permit decision,

the organizational policy may come to a different conclusion and set the response to Deny.

The Rule Hierarchy Algorithm follows:

```
INIT {resAns, resObl, resDec, resCl, resRule} to {fedAns,
fedObl, fedDec, fedCl, fedRule}                          (1)
IF fedOver = true THEN                                   (2)
        IF stAns <> null THEN                            (3)
                IF stAns = fedAns THEN                   (4)
                        resAns = resAns + stAns          (5)
                        resObl = resObl + stObl          (6)
                        resAns = resDec + stDec          (7)
                        resAns = resCl + stCl            (8)
                        resAns = resRule + stRul         (9)
                ELSE                                     (10)
                        resAns = stAns                   (11)
                        resObl = stObl                   (12)
                        resAns = stDec                   (13)
                        resAns = stCl                    (14)
                        resAns = stRule                  (15)
                END IF                                   (16)
        END IF                                           (17)
        IF (orgAns <> null) AND (((stAns <> null) AND
(stOver = true)) OR (stAns = null))) THEN                (18)

                IF orgAns = resAns THEN                  (19)
                        resAns = resAns + orgAns         (20)
                        resObl = resObl + orgObl         (21)
                        resAns = resDec + orgDec         (22)
                        resAns = resCl + orgCl           (23)
                        resAns = resRule + orgRul        (24)
                ELSE                                     (25)
                        resAns = orgAns                  (26)
                        resObl = orgObl                  (27)
                        resAns = orgDec                  (28)
                        resAns = orgCl                   (29)
                        resAns = orgRule                 (30)
                END IF                                   (31)
        END IF                                           (32)
END IF                                                   (33)

RETURN resAns, resObl, resDec, resCl, resRule            (34)
```

In (1) the Result variables for the Answer, Obligations, Decision Source, Clauses and Rules are initialized to the corresponding federal variables, which were retrieved from Protégé. In (2) the Federal Override variable is evaluated to determine whether other rules are to be evaluated. If so, (3) checks for State answer existing and, if found, (4) determines if the Federal and State answer match. Lines (5)-(9) adds the State variables to the Result variables when the Federal and State match while (11)-(15) set the Results variables to the State results when there is no match.

For the Organization level, Line (18) determines if there is an Organization result and whether there is a State result with a State Override flag set to true or there is no State answer. If (18) is true, then (20)-(24) adds the Organization variables to the Result variables, while (26)-(30) set the Results variables to the Organization results. At the end of processing (34) the Results variables are passed back to the workflow via the YAWL API.

IV. System Implementation

The prototype was developed using the YAWL (Yet Another Workflow Language) workflow engine with Java classes that respond to the YAWL event handlers to trigger the ontology processing and Rule Hierarchy Algorithm. As seen in Fig. 3, the consent workflow gathers additional information regarding aspects of the tasks being performed, the requester and the subject before executing a call to the Consent Service in the "Check Consent" step. A final step is provided for validating that the results are acknowledged before returning the response to the associated EMR.

The first YAWL screen shown in Fig. 4 is for the "Get Request Information" step in the workflow process to describe why the request is needed, what part of the medical record is to be accessed, in what state the action is being performed and, for research purposes, whether the request is for an individual or group. Each of the three Get steps have a similar screen. The "AckPermit" screen in Fig. 4 shows the results, pre and post-conditions for using the information, and an input box to enter in acceptance. For an implementation such as an integration with the OpenMRS, these YAWL screens will be replaced with others that will be embedded in the EMR product.

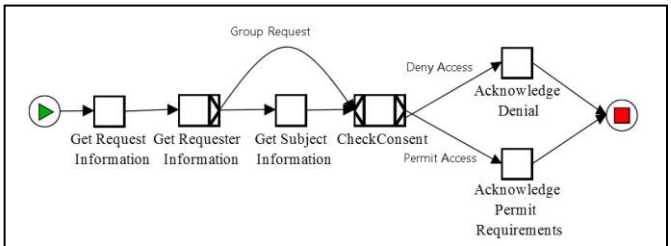


Fig.3. Genetic Privacy Workflow

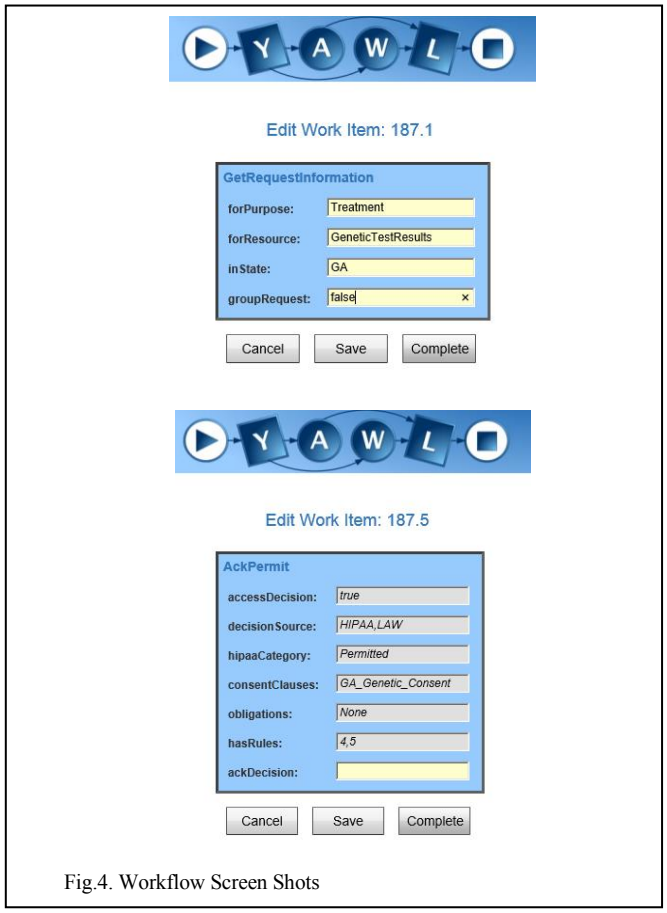


Fig.4. Workflow Screen Shots

Once the consent service is called and the results generated, the latter are displayed for validation by the user. EMR integration will allow some of the tasks, such as generating consent letters, to be implemented and enforced within the product. The Consent Service serves as the integration engine between the workflow/EMR and the ontology. The Java-based Consent Service is triggered by a YAWL event handler on the Check Consent workflow step. The service then gathers all the data from the workflow entries to create and populate the ontology instances including the data and object properties. The object properties link the instances such as establishing the makesRequest relationship between the Requester instance and the Request. Once the data has been populated in the ontology, the reasoner generates the responses and stores the information. The service extracts the response information for evaluation using the Rule Hierarchy Algorithm.

The ontology is implemented using the Protégé platform with the laws and regulations (Federal and State) plus the organization policies enforced via SWRL rules and the Pellet reasoner. The predicate of each rule uses the Request instance with the associated object properties to gather additional information on the Requester, Subject, Purpose and the Resource. (These values were all gathered and populated by the workflow and consent service.) For example, the Request instance is linked in the ontology to the associated Purpose using the hasPurpose object property. The appropriate Response instance (Federal, State or Organization) stores the outcome of the rule regarding whether access is permitted or denied, whether an override is allowed (Federal and State), the HIPAA Category (Federal), the specific law or policy that generated the result, any appropriate obligations and clauses (via hasObligation and hasClause object properties), and a rule number that maps to the SWRL rule.

An example of the implementation is a request to access the Genetic Test Results resource for the Treatment purpose in Georgia. As seen in Fig. 5, there are two different aspects to the Request: establishing relationships to other objects with relevant information and specific data properties for this request. The first object property assertion links the request to the part of the medical record the requester would like to access. The next three object assertions link to response objects that will hold the access permission (permit/deny) and other information associated with the rules for each level (Organization, State and Federal). The next two object assertions link indicate which person is the subject of the request (generally a patient) and the purpose for accessing the medical record. The data

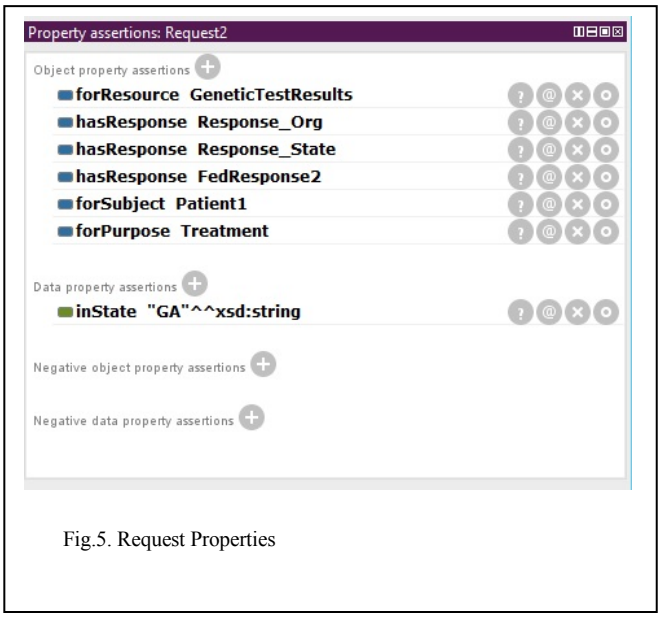


Fig.5. Request Properties

assertion states that the request is being made in the state of Georgia ("GA").

The first SWRL rule below as seen in Protégé addresses the Federal law for access under the Treatment purpose.

makesRequest(?r, ?req), forPurpose(?req, ?pur), purposeDesc(?pur, "Treatment"), hasResponse(?req, ?res), responseLevel(?res, "Federal") -> isAllowed(?res, true), canOverride(?res, true), hipaaCategory(?res, "Permitted"), decisionSource(?res, "HIPAA"), hasRule(?res, 4)

In this example,

- ?r is for the Requester for the Request
- ?pur is the Purpose for "Treatment"
- ?req is the Request being made for the Federal Level with the Treatment Purpose
- ?res is the Federal Response that is associated with the Request.

The explanation for each of these SWRL statements is provided in Table I.

TABLE I. SAMPLE FEDERAL RULE

SWRL Statement	***Explanation***
makesRequest(?r, ?req)	Links Requester to the Request
forPurpose(?req, ?pur)	Links Request with the Purpose
purposeDesc(?pur, "Treatment")	Restricts the rule to only execute for the Treatment purpose description
hasResponse(?req, ?res)	Links the Request with a Response to store answer
responseLevel(?res, "Federal")	Gets the Response for Federal level
-> isAllowed(?res, true)	Sets access to true in Response
canOverride(?res, true)	Sets override to true
hipaaCategory(?res, "Permitted")	Sets HIPAA category to Permitted
decisionSource(?res, "HIPAA")	Sets the decision source as HIPAA
hasRule(?res, 4)	Sets the rule number to 4

When the Pellet reasoner finds a set of instances that matches the Treatment and Federal conditions, the rule is executed and the ?res data properties populated with the values indicated. As seen in Fig. 6, the Federal Response is updated with the final values.

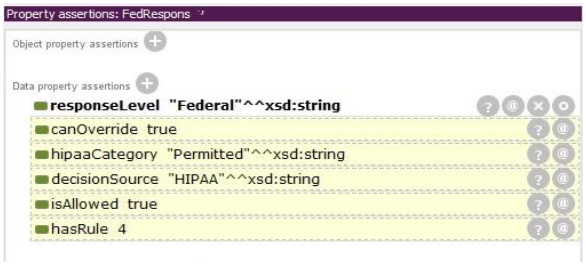


Fig.6. Federal Response

The next part of the example below shows the SWRL rule for the State response, the SWLR statements explained in Table II, and the response in Fig. 7. In the SWRL rule, the predicate sets the location as Georgia and that the rule can be executed if the Federal response allows an Override. The predicate also retrieves an additional obligation for a Consent Agreement and the agreement must have text specific to Georgia. The State response then is set to allow access with no override and information that the decision was based on Georgia Law. The response is linked to an obligation for a Consent Agreement and the consent clause with text specific to Georgia.

isSelf(?r, false), makesRequest(?r, ?req), inState(?req, "GA"), forResource(?req, ?resource), forPurpose(?req, ?pur), purposeDesc(?pur, "Treatment"), resourceName(?resource, "GeneticTestResults"), hasResponse(?req, ?res), responseLevel(?res, "Federal"), canOverride(?res, true), hasResponse(?req, ?resst), responseLevel(?resst, "State"), oblName(?obl, "ConsentRequired"), clauseName(?clause, "GAGeneticConsent") -> isAllowed(?resst, true), canOverride(?resst, false), decisionSource(?resst, "GA_LAW"), hasObligation(?resst, ?obl), hasClause(?resst, ?clause), hasRule(?resst, 5)

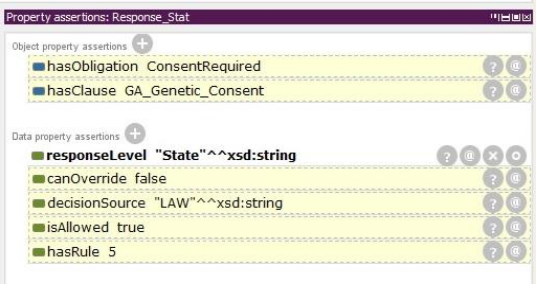


Fig.7. State Response

In the State example, the additional instances used are:

- ?resource is for the "GeneticTestResults" part of the medical record
- ?r is the Requester associated with the Request
- ?obl has the Obligation that ConsentRequired must be obtained for this request
- ?clause indicates the consent agreement for the patient must include the GAGeneticConsent clause
- ?resst is the State response associated with the Request

The explanation for each of these SWRL statements is provided in Table II.

TABLE II. SAMPLE STATE RULE

SWRL Statement	*Explanation*
isSelf(?r, false),)	Verifies Requester is not the subject
makesRequest(?r, ?req),	Links Requester for the Request
inState(?req, "GA"),	Verifies Request is for Georgia
forResource(?req, ?resource)	Links Request with the Resource
forPurpose(?req, ?pur)	Links Request with the Purpose
purposeDesc(?pur, "Treatment"),	Restricts the rule to only execute for the Treatment purpose description
resourceName(?resource, "GeneticTestResults")	Verifies Resource request is for the Genetic Test Results
hasResponse(?req, ?res)	Links the Request with a Response to check previous rule results
responseLevel(?res, "Federal")	Limits the previous Response to Federal
canOverride(?res, true)	Verifies the Federal rule allows overrides
hasResponse(?req, ?resst)	Links the Request with a Response to store answer
responseLevel(?resst, "State")	Gets the Response for State level to store answers
oblName(?obl, "ConsentRequired")	Gets the Obligation for Consent Required
clauseName(?clause, "GAGeneticConsent")	Gets the Clause for Consent Required
-> isAllowed(?resst, true)	Sets the State response to access is allowed
canOverride(?resst, false)	Sets the state Response to not allow override by organization
decisionSource(?resst, "GA_LAW")	Sets the State response to reflect the decision source as state law
hasObligation(?resst, ?obl)	Links the retrieved Obligation with the State response
hasClause(?resst, ?clause)	Links the retrieved Clause with the State response
hasRule(?resst, 5)	Sets the rule number to 5 for reference

When the Pellet reasoner finds a set of instances that matches the Treatment for someone besides the Requester in GA for GeneticTestResults and the Federal response has Override set to True, the rule is executed and the ?resst data properties populated with the values indicated. In addition, the ?obl and ?clause instances are associated with the response as conditions to accessing the record.

V. CONCLUSION

Our prototype brings together the operational data in an EMR workflow for protecting genetic information privacy with the applicable laws, regulations and policies to provide a definitive and consolidated response for access and the associated pre/post conditions for use. Currently, we continue to implement additional Federal and State rules, policies and regulations to develop a comprehensive repository and rule base. The following phase in the prototype will build upon these capabilities for Federal/State laws and regulation enforcement to accommodate the policies and procedures for a selected medical organization. The resulting prototype will demonstrate the overall capabilities needed to meet the medical community's access requirements while balancing the individual rights to privacy and ownership of their genetic medical data.

REFERENCES

[1] M. D. Ritchie, E. R. Holzinger, R. Li, S. A. Pendergrass, D. Kim. "Methods of integrating data to uncover genotype-phenotype interactions." Nature Reviews Genetics 16.2. 2015. 85-97.

[2] A. H. Németh, A. C. Kwasniewska, S. Lise, R. P. Schnekenberg, E. B. Becker, K. D. Bera, ..., & K. Talbot. "Next generation sequencing for molecular diagnosis of neurological disorders using ataxias as a model." Brain 2013. awt236.

[3] C. Pihoker, L. K. Gilliam, S. Ellard, D. Dabelea, C. Davis, L. M. Dolan, ... & E. Mayer-Davis. "Prevalence, characteristics and clinical diagnosis of maturity onset diabetes of the young due to mutations in HNF1A, HNF4A, and glucokinase: results from the SEARCH for Diabetes in Youth." The Journal of Clinical Endocrinology & Metabolism 98.10. 2013. 4055-4062.

[4] W. W. Lowrance, & F. S. Collins. “Identifiability in genomic research.” SCIENCE 317. 2007. 600-602.

[5] A. L. McGuire, & R. A. Gibbs. "No longer de-identified." SCIENCE-NEW YORK THEN WASHINGTON- 312.5772. 2006. 370.

[6] F. D’Abramo, J. Schildmann, & J. Vollmann. "Research participants’ perceptions and views on consent for biobank research: a review of empirical data and ethical analysis." BMC medical ethics 16.1. 2015. 1.

[7] J. E. Lunshof, R. Chadwick, D. B. Vorhaus, & G. M. Church. "From genetic privacy to open consent." Nature Reviews Genetics 9.5. 2008. 406-411.

[8] The Health Insurance Portability and Accountability Act of 1996 (HIPAA). Pub. L. 104-191, 110 Stat. 1936, codi_ed as amended at 42 U.S.C x300gg and 29 U.S.C x1181 et seq. and 42 U.S.C x1320d et seq.

[9] Genetic Information Non-discrimination Act of 2008 (GINA). Pub. L. 110-233, 122 Stat. 883, codified as amended in scattered sections of 26, 29, and 42 U.S.C.

[10] D. Mascalzoni, A. Hicks, P. Pramstaller, & M. Wjst. "Informed consent in the genomics era." PLoS Med 5.9. 2008. e192.

[11] L. O. Gostin, & J. G. Hodge Jr. "Genetic privacy and the law: an end to genetics exceptionalism." Jurimetrics 1999. 21-58.

[12] A. E. Prince and M. I. Roche. "Genetic information, non-discrimination, and privacy protections in genetic counseling practice."Journal of genetic counseling 23.6. 2014. 891-902.

[13] S. M. Liao. "Is there a duty to share genetic information?." Journal of medical ethics 35.5. 2009. 306-309.

[14] A. Lucassen, & J. Kaye. "Genetic testing without consent: the implications of the new Human Tissue Act 2004." Journal of medical ethics 32.12. 2006. 690-692.

[15] American Society of Human Genetics Social Issues Subcommittee on Familial Disclosure.

[16] ASHG STATEMENT Professional Disclosure of Familial Genetic Information. Am. J. Hum. Genet. 62 (1998): 474–483.

[17] E. Sherlock. “disclosure of patient's genetic information without their consent- Is the "public interset" really a Sufficient Justification?.” Genomics Law Report. 2009. retrieved March 2, 2015, from http://www.genomicslawreport.com/index.php/2009/11/10/disclosure-of-patientsgenetic-information-without-their-consent-is-the-public-interest-really-a-sufficient-justification/

[18] J. Kaye, S. M. Gibbons, C. Heeney, M. Parker & A. Smart. "Governing biobanks: Understanding the interplay between law and practice. " Bloomsbury Publishing, 2012

[19] D. Hallinan, & M. Friedewald. "Open consent, biobanking and data protection law: can open consent be ‘informed’under the forthcoming data protection regulation?." Life sciences, society and policy 11.1. 2015. 1.

[20] J. Kaye, C. Heeney, N. Hawkins, J. De Vries, & P. Boddington. "Data sharing in genomics—re-shaping scientific practice."Nature Reviews Genetics 10.5. 2009. 331-335.

[21] D. Mascalzoni, A. Hicks, P. Pramstaller, & M. Wjst. "Informed consent in the genomics era." PLoS Med 5.9. 2008. e192.

[22] J. Belmont, & A. L. McGuire. "The futility of genomic counseling: essential role of electronic health records." Genome medicine 1.5. 2009. 1.

[23] M. T. Scheuner, H. de Vries, B. Kim, R. C. Meili, S. H. Olmstead, and S. Teleki. "Are electronic health records ready for genomic medicine?." Genetics in Medicine 11.7. 2009. 510-517.

[24] M. H. Ullman-Cullere and J. P. Mathew. "Emerging landscape of genomics in the electronic health record for personalized medicine." Human mutation 32.5. 2011. 512-516.

[25] M. Gymrek, A. L. McGuire, D. Golan, E. Halperin, & Y. Erlich. "Identifying personal genomes by surname inference." Science 339.6117. 2013. 321-3

Effects-Based Air Operations Planning Framework: A Knowledge-Based Simulation Approach

André N. Costa[1,2]
[1]Institute for Advanced Studies
Brazilian Air Force
São José dos Campos, SP, Brazil
anegraoc@c4i.gmu.edu

Paulo C. G. Costa[2]
[2]C4I & Cyber Center
George Mason University
Fairfax, VA, USA
pcosta@gmu.edu

Abstract— **Planning air warfare operations has always been a complex endeavor. However, as technology evolves at an increasingly fast pace, so does the complexity of managing its resources. In modern air operations, planners have to deal with a highly changing environment influenced by enemy air defenses, weather forecasts, among many other factors, demanding much effort to handle the great number of constraints and uncertainties presented by them. As a result, a number of decision-support systems have emerged attempting to facilitate the planning of air warfare operations. These systems usually rely on a wide variety of methodologies, which sometimes present a challenge in themselves when it comes to assessing the feasibility and effectiveness of the produced plans. Computer simulations are a practical way of providing this assessment, usually by running the resulting plans multiple times and checking the results against key criteria. Yet, establishing the right criteria, properly accounting for the "fog of war," and avoiding impractical simulation run times and costs are still major challenges. This paper addresses such challenges by proposing the development of a decision-support framework that combines ontology-based agile knowledge and a simulation-based mission planning methodology that accounts for the inherent uncertainties that air operations face. We avoid costly computation times required by simulation-intensive course-of-action analyzers by initially pruning the solution space through ontological reasoning. Moreover, the approach complies with the Effects-Based Approach to Operations, having a clear correspondence of processes with it. The explanations are focused on a specific scenario concerning intelligence, surveillance, and reconnaissance operations.**

Keywords—ontologies; effects-based planning; modeling and simulation; semantic matchmaking

I. INTRODUCTION

The fast pace by which complexity of current military operations is increasing has become a major challenge for mission planners, requiring a much more meticulous planning process to handle all the factors that might influence the outcomes of an operation. Several planning methodologies have been observed in the last years, yielding a number of systems to support air operations planning. To deal with complexity, these systems usually rely on heavy computing power, as well as on specialized operators that must be highly trained in the methodology associated with the system. Such requirements make the planning process brittle, as both the hardware and the operators become scarce resources that usually are centralized and not easily accessible by those who conduct the operations.

To put from a different perspective, this centralization of the planning resources results in distancing the plan development from those who will execute it, since the required planning resources are hardly available on the operations commands. It also impacts the agility of the process, especially when considering highly dynamic mission planning contexts that usually require re-planning to address emerging situations.

The framework presented in this paper leverages semantic technologies to formalize the knowledge required for planning air warfare operations. The reasoning behind this approach is two-fold, (1) to avoid the need for highly trained system planners, and (2) to decentralize the plan building and evaluation process, thus reducing the dependence on heavy computing power.

The planning knowledge to be captured is based on the Effects-Based Approach to Operations (EBAO). This paradigm has been extensively sought by several research and development efforts in the last decades, paving the way to its fruition and further application on the field [1]. According to [2], "EBAO informs every aspect of how the Air Force designs, plans, executes, assesses, and adapts operations." Therefore, it should guide any framework that proposes to aid the planning process of air operations. Following this premise, the backbone of the matchmaking process within this work is the Effects-Based Approach to Planning (EBP), which ultimately defines the semantic description of the domain and how it relates to the planning procedure.

Once the initial states of the EBAO knowledge is made explicit through ontology engineering, the focus of our development becomes to provide a solution that does not require large amounts of computing power and time. Rather, it may be done using portable computers by the operations planners. We achieve that by leveraging the EBAO mission concepts via a logical engine that pre-selects the possibilities given the planning data provided, greatly limiting the solution search space. This way, optimization methods can be used in a much more effective way, applied to scenarios that are simulated in a simplified fashion, and allowing for a quick means of assessing the optimization parameters. In a second step, these generated low-resolution solutions are evaluated based on criteria derived

from the EBAO ontology. The most promising ones then go through a more complex simulation environment that, through entity-level simulation, would provide a much more detailed outcome that includes mission-specific prognostics.

For providing a clearer view of how this framework can be operationalized, an ISR (intelligence, surveillance, and reconnaissance) scenario is built with unclassified data from the Brazilian Air Force. The database includes a number of platforms and sensors that have to be assigned to tasks, leading to actions that generate the desired operational effects. Since not all sensing sources can provide the needed information to task requirements, because the sources are context sensitive [3], this assignment can be very challenging to the planning staff.

ISR operations proved to be a good choice for this initial scenario, since they contain multiple factors that directly affect the planning process, and also make its optimization very important. Also, since ISR assets are oftentimes highly complex and valuable, a less than adequate planning will lead to the loss of costly flight hours and very specialized crews work. Nevertheless, the application on the planning process of other air operations can be made based on the same framework structure, converting the sensor matchmaking phase to a weapon to target matching in the case of airstrike or managing electronic warfare (EW) measures on a suppression of enemy air defense (SEAD) mission, both sharing many complexities with ISR operations.

At this stage of our development, we did not yet reach full circle or obtained conclusive results of a detailed simulation. Thus, our focus on this paper is to provide the long-range vision of the framework, its goals, and an overview of the technical approaches determined by our preliminary research efforts to solve the challenges encountered so far.

The paper is organized as follows. Section II gives a brief overview of previous research on operation planning frameworks like the one proposed, as well as on semantic matchmaking efforts. Section III provides the main concepts involved on EBAO, emphasizing the EBP process. Section IV presents the framework, including the description of the software applications to be used on its implementation. Section V focuses explicitly on the semantic part of the framework. Section VI displays the considered scenario, describing resources and critical conditions that may influence how the image requirements can be met. Finally, Section VII summarizes the paper, pointing to the next steps to be taken.

II. BACKGROUND

A. Planning Simulation Framework

Several planning frameworks are available within the military research context. However, much of the work available is either too complex or remains inaccessible (*e.g.*, classified). The complexity is often directly related to the high resolution required to generate reliable results.

While trying to present an alternative to this complexity problem, some authors have proposed the use of lower resolution simulation and optimization methodologies, which deal with less factors at a time.

Rosenberg *et al.* [4] suggest a collection of decision-support tools for planning generation that consists of "a method to define an operational scenario, an optimization engine to generate a diverse set of solutions, and a suite of visualization and analysis tools to review, analyze, and visualize generated plans." To provide a solution in a timely manner, the authors propose a rapid evaluation of candidate solutions through agent-based modeling and simulation (ABMS). They leveraged the same software used in our proposed framework.

Similarly, [5] – also using the same software application – focuses on a Joint Suppression of Enemy Air Defenses (JSEAD) scenario in which plans are generated, optimized, and simulated. The authors also rely on an ABMS of two sides containing different types of entities, usually targets and air defenses for the opposing side and strikers and JSEAD units for the friendly side. Their results illustrated the potential of low-resolution simulation as a rapid evaluation tool of generated plans, which will be in time described within our own approach.

Unlike in our approach, these research efforts do not apply semantic methods as a form of structuring the modeling and simulation process (*e.g.* [6]), or as a conceptual basis for the framework as those in the subsection below.

B. Semantic Matchmaking Framework

The literature on assigning sensors to missions or tasks is vast, but the use of semantic techniques for this purpose is rather limited. Therefore, it is worth pointing out [7], which advocates for an ontological problem-solving architecture to facilitate automated inference of assigning sensors to missions. This work limits the solution domain as a means of including a coordination system to emulate the assets and complete bears similarities with the aforementioned planning simulation frameworks and the one we propose in this paper.

One of the most productive solutions is sponsored by the U.S. Army Research Laboratory and the United Kingdom Ministry of Defence ([8], [9], [10], [11], [12], [13]). Its authors conceive a system that relies on a series of ontologies for assigning sensors to missions. The backbone of this process is the "Mission and Means Framework" that is claimed to "provide a model for explicitly specifying a military mission and quantitatively evaluating the mission utility of alternative warfighting solutions" [12]. The three basic elements of their methodology are [14]:

- Top-to-bottom solution to the problem of deploying sensors to meet the information needs of tasks in a mission context;
- Combination of reasoning at mission-planning time, and optimization algorithms at mission execution-time; and
- Dynamic deployment configuration of selected sensor instances by means of a sensor infrastructure.

The work includes modular ontologies that cover task requirements, sensor capabilities, and a structured framework to associate tasks with sensors. The ontologies specify the requirements of the missions and the capabilities of the sensors so that the framework is able to decide between combinations of sensors to satisfy the requirements of a given mission [12].

Even though providing a proven assignment system [3], with very well-structured ontologies, this work does not focus on dealing with the uncertainties that a planning scenario presents. This is due to the use of logical reasoners and mostly deterministic functions. Stochasticity is not considered, just comparisons between deterministic possibilities. In addition, the Mission and Means Framework ontology focuses on tasks instead of effects. Thus, although providing a direct and clear way of breaking down missions [9], the approach does not emphasize the holistic view advocated in our work. Finally, [15] also provides more details on how this framework may be structured as an ontology.

III. Effects-Based Planning

Even though utilizing the Mission and Means Framework, [9] states that alternative mission planning approaches, such as effects-based planning, may be structured in a similar way, with the goal of assigning resources to missions.

"Planning to achieve an effect" has been used naively as a rather straightforward definition of EBP. However, the vast majority of planners would argue that any previous approach to military planning would include this asseveration [16]. Therefore, it is imperative to clearly define this concept upfront.

The US Air Force doctrine [2] holds that "there is no single 'effects-based planning' methodology or process. Rather, understanding the principles of an effects-based approach to operations should yield certain insights and enhance comprehension of many general planning concepts". This is the reason why it is important to first understand what EBAO means.

Reference [17] presents the US Joint Forces Command definition of EBAO as "a process for obtaining a desired strategic outcome or effect on the enemy through the synergistic and cumulative application of the full range of military and nonmilitary capabilities at all levels of conflict". Another definition presented on [1] is that "effects-based operations are operations conceived and planned in a systems framework that considers the full range of direct, indirect, and cascading effects, which may—with different degrees of probability—be achieved by the application of military, diplomatic, psychological, and economic instruments".

What both of these definitions emphasize is that the process of planning has to be much more intentional on the pursuit of a holistic view of the operation. There is a focus on addressing not only direct physical effects, but several types of indirect effects, which are influenced by each other. Planners are encouraged to maintain a very broad view of the "big picture", especially during execution, not being caught up in details that can tarnish the end state visualization.

A better understanding of our approach requires exploring EBO's main concepts, which are described in the next Section. However, our framework greatly relies on the EBO principles listed below, which were suggested by [1].

A. Uncertainty

The first principle says that effects-based operations (EBO) planners have to rely on methods that explicitly deal with probabilities and randomness to properly address the inherent uncertainties contained in the air operations. EBP has to fully confront the scope and magnitude of these uncertainties, especially when dealing with outcome predictions.

B. Qualitative modeling

Secondly, in this uncertainty-sensitive framework it is imperative to possess a trustworthy qualitative modeling, including frictional, credibility and cognitive factors that are oftentimes closely related to indirect effects. This is highly dependable on the availability of subject matter experts (SME) to provide information about systems and operations.

C. Agent-based modeling

The qualitative modeling also requires a focus on decision-making, which can be addressed by agent-based modeling approaches, accurately depicting the C4ISR aspects of the operations. Cognitive models may be housed in agent architectures, allowing analyzes of emerging scenarios closer to the reality and with a clear focus on the command and control structure.

D. Capability planning

Is expected from EBP to determine a range of circumstances that provides degrees of confidence towards the meeting of the conditions that characterize a desired end state. These operational circumstances have to be linked to the necessary capabilities to provide this confidence, not only the necessary means.

E. Empirical information

As stated when speaking of the qualitative modeling, empirical information provided by SMEs is extremely important for a successful EBP. In addition to that, information from history, war-gaming, simulations and experiments should be strongly pursued so that the complex models can be modelled and uncertainties reduced.

F. Adaptation

The last principle relates to planning for adaptation. Since a lot of uncertainties are present and the scenarios may present emergent behavior, it is very important to be able to adapt and dynamically change plans even during execution time.

IV. Framework

Before presenting our framework itself, we must first provide the necessary context, which is conveyed in Fig. 1. In EBAO, effects are defined as results of actions. These actions are simply assigned tasks. The ontology described in Section V, is used to support a matching process between effects and resources. The resources in the analyzed scenario are platforms and sensors, which may be mounted to the platforms or not. The objectives that defined the desired effects are then translated to fitness values within the simulation, providing a means for the plans optimization. On the tactical level, these objectives form a specific mission that, on the operational level, leads to the desired end state.

ABMS is used to represent this mission, possessing cognition models that encompass the available expert

information as well as showing the interaction and coordination between the agents, representing the C4ISR processes involved. With several runs of the simulation scenario, uncertainties can be added mostly on the hostile units' locations, on available capabilities, and on different behavior patterns employed. Also, time issues may be initially addressed, since the agents' interactions allow for identifying some of the interferences they generate on each other through the simulation run.

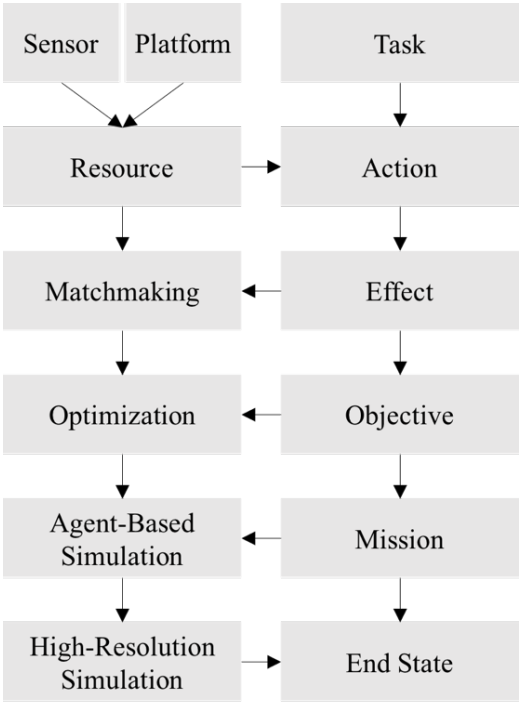


Fig. 1. Air operations planning framework.

Generating cognition models can be a strenuous process, especially when dealing with rule-based scripts for the agent behavior definition, which, besides being hard to implement, do not capture the uncertainties present on military operations. This is why the approach for the friendly and enemy forces' threat assessment process relies on probabilistic models, such as Bayesian networks, capable of representing the dependencies between the entities' actions and the evidence accrued from the C4ISR sources available.

From the ABMS, each of the generated plans can be properly simulated and consequently evaluated based on criteria originated from the three superior goals defined by [18]: flight safety, combat survival, and mission accomplishment. The first goal relates to the need of the pilot to concentrate on flying the aircraft in a safe way, for instance assuring that it has the necessary amount of fuel, that it flies through proper fly zones, avoiding collisions with other aircrafts and the terrain. The second focuses on the chances of enduring through the mission, considering the capabilities of the hostile forces and the exposure to them. Lastly, the third goal illustrates the original objective of the mission performed, such as gathering intelligence information, striking a ground target, or suppressing the enemy's air defenses.

During the optimization parameters definition there is a need of defining a prioritization of these three basic goals. This process depends on several factors, such as rules of engagement, value of the assets, and criticality of the mission. These factors have to be properly valued by the leadership and then parametrized by the analysts to correctly represent the commander's intent (CI) on a top-to-bottom fashion.

At the end, the framework consists of a deeper and more thorough entity-level simulation with the goal of determining if the conditions that define the end state are met within a feasible timeframe by the previously selected best plan. Also, this phase allows for mission rehearsal and order generation.

To summarize, as extracted from [2] and [19], the right-hand side terms of Fig. 1 can be individually defined as:

- **Resources**: all the available assets to generate the desired effects;
- **Tasks**: an action or actions that have been assigned to someone to be performed;
- **Actions**: result of assigned tasks;
- **Effects**: all the physical, functional or psychological outcomes, events or consequences that results from specific military or nonmilitary actions;
- **Objectives**: the clearly defined, decisive, and attainable goals towards which every operation is directed; and
- **End state**: the set of required conditions that defines achievement of the commander's objectives.

As one can notice, the elements presented in the previous Section are met, since the resources are approached as capabilities and contain several qualitative and empirical information, which also permeates the other concepts of the framework. Uncertainty is handled through simulation layers, with the ABMS suggestion alongside. Lastly, the design for adaptation is taken in consideration through the process of generation of multiple plans, and mostly by the ontological reasoning that can quickly change the initial constraints, leading to a faster plan evaluation during dynamic re-planning.

Each of the last four boxes on the left-hand side of Fig. 1 is performed by a different software application that are respectively described as follows:

A. Semantic Modeling: Protégé

Protégé is one of the most popular knowledge-modelling environments. It not only allows users to interactively edit knowledge-bases within its graphic user interface, but also presents a series of plugins that add a number of functionalities and services, such as ontology management tools, multimedia support, querying and reasoning engines, and problem solving methods. Also, it has experienced several actualizations in the last decades and has a vast user community, featuring high stability and usability ratings. As well as the two following applications, it is written in Java, allowing for a smoother integration in the future ([20], [21], [22]).

B. Optimization: ECJ

ECJ is a general-purpose evolutionary computation and genetic programming framework designed for large, heavy-weight experimental needs. It is a free open-source application developed by the Department of Computer Science of George Mason University. In spite of being more than 10 years old, it shows great stability and an optimized design, attested by a large number of users in the genetic programming community.

Besides its main goal of attempting to permit as many valid combinations as possible of individual representation and breeding method, fitness and selection procedure, evolutionary algorithm, and parallelism, it contains multi-objective optimization algorithms, island models, master/slave evaluation facilities, coevolution, steady-state and evolution strategies methods, parsimony pressure techniques, and various individual representations ([23], [24], [25]).

C. Agent-Based Simulation: MASON

MASON is a single-process discrete-event multi-agent simulation toolkit written in Java that comprises a fast core engine and a fully separated visualization display. It is very versatile and easily expandable, providing friendly licensing options and excellent performance. In addition, it is designed to support large numbers of agents relatively efficiently on a single machine in models that are entirely encapsulated. Even the elements of the system itself are highly independent, providing a modular and consistent way to combine its different parts in various ways. Some of these parts form a large set of utilities that has the goal of supporting model design. Finally, as well as ECJ, it is developed and maintained by a research group from George Mason University ([26], [27], [28], [29]).

D. High-Resolution Simulation: VR-Forces

VR-Forces is a simulation environment created by VT MÄK for scenario generation [30]. The platform is widely used throughout the industry, and provides a well-engineered basis for integrating CGFs with urban, battlefield, maritime, and airspace activity. Apart from the graphical interface (front-end), VR-Forces consists of a back-end application, which is its actual simulation engine. As such, VR-Forces scenarios can be scaled up by running multiple front-ends and/or back-ends, communicating through its networking toolkit. Moreover, both the VR-Forces front-end and back-end can be extended either by being embedded into another application or through plug-ins, using the C++ API provided.

Reference [31] provides a study comparing several CGF simulation software in terms of autonomy, learning and adaptation, organization, realism, and architecture. VR-Forces was considered to be the most suitable as a development platform, mostly because its AI capability built-in, very good documentation and technical support, and support for data logger export. The same conclusion was drawn by [32] in a much more thorough analysis.

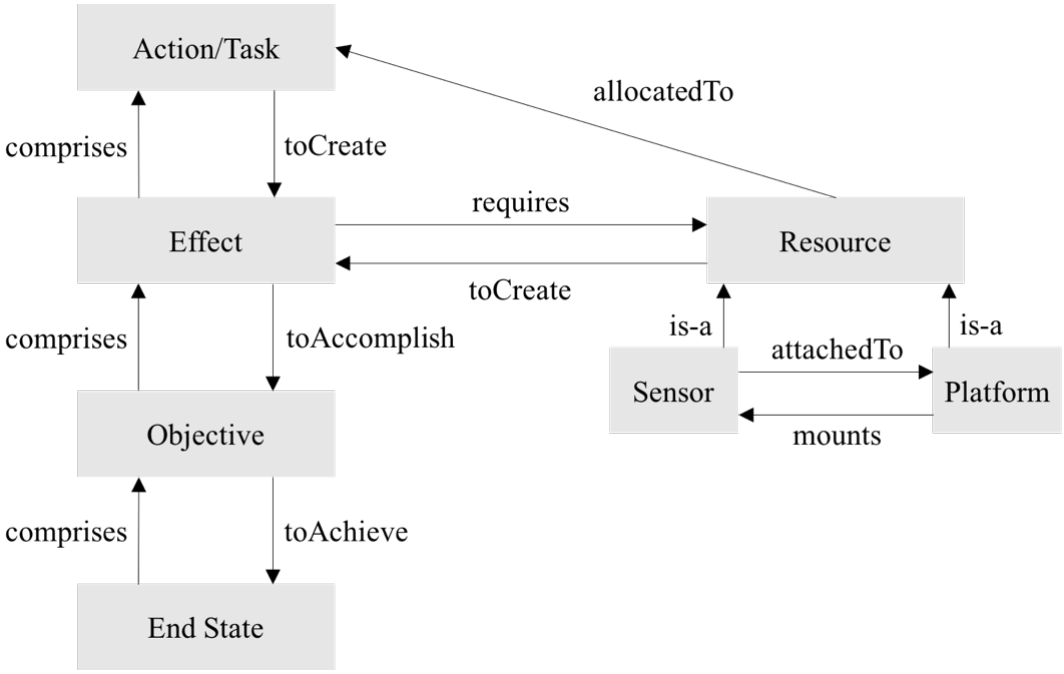


Fig. 2. EBP concepts and relationships.

V. EBO Ontology

A. Knowledge-base

One of the main reasons why this work advocates for the use of ontologies for EBP planning is that they can contain detailed information about the military domain in a very structured way. This is made very formally, explicitly expressing clear and precise definitions of concepts and relationships [33]. Besides, it provides a domain conceptualization of EBAO as expressed on Fig. 2, allowing for a better understanding and application of its features.

Some of the information contained on our knowledge-base will be presented on the next section, but Fig. 3 shows a screenshot taken from the proposed ontology on Protégé. It is structure from three basic domain concepts: capabilities, sensors and platforms. The two latter are the resource descriptions, with their properties and limitations taken into account. The first, is related to the actions that can generate the desired effects, but also to some of the constraints that can influence the mission, as discussed in the next section.

After properly modeling the domain with the most significant parameters and properties, the semantic component of the framework needs to perform the matchmaking between resources and the required tasks for effect generation. This calls for a semantic breakdown of the effects, so that the available capabilities may be used as generation factors for them. After that, the matchmaking methods are able to assign the proper resources as follows.

B. *Semantic matchmaking*

The notion of matchmaking consists of a procedure to find correspondences between entities in ontologies [34]. Whereas process is made by several existing techniques, this work will focus on a description logic approach as advocated by [35].

First, it is important to define that matchmaking takes place as a process in which a requester party triggers the mechanism of finding resources relevant to the request., while the provider party describes the available resources in advance. With that, the matchmaking is made through automated analysis and comparisons of the semantic descriptions of the involved resources.

For doing so, the entities of an ontology and their relationships have to be carefully modelled, representing the air operations domain as a set of concrete resources that vary on several properties. This variance is intended to allow the specification of the resources, having different parameters. However, due to incomplete information, these specifications not necessarily describe all the parameters completely. To deal with that, the notions of entailment and satisfiability back the testing if all request formulas hold in all models of a knowledge base and if these formulas are logical consequences of it.

There are several matching infrences that are able to account for this variance and that can be directly realised by description logic, ranked in the following way according to their degrees of matching [36]:

1) No match: empty intersection between two descriptions;

2) Intersection: non-empty intersection between two descriptions;

3) Non-Disjointness: non-empty intersection between two descriptions in every possible world;

4) Specialisation: subsumption between two descriptions holds from right to left;

5) Generalisation: subsumption between two descriptions holds from left to right; and

6) Equivalence: subsumption between two descriptions holds in both directions.

As stated in section VII, our next step is to test this implementations to verify if the limitations of classical description logic matchmaking are significant for in this context, generating undesired matching behaviors. If so, other methodologies may be embraced, such as the use of nonmonotonic formalisms, such as terminological defaults, autoepistemic and circumscriptive description logic [35].

Fig. 3. Air operations planning framework.

VI. Scenario

The proposed scenario represents the definition of a desired effect yielding intelligence requirements. These requirements are influenced by several factors including the resource availability, environment conditions, and hostile activity. Each factor imposes restrictions on the matching process. The availability is directly related to the instantiation, the environment produces constraints for some sensors and platforms, and the opposing forces impact on the survivability probabilities as well as on the mission success measurements.

To illustrate the EBP focus, the chosen scenario contains the requirement of an effect of assessing, gaining, and maintaining air superiority in support of land and maritime schemes of maneuver, as proposed by [37]. The author already exemplifies how this effect yields ISR actions, such as: detect, discover, and degrade key components of defense systems; confirm damage to target acquisition radars and height-finding radars.

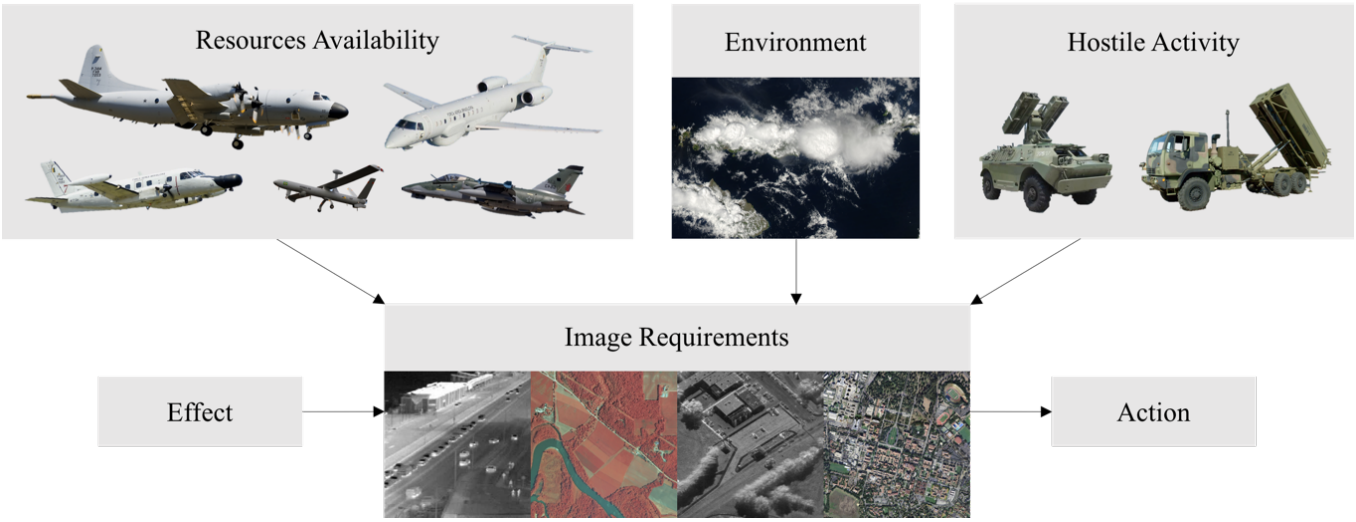


Fig. 4. Scenario factors of influence.

With these actions in hand, the system needs to take into consideration the available sensors and platforms to determine which ones are capable of properly performing them. However, their characteristics have also to be confronted with the instance availability, environment conditions and the hostile threats expected, since this will directly influence the matchmaking process, as depicted in Fig. 4.

A. *Sensors*

Aircraft mounted sensors present different characteristics that may also require different flight altitudes and visibilities in order to properly work. Besides, depending on the applications the image demands may be different, providing alternative information from various sensors. The available sensors considered in this work are:

- OPT: Optical sensors;
- FLIR: Forward-looking infrared cameras;
- MSS: Multispectral Sensors;
- SAR: Synthetic Aperture Radar;
- NCES: Non-communications exploitation systems; and
- CES: Communication exploitations systems.

B. *Platforms*

As showed in Fig. 4 the considered platforms are some of the aircrafts used by the Brazilian Air Force as ISR assets. These platforms mount the aforementioned sensors according to TABLE I. Besides, each one presents different values for range and average speed, which may considerably influence the operations. The aircrafts are:

- Elbit Systems Hermes 450 (RQ-450): medium size unmanned aerial vehicle (UAV);
- Lockheed P-3 Orion (P-3AM): four-engine turboprop maritime surveillance aircraft;
- Embraer EMB-111 Bandeirante Patrulha (P-95BM): twin-turboprop maritime patrol aircraft;
- Embraer EMB-145 RS (R-99): twinjet remote sensing aircraft; and
- AMX International AMX-R (RA-1): ground-attack aircraft for reconnaissance.

TABLE I. SENSORS ATTACHMENTS TO PLATFORMS

Platform	Sensors
RQ-450	FLIR, SAR
P-3AM	FLIR, NCES, SAR
P-95BM	NCES, SAR
R-99	CES, FLIR, MSS, NCES, OPT, SAR,
RA-1	FLIR, OPT

VII. CONCLUSION AND FUTURE WORK

The main goal of this paper was to provide an analysis of the problem and a preliminary structure of the framework advocated to solve it. The work focused on establishing a theoretical basis for delineating this solution, adapting it to effects-based approach to operations concepts. An added constraint was to utilize free and open source applications to form the framework, at least on its initial phases (the only exception being VR-Forces, because of the lack of open alternatives that would provide similar simulation capabilities). Moreover, these applications should be light enough to allow for the execution of the framework on a single machine, what they arguably are.

The development of the framework not only justifies itself as being an explicit representation of EBAO, but also on the combination of simulation methods with an initial semantic matchmaking process that reduces the solution space, allowing for a potentially more agile way of determining operational plans. Additionally, the ABMS phase allows for numerous and fast simulation runs, acting as a fitness evaluation tool for the optimization process as well as an analyzer of emerging behaviors and complex C4ISR interactions.

At the time of this writing, only the initial implementations of the ontology described have been performed. Next steps

include the full development and implementation of the matchmaking process. This step is needed so the optimization can be executed giving continuity to the proposed methodology.

Finally, more information regarding the scenario has to be gathered, also allowing for an expansion of its scope, including gradually more Air Force related activities, for instance airstrike.

ACKNOWLEDGMENT

The authors thank Major Breno Leite, from the Brazilian Air Force, for providing data for the scenario construction as well as serving as a subject matter expert, contributing with empirical information and insights for the modeling. Our gratitude is also extended to Danny Williams, from VT MÄK, who was instrumental in providing technical expertise in the VR-Forces suite.

REFERENCES

[1] P. K. Davis, *Effects-based Operations: A Grand Challenge for the Analytical Community*. RAND, 2001.

[2] Joint Staff, "ANNEX 3-0 Operations & Planning," Washigton, DC, Aug. 2011.

[3] R. Ganger, G. de Mel, T. Pham, R. Rudnicki, and Y. Schreiber, "Sensor assignment to mission in AI-TECD," 2016, p. 98310E.

[4] B. Rosenberg, M. Richards, J. T. Langton, S. Tenenbaum, and D. W. Stouch, "Applications of multi-objective evolutionary algorithms to air operations mission planning," 2008, p. 1879.

[5] J. P. Ridder and J. C. HandUber, "Mission Planning for Joint Suppression of Enemy Air Defenses Using a Genetic Algorithm," in *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation*, New York, NY, USA, 2005, pp. 1929–1936.

[6] A. Tolk, Ed., *Ontology, Epistemology, and Teleology for Modeling and Simulation*, vol. 44. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.

[7] J. Qualls and D. J. Russomanno, "Ontological problem-solving framework for assigning sensor systems and algorithms to high-level missions," *Sensors*, vol. 11, no. 9, pp. 8370–8394, 2011.

[8] A. Preece, T. Norman, G. de Mel, D. Pizzocaro, M. Sensoy, and T. Pham, "Agilely Assigning Sensing Assets to Mission Tasks in a Coalition Context," *IEEE Intell. Syst.*, vol. 28, no. 1, pp. 57–63, Jan. 2013.

[9] A. Preece *et al.*, "Matching sensors to missions using a knowledge-based approach," 2008, p. 698109.

[10] A. Preece *et al.*, "Reasoning and resource allocation for sensor-mission assignment in a coalition context," in *MILCOM 2008 - 2008 IEEE Military Communications Conference*, 2008, pp. 1–7.

[11] G. de Mel, M. Sensoy, W. Vasconcelos, and A. D. Preece, "Flexible Resource Assignment in Sensor Networks : A Hybrid Reasoning Approach," presented at the 1st International Workshop on the Semantic Sensor Web (SemSensWeb), Heraklion, Crete, Greece, 2009, pp. 1–15.

[12] M. Gomez, A. Preece, and G. de Mel, "Towards Semantic Matchmaking in Sensor-Mission Assignment: Analysis of the Missions and Means Framework," International Technology Alliance, Technical Report 1.3, Mar. 2007.

[13] A. Preece, D. Pizzocaro, K. Borowiecki, G. de Mel, M. Gomez, and W. Vasconcelos, "Sensor Assignment to Missions in a Coalition Context: The SAM Tool," presented at the 28th Conference on Computer Communications, Rio de Janeiro, Brazil, 2009.

[14] M. Gomez *et al.*, "An Ontology-Centric Approach to Sensor-Mission Assignment," in *Knowledge Engineering: Practice and Patterns*, A. Gangemi and J. Euzenat, Eds. Springer Berlin Heidelberg, 2008, pp. 347–363.

[15] P. H. Deitz, B. E. Bray, and J. R. Michaelis, "The missions and means framework as an ontology," 2016, vol. 9831, pp. 983109–983109–12.

[16] B. Bullen, "A Bayesian Methodology for Effects Based Planning," Dec. 2006.

[17] I. Duyvesteyn and J. Angstrom, *Modern War and the Utility of Force: Challenges, Methods and Strategy*. Routledge, 2010.

[18] A. Schulte, "Cognitive Automation for Tactical Mission Management: Concept and Prototype Evaluation in Flight Simulator Trials," *Cogn. Technol. Work*, vol. 4, no. 3, pp. 146–159.

[19] Joint Staff, "Joint Publication 5-0: Joint Operation Planning," Washigton, DC, Aug. 2011.

[20] "Protégé." [Online]. Available: http://protege.stanford.edu/. [Accessed: 14-Aug-2016].

[21] J. H. Gennari *et al.*, "The evolution of Protégé: an environment for knowledge-based systems development," *Int. J. Hum.-Comput. Stud.*, vol. 58, no. 1, pp. 89–123, Jan. 2003.

[22] R. Sivakumar and P. V. Arivoli, "Ontology Visualization Protégé Tools - A Review," *Int. J. Adv. Inf. Technol. IJAIT*, vol. 1, no. 4, Aug. 2011.

[23] "ECJ." [Online]. Available: http://cs.gmu.edu/~eclab/projects/ecj/. [Accessed: 14-Aug-2016].

[24] D. R. White, "Software review: the ECJ toolkit," *Genet. Program. Evolvable Mach.*, vol. 13, no. 1, pp. 65–67, Aug. 2011.

[25] S. Luke, "The ECJ Owner's Manual: A User Manual for the ECJ Evolutionary Computation Library," George Mason University, Version 23, Jun. 2015.

[26] "MASON Multiagent Simulation Toolkit." [Online]. Available: http://cs.gmu.edu/~eclab/projects/mason/. [Accessed: 14-Aug-2016].

[27] S. Luke, C. Cioffi-revilla, L. Panait, K. Sullivan, and G. Balan, *MASON: A Multi-Agent Simulation Environment*. .

[28] S. Luke, C. Cioffi-revilla, L. Panait, and K. Sullivan, "MASON: A new multi-agent simulation toolkit," in *University of Michigan*, 2004.

[29] S. Luke, "Multiagent Simulation And the MASON Library," George Mason University, Version 19, Jun. 2015.

[30] "VR-Forces: Computer Generated Forces - VT MÄK." [Online]. Available: http://www.mak.com/products/simulate/vr-forces. [Accessed: 14-Aug-2016].

[31] N. Abdellaoui, A. Taylor, and G. Parkinson, "Comparative Analysis of Computer Generated Forces' Artificial Intelligence," Oct. 2009.

[32] G. Parkinson, "AI in CGFs Comparative Analysis," Defence R&D, Ottawa, Canada, Summary Report, Dec. 2009.

[33] M. Hadzic, P. Wongthongtham, T. Dillon, and E. Chang, *Ontology-Based Multi-Agent Systems*, vol. 219. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.

[34] J. Euzenat and P. Shvaiko, *Ontology Matching*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2007.

[35] S. Grimm, *Studies on the Semantic Web : Semantic Matchmaking with Nonmonotonic Description Logics*. Burke, NL: IOS Press, 2009.

[36] R. Studer, S. Grimm, and A. Abecker, *Semantic Web Services: Concepts, Technologies, and Applications*. Springer Science & Business Media, 2007.

[37] D. R. Johnson, "Enabling Intelligence, Surveillance, and Reconnaissance Effects for Effects-Based Operations Conditions (Maxwell Paper, Number 34)," Jun. 2005.

Extended Abstracts

A Holistic Approach to Evaluate Cyber Threat

Márcio Monteiro[1], Thalysson Sarmento[1], Alexandre Barreto[1] and Paulo Costa[2]
[1]Instituto de Controle do Espaço Aéreo, São José dos Campos, Brazil
[2]C4I Center, George Mason University, Fairfax, USA
E-mails: {contemmcm, thalyssontfs, barretoabb}@icea.gov.br, pcosta@c4i.gmu.edu

***Abstract*—Several vulnerability databases and standards are currently available for assessing the degree of security of IT infrastructures in general. These standards focus on different aspects of the systems, while generally failing to provide support for holistic analyses - a key aspect in ensuring a secure IT infrastructure. This work aims to address this gap by presenting a new methodology for evaluating the overall security risks of a networked system that adopts an ontology-based approach we presented in previous work. We leverage current security standards and databases, while also considering the human factors to build a broader and interconnected view. Our methodology is meant to achieve a more realistic picture of the network security, hence improving situation awareness for its administrators. To illustrate our approach, this paper brings a case study applying the new methodology to a few target networks. The proof of concept is meant to underscore the methodology's effectiveness in assessing the security of the whole network.**

I. INTRODUCTION

Cyber security assessment has a importance role in a modern society. has become more interconnected through computer systems and networks. It is well-established that cyber threats can cause on corporations severe economic losses and damages to their reputation [1]. As a result, investments on cyber security has been growing significantly, even during market crises [2].

A basic standard for cyber security assessment is the Common Vulnerabilities and Exposures (CVE), which is the *de facto* standard to report and communicate software vulnerabilities between organizations and entities. Currently, the CVE has been standardized by the Telecommunication Standardization Sector of the International Telecommunication Union (ITU-T) [3] and is being heavily used by automatic security assessment tools (*e.g.*, Nessus and OpenVAS) to identify software vulnerabilities on target hosts.

On top of CVE, another standard was established to score the vulnerabilities with respect to their severity, impact and exploitation capacity. This standard is called Common Vulnerability Scoring System (CVSS). One of the most important CVSS databases is hosted and managed by the National Vulnerability Database (NVD), which provides the scores for most known vulnerabilities.

Although those standards are very efficient in cataloging and prioritizing software vulnerabilities, system administrators are usually interested in knowing how vulnerable is their entire network, no only individual hosts.

For instance, if a web server is highly protected against external threats, but vulnerable hosts in the same local area network have open access to the server, this condition should impact the overall score of the system. In addition, users can also be considered vulnerabilities of the system, as they could be deceived (or "exploited") somehow to execute malicious software. Then, security unaware or careless users should also impact the overall score of the system.

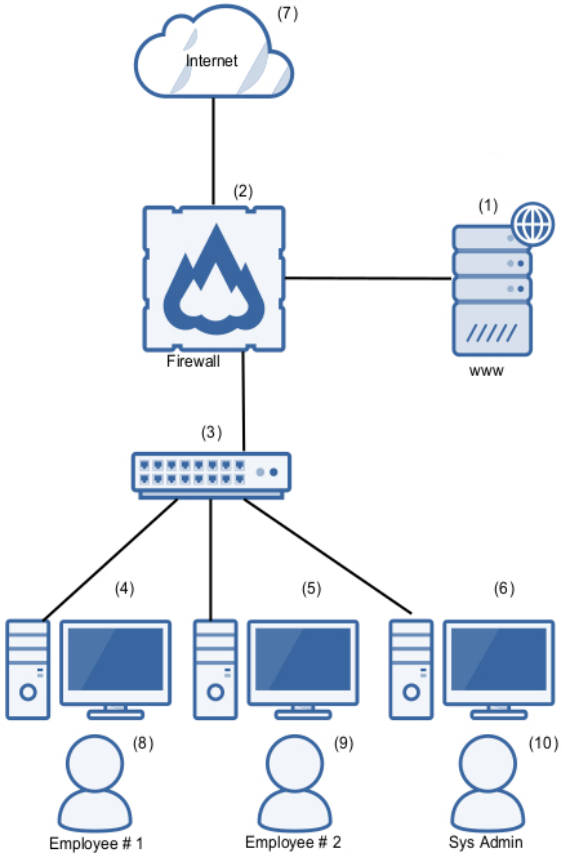


Fig. 1: How secure is this network?

In this work we propose to analyze those aspects (CVE, CVSS and human factors) in a unified manner for a target network, where vulnerabilities scores are propagated through the network's trusted relationships (intentional or not). This way, we provide an overall security metric that can be used to classify entire networks.

This work is organized as follows: Sec. II briefly details the main attributes of CVE and CVSS; Sec. III presents the proposed metric; and Sec. IV concludes with final remarks.

II. Overview

A. Common Vulnerabilities and Exposures

The Common Vulnerabilities and Exposures (CVE) is a standard for cataloging vulnerabilities of computer systems. It consists of a list of information of security vulnerabilities and exposures, mainly reported by the community, aiming to provide common names for publicly known problems. It allows to share data about vulnerability capabilities (tools, repositories, and services).

The main attributes of a CVE are:

- CVE identifier number (i.e., CVE-1999-0067);
- Vulnerability type: buffer overflow, cross site request forgery (CSRF), cross site scripting (XSS), directory traversal, incorrect access control, insecure permissions, integer overflow, missing SSL certificate validation, SQL injection, XML external entity (XXE), and others or unknown;
- Vendor of the product(s);
- List of vulnerable products and versions;
- Attack type: context-dependent, local, physical, remote, other;
- Impact: code execution, denial of service, escalation of privileges, information disclosure, other.

Currently, the MITRE Corporation is responsible for managing CVE identifiers generation and publication through its web site [4]. In addition, MITRE also delegates this attribution to its several CVE numbering authorities (CNAs).

B. Common Vulnerability Scoring System

The Common Vulnerability Scoring System (CVSS) is an open framework for describing specific characteristics of software vulnerabilities. It consists of three metric groups: base, temporal, and environmental.

The *base* group represents the intrinsic qualities of a vulnerability, the *temporal* group reflects the characteristics of a vulnerability that changes over time, and the *environmental* group represents characteristics of a vulnerability that are unique to the user's environment.

In this work, we focus on the *base* metric, which produces a score ranging from 0.0 to 10.0. It is composed by the impact subscore (ranging from 0 to 6) and the exploitability subscore (ranging from 0 to 4). However, the overall CVSS score of a single vulnerability is also impacted by the *temporal* and *environmental* metrics. Readers are encouraged to refer to [5] for more information on CVSS specifications and formulas.

The main attributes of CVSS base score are:

- Attack vector (AV): network (N), adjacent network (A), local (L), and physical (P);
- Attack complexity (AC): low (L), high (H);
- Privileges required (PR): none (N), low (L), high (H);
- User interaction (UI): none (N), required (R);
- Scope (S): unchanged (U), changed (C);
- Confidentiality impact (C): none (N), Low (L), high (H);
- Integrity impact (I): none (N), Low (L), high (H);
- Availability impact (A): none (N), Low (L), high (H);

Usually, the CVSS is represented as a vector string, a compressed textual representation of the values used to derive the score. String (1) below is an example of a CVSS vector string.

CVSS:3.0/AV:N/AC:L/PR:L/UI:R/S:C/C:L/I:L/A:H (1)

The equations adopted to calculate the CVSS base score are provided in Sec. III.

C. Human Factors

Human factors play an important role in the security of an organization, since users are used as both targets and vectors of attacks. Several social engineering methods can be employed to obtain key information and select the most vulnerable employees.

In this work we propose to model the users' "vulnerabilities" as a CVSS-like metric. In other words, the users would also be rated by the impact and exploitability subscores. As an example, users with high privileges in the network would have a high impact factor, because if they get "compromised" that would grant intruders deeper access to the network.

On the other hand, users unaware of security issues or careless about it can be considered highly "exploitable", that is, they can be easily deceived to execute malicious software on their computers. There are numerous methods to do so, such as telephone calls from fake IT staff, phishing campaigns, malicious websites, etc.

To prevent such situations, the staff should perform security awareness training. Besides, the corporation should have a solid information security policy and all means should be employed to enforce it.

III. The Proposed Metric

System administrators usually focus heavily in protecting their networks against external cyber attacks. For this reason, the insider threats might receive insufficient attention and, consequently, the security can be impacted. Considering that every host connected to the Internet is a potential attack vector through phishing campaigns (someone trying to convince the user to execute the malicious code) and applications vulnerabilities (browsers, e-mail and document readers), and that the protection against known hosts is reduced, then a single host can severely compromise the security of the entire network.

The proposed metric in this work is obtained by a five-step approach, each one being required for computing the overall security of a given network. The technique involves building a graph representing the overall network as well as the relationship between each step. The relative importance of each step is assessed using multi-criteria decision analysis concepts.

There are different approaches for building such graph and defining the metric. However, the specific aspects of the cyber security domain involving different perspectives (e.g. technical, human factor, standards, etc.) naturally led us to reuse/adopt the ontology-based approach previously presented in [6]. The general idea is to use semantic techniques in

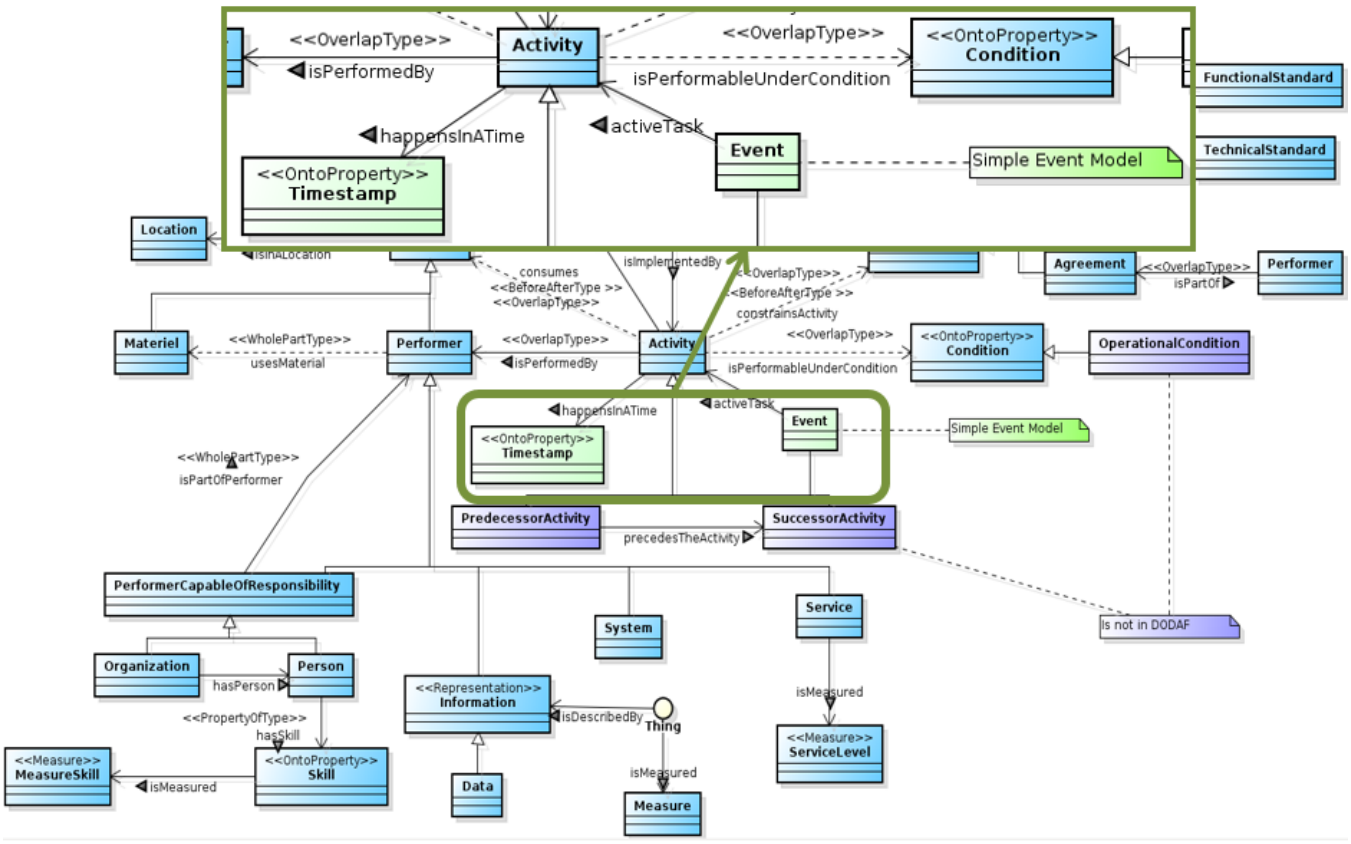


Fig. 2: Mission Ontology

supporting the definition of the target mission, its support task, as well as the services and network configuration required to accomplish a mission. As in the aforementioned previous version, in this work we use the DoDAF Conceptual Data Model to represent the concepts involved in the mission. The difference, however, is that in this work we extend this approach by incorporating time and event descriptions [7].

The ontology is presented in Figure 2, which conveys the queries that can be performed in cyber-situation awareness: **WHAT** (Activity), **WHY** (Goal and Desire Effect), **HOW** (Resource and Guidance), **WHO** (Performer), **WHERE** (Location), and **WHEN** (Timestamp and Event).

1) Complete inventory: The first step consists in obtaining a complete and detailed asset inventory record of the target network, including hubs, switches, routers, software list, etc. This is fundamental for every security approach and should not be a problem for security aware corporations.

2) Communications: The second step consists in mapping the communication between the assets (including the users). If the network contains N assets, this can be mapped into a $N \times N$ matrix.

Taking Fig. 1 as example, we can derive its access matrix as presented in Table I, where the rows represent the asset with communication initiative, the columns represent the communication destination, and a cell filled with a 'Y' informs that such communication is allowed (or that there is nothing forbidding such communication).

TABLE I: Trusted relationships between assets (matrix).

	1	2	3	4	5	6	7	8	9	10
1	-						Y			
2	Y	-	Y	Y	Y	Y	Y			
3		Y	-	Y	Y	Y				
4				-	Y	Y	Y			
5				Y	-	Y	Y			
6	Y	Y	Y	Y	Y	-	Y			
7	Y						-			
8				Y				-		
9					Y				-	
10	Y					Y				-

To generate the aforementioned table, a SPARQL query is performed on the Mission Ontology. This greatly simplifies the otherwise complex task of discovering and mapping connections, in spite of these being hidden or not.

An alternative representation of Table I can be achieved through directed graphs, as depicted in Fig. 3. The main advantage of this approach is that it makes relatively easier to identify nodes with a higher impact higher to the overall security of the network. Also, it becomes possible to derive attack chains throughout the network.

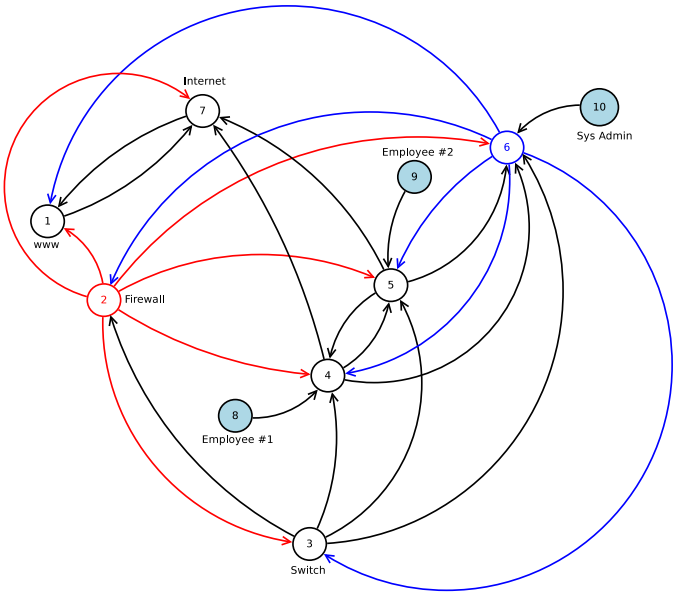


Fig. 3: Trusted relationships between assets (graph).

3) Vulnerabilities assessment: The third step is to obtain the CVE IDs and CVSS base vector string for all N hosts of the network. There are many automated tools that can help in obtaining this information, such as the Nessus Vulnerability Scanner [8] and the Open Vulnerability Assessment System (OpenVAS) [9].

4) Calculating Scores: Once the vulnerabilities are obtained, for every CVSS string we need to compute the impact sub score α and the exploitability sub score β.

The impact sub score α can be computed according to (2):

$$\alpha = \begin{cases} 6.42 \times ISC_{\text{Base}}, & \text{if S} = \text{U}, \\ 7.52 \times [ISC_{\text{Base}} - 0.029] - & \\ 3.25 \times [ISC_{\text{Base}} - 0.02]^{15}, & \text{if S} = \text{C} \end{cases} \tag{2}$$

where

$$ISC_{\text{Base}} = 1 - [(1 - \Delta_{\text{C}}) \times (1 - \Delta_{\text{I}}) \times (1 - \Delta_{\text{A}})]. \tag{3}$$

The confidentiality impact (C), integrity impact (I) and availability impact (A) parameters are given by:

$$\Delta_{\text{C/I/A}} = \begin{cases} 0.56, & \text{if C/I/A} = \text{Low (L)}, \\ 0.22, & \text{if C/I/A} = \text{High (H)}, \\ 0, & \text{if C/I/A} = \text{None (N)}. \end{cases} \tag{4}$$

The exploitability sub score can be computed as:

$$\beta = 8.22 \times \Delta_{\text{AV}} \times \Delta_{\text{AC}} \times \Delta_{\text{PR}} \times \Delta_{\text{UI}}. \tag{5}$$

The attack vector (AV) parameter is given by (6):

$$\Delta_{\text{AV}} = \begin{cases} 0.85, & \text{if AV} = \text{Network (N)}, \\ 0.62, & \text{if AV} = \text{Adjacent Network (A)}, \\ 0.55, & \text{if AV} = \text{Local (L)}, \\ 0.20, & \text{if AV} = \text{Physical (P)}. \end{cases} \tag{6}$$

On the sequence, the attack complexity (AC) parameter is given by (7)

$$\Delta_{\text{AC}} = \begin{cases} 0.77, & \text{if AC} = \text{Low (L)}, \\ 0.44, & \text{if AC} = \text{High (H)}. \end{cases} \tag{7}$$

For unmodified scope (S:U), the following equation applies for the privileges required (PR) parameter:

$$\Delta_{\text{PR}} = \begin{cases} 0.85, & \text{if PR} = \text{None (N)}, \\ 0.62, & \text{if PR} = \text{Low (L)}, \\ 0.27, & \text{if PR} = \text{High (H)}. \end{cases} \tag{8}$$

However, for modified scope (S:C), the following equation applies for PR:

$$\Delta_{\text{PR}} = \begin{cases} 0.85, & \text{if PR} = \text{None (N)}, \\ 0.68, & \text{if PR} = \text{Low (L)}, \\ 0.50, & \text{if PR} = \text{High (H)}. \end{cases} \tag{9}$$

Finally, the user interaction (UI) parameter can be given by (10):

$$\Delta_{\text{UI}} = \begin{cases} 0.85, & \text{if UI} = \text{Not Required (N)}, \\ 0.62, & \text{if UI} = \text{Required (R)}. \end{cases} \tag{10}$$

5) Computing the proposed metric: After computing the impact sub score (α) and exploitability sub score (β), for every vulnerability found in previous steps we need to assemble a P matrix, where the first column ($p_{i,1},\ \forall i$) corresponds to the impact sub score (α), and the second column ($p_{j,2},\ \forall j$) corresponds to the exploitability sub score (β). Then, we need to append three additional points to this matrix such that its final version is according to (11):

$$P = \begin{bmatrix} p_{1,1} & p_{1,2} \\ \vdots & \vdots \\ p_{N,1} & p_{N,2} \\ 0 & 0 \\ \max(p_{1,1}, \ldots, p_{N,1}) & 0 \\ 0 & \max(p_{1,2}, \ldots, p_{N,2}) \end{bmatrix} \tag{11}$$

where the function $\max(\cdot)$ returns the maximum value of its arguments and N denotes the number of vulnerabilities found on previous steps.

Finally, we must compute the convex hull of the matrix P and its 2D area (considering the outmost vulnerabilities as vertices of the polygon), and divide resulting area by the highest possible CVSS subscores ($6 \times 4 = 24$). Conducting the calculations this way ensures that the proposed metric is presented as percentage. The results are then used to rate the network security according the intervals presented on Table II.

Fig. 4 depicts an example of a fictitious network composed of three nodes. The overall vulnerability metrics has been appointed as 70.4476 %, which corresponds to the rating *Highly Vulnerable*, according to Table II. Every marker on this figure

TABLE II: Ratings

Min (%)	Max (%)	Rating
00.00	00.00	None
00.01	39.99	Low
40.00	69.99	Medium
70.00	89.99	High
90.00	100.0	Critical

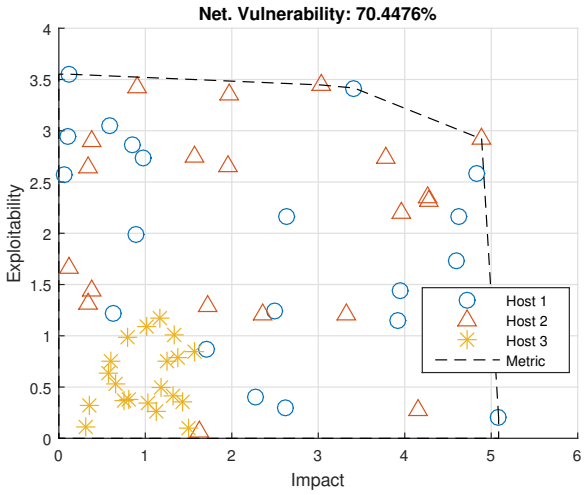


Fig. 4: Vulnerability assessment using the proposed metric for a highly insecure network.

corresponds to a CVSS metrics (impact and exploitability sub scores).

Likewise, Fig. 5 presents a second network with less severe individuals vulnerabilities throughout the nodes of the network. Notice that the overall vulnerability was 16.7402 %, which corresponds to the rating *Low*, according to Table II.

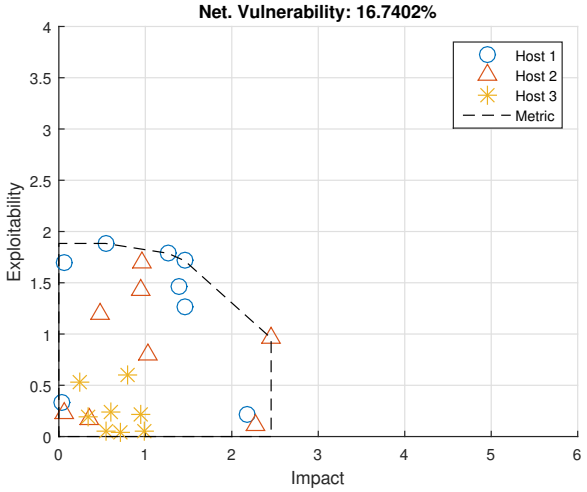


Fig. 5: Vulnerabilities of hosts of the network.

IV. Final Remarks

This work presented an ontology-based approach for analyzing the vulnerability of a network in a holistic way, using multiple-criteria analysis and modeling the human factor as CVSS v3 base scores. An example on a fictitious network was performed in order to demonstrate the practicality of the proposed metric. Further, the reuse of concepts previously defined in an existing ontology we had developed suggests that the approach can be generalized to encompass the diverse aspects that permeate the way different corporations are structured.

ACKNOWLEDGMENT

Márcio Monteiro, Thalysson Sarmento and Alexandre Barreto would like to thank the financial support of the Brazilian agencies MCTI and FINEP (Ref. 04/2013/12).

References

[1] CNN Money, "Cybercrime costs the average U.S. firm $15 million a year," 2015, [accessed 05-Sept-2016]. [Online]. Available: http://money.cnn.com/2015/10/08/technology/cybercrime-cost-business/

[2] Reuters, "Cyber security investing grows, resilient to market turmoil," 2015, [accessed 05-Sept-2016]. [Online]. Available: http://fortune.com/2015/09/23/cyber-security-investing/

[3] Study Group 17, *ITU-T Recommendation X.1520: Common vulnerabilities and exposures*, Std., April 2011.

[4] MITRE, "Common vulnerabilities and exposures – the standard for information security vulnerability names," [accessed 05-Sept-2016]. [Online]. Available: https://cve.mitre.org/

[5] FIRST, "Common vulnerability scoring system v3.0: Specification document – version 1.7," [accessed 05-Sept-2016]. [Online]. Available: https://www.first.org/cvss/specification-document

[6] A. Bareto, "Cyber-argus framework – measuring cyber-impact on the mission," Ph.D. dissertation, Instituto Tecnológico de Aeronáutica, Brazil, 7 2013.

[7] W. R. e. a. VAN HAGE, "Design and use of the simple event model (sem)," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 9, no. 2, Sep 2011.

[8] Tenable Network Security.

[9] "Open vulnerability assessment system (OpenVAS)," [accessed 05-Sept-2016]. [Online]. Available: http://www.openvas.org

A Practical Approach to Data Modeling using CCO

Rod Moten
Datanova Scientific
Baltimore, Maryland

Bill Barnhill
EOIR Technologies
APG, MD

***Abstract*—In this paper, we present work in progress on using the Information Domain ontologies of CCO (Common Core Ontologies) as a domain model for land combat. Our goal is to use the domain model as a common semantics for multiple land combat logical models. In the paper, we show how our domain model can be mapped to different logical models in a manner that is less labor intensive than the approach commonly used by users of CCO. We demonstrate our approach by describing how our domain model, which is a domain ontology of CCO, is mapped to logical models created in Ecore and NIEM (National Information Exchange Model).**

I. Introduction

There are three primary forms of a data model, domain model, logical model, and a physical model [1]. A domain model specifies the concepts that data represents, the properties of the concepts and the relationships between concepts. A logical model species the logical structure of data. A physical model species how data is represented in machine readable format. Ideally, a logical model is derived directly from a domain model or a formal relationship is defined between the domain model and the logical model. In these cases, the domain model serves as the semantics of the logical model. Semantics is assigned to the logical model via a mapping between the domain model and the logical model.

There are multiple approaches of performing this mapping. One approach is to develop a mapping between objects in the domain model and and objects in the logical model. For example, the domain model could be defined using an ontology. The mapping specifies how to convert objects in the logical models to individuals in the ontology.

We used this approach for several projects where the domain models were domain ontologies of CCO (Common Core Ontologies) [2]. CCO is a collection of upper, middle, and domain ontologies in OWL that extend BFO (Basic Formal Ontologies) [3]. Figure 1 contains a diagram of the ontologies in CCO.

One of the authors of this paper has used CCO for creating domain ontologies for a motion imagery analysis application [4] and other projects. In all of these projects, we sought to use ontologies conformant to the CCO as domain models. In addition, we sought to create mappings from the logic models of existing tactical military software systems to the domain models. We required the assistance of an ontologist with in-depth knowledge of CCO to create the mappings. As a result, using CCO may have a higher cost than an approach that allows programmers or data architects to develop the mapping independently. As a result, the government sponsor of the projects considered the use of CCO impractical for tactical military systems.

We believe that CCO is practical for tactical military systems. The problems we encountered were due to how CCO was used. The problems we encountered occurred because of differences in the modeling objectives of a logical model and a domain model defined as a formal ontology. A logical model defines the symbolic structure of entities for automated processing and analysis. The structure is chosen in order to simplify processing and analysis. For example, the essential properties of a person, such as name and birth date, are modeled as attributes of the same object in a logical model. However, the domain ontologies of CCO are specifications of the metaphysical make up of entities. Therefore, essential properties of the same entity may have different structural representations as individuals in the CCO. In other words, the graph patterns of the triples representing the essential attributes of the same entity may be different. For example, a birth date for a person is a temporal interval for a birth event that occurs on a person agent. A name of a person is an information bearer that inheres on a person agent. This means to map a person entity in a logical model requires determining how each attribute is represented metaphysically and then create the triples accordingly.

An approach that requires examining each attribute equates to defining a separate function for converting each attribute to individuals in the domain model. If we measure the cost of creating a mapping based on the number of functions that have to be created, then an approach that used a single function for mapping sets of entities to concepts may be less expensive than an approach that required a function for each attribute.

To develop an approach based on converting sets of entities to concepts, we propose modeling a domain model as information about the metaphysical properties of entities. In other words, consider the domain to be the terms that designate the entities and relationships between the entities. For example, Aircraft and F-14 would be concepts where F-14 is subsumed by Aircraft. In this case, there are multiple Aircraft individuals and multiple F-14 individuals which are also Aircraft individuals. However, in an information model, there is only one designator term for all aircrafts and one designator term for all F-14s. The subsumption relationship between Aircraft and F-14 could be modeled using a descriptive term, such as derives-from. More specifically, the relationship could be modeled as the triple 'F-14 derives-from Aircraft'. This means the domain ontology has to extend the Information Domain ontologies of

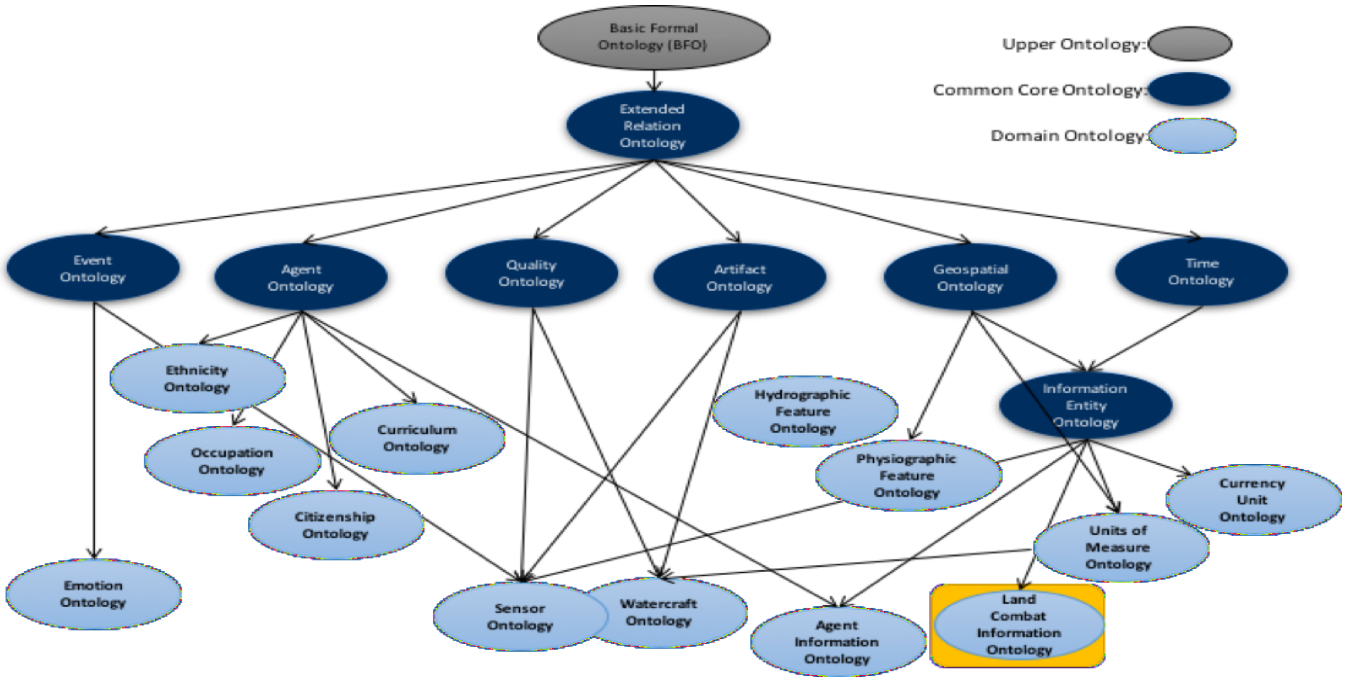


Fig. 1. The ontologies of CCO and the Land Combat Information Ontology.

CCO. However, we have to ensure that the domain ontology isn't just an OWL encoding of a logical model. This approach is used by some techniques for automatically creating schema from ontologies [5].

Using this approach, we do not map objects in the logical model to individuals in the ontology. Instead, we create a mapping where the domain model represents concepts that have direct mapping to syntactic classes in the logical model. This mapping should be more intuitive to data architects since it requires little knowledge of CCO and ontology development.

In this paper, we demonstrate a method for creating domain ontologies in CCO that can be systematically mapped to logical models. In Section II, we provide an overview of the Information Domain ontologies of CCO. Then in Section III we describe how a domain ontology should extend the Information Domain ontologies by creating a proof–of–concept domain ontology for land combat. Then in Section IV we describe how the domain ontology maps to logical models in ECore [6] and NIEM (National Information Exchange Model)[1]. We conclude the paper in Section V with a discussion on why we think our approach faithfully encodes the semantics of the domain and isn't merely a logical model in OWL.

II. Information Ontologies in CCO

The information entity ontology is partitioned into two class hierarchies, *information bearing entities* and i*nformation content entities*. We call information bearing entities *information bearers* for short.

An information bearers is and independent continuant that carries information. For example, a track of an aircraft is an information bearer because it contains information about the flight pattern of an aircraft.

Information content entities are things used to represent information for an information bearer. For example, a 2D graph could be the information content entity of an air track. In this case, the 2D graph is the information that represents the flight pattern of an aircraft. In addition, a 3D graph could be the information content of the air track. The information content entity does not have to be unique to its bearer. For example, -20 degrees Celsius is an information content entity that inheres in many information bearers, such as the current temperature or the lowest operating temperature.

Information content entities are organized into three hierarchies, *directive information*, *designative information*, and *descriptive information*. In this paper, we only use designative and descriptive information entities. Therefore, we omit describing directive information. Designative content entities consist of a set of symbols that denote some entity. Type codes are an example of designative content entities. Descriptive content entities consist of a set of propositions that describe some entity. Numeric scales are examples of descriptive content entities.

There is only one class for Information Bearers, Information Entity Bearers. Our domain ontology for land combat will define a hierarchy for land combat terms with Information Entity Bearer as the root.

III. Land Combat Domain Model

In this section, we give an overview how we created the land combat domain model as an extension of the Information Entity Ontology.

[1]https://www.niem.gov/

Descriptive Name	Acronym/Standard Name
Common Warfighting Symbology	MIL-STD-2525C
Variable Message Format	MIL-STD-6017C
US Message Text Format	MIL-STD-6040 Rev. B
Modernized Intelligence Database	MIDB
Ground-Warfighter Geospatial Data Model	GGDM

TABLE I
LAND COMBAT DOMAIN SOURCES

A. Identify Sources

The first step in creating the domain model is identifying the sources of the information entities. For the land combat proof–of–concept, we use the standards in Table I.

B. Define Class Hierarchy

For the second step, we defined a class hierarchies that extend Information Bearing Entity and Information Content Entity.

Our approach is based on the assumption that the domain model is a conceptualization of information about entities. More specifically, the domain model consists of concepts that can be classified as an *entity report*, an *entity artifact*, or an *entity representation*. An entity report is a concept which captures in a structured machine-readable form one or more observations about an entity's state at a given time, as observed by an agent with a given location (where the agent can be human or software). An entity artifact is a concept which describes assertions about an entity. Entity artifacts are derived either from entity records or from other entity artifacts. For example, a detailed entity artifact about a person can be created from multiple entity records obtained from HUMINT sources. There can be more than one entity artifact asserting information about a given entity or there may be no entity artifacts asserting information about a particular entity. An entity representation is a concept describing human understandable signs and symbols which can be presented to a human actor via some sensory medium (e.g., an audible alert, a PowerPoint deck, a printed document). Figure 2 shows an example of the entity informational categories.

We partition the terms into two groups. We define OWL classes for each of these groups. The first group of terms are terms representing entity artifacts and entity reports. We call these terms *LC (Land Combat) Information Entities*. The second group of terms contain qualities, traits, roles, and characteristics of the entity referenced by an entity artifact or an entity report. The class for this group of terms will be Information Content Entity classes. Figure 3 shows a snapshot of the object properties, LC Information Bearing Entities, and the Information Content Entity classes.

C. Convert Terms to Individuals

In this step, we present the guidelines we used to determine the terms from the source documents we used as individuals in the ontology. We use the noun and adjective phrases in the source documents to create the individuals in the ontology. For example, the terms 'aircraft carrier', 'light', 'guided missile',

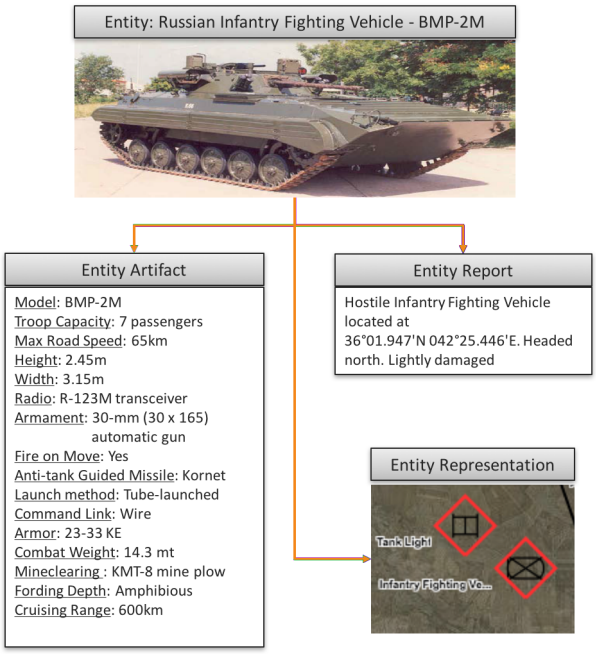


Fig. 2. Example depicting informational entity categories.

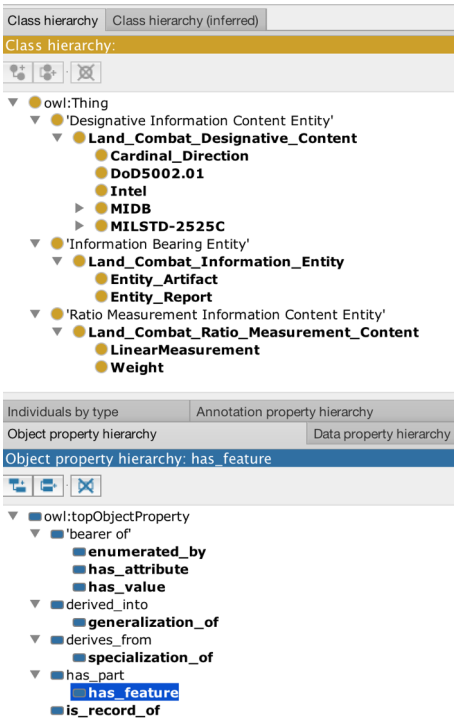


Fig. 3. Screen shot of the Land Combat Domain Model T-Box in Protégé.

and 'nuclear powered' are noun and adjective phrases in USMTF. Each of these terms will be an individual. The adjective phrases will become land combat designative content individuals. The noun phrases will become information content entity individuals and information bearing entity individuals.

The usage of the noun phrase determines which class the term belongs to. If the noun phrase is an entity, such as aircraft carrier, then it will become an LC Information Entity. If the noun phrase is the value of a type code, then it will become an Information Content Entity. More specifically, it will be an individual of a subclass of LC Designative Content. If it is a multi-valued numeric attribute, then it will be an individual of an LC Ratio Measurement Content subclass.

The individuals of the LC Relation class are verb phrases that describe a relationship between terms in the standard. For example, 2525C contains a taxonomy of air tracks about different kinds of aircraft. Therefore, 'is about' is a relation between the LC Information Entities. Notice that the relation individuals may not be verb phrases in the standard. Instead, they are conceptualization of the relationships between terms in the standard.

D. Define Ontological Relationships of the Domain

By defining relationships between terms using an individual, we can support defining an arbitrary number of relations. We can use OWL properties as *meta–relationships* between individuals. More specifically, we define a fixed set of OWL properties for defining subsumption and composition relationships between individuals. These relationships hold for all domains.

Each of the meta–relation properties is a CCO property or a sub-property of a CCO property. Figure 4 depicts pictorially a sample of triples using all of the meta–relation properties. The CCO properties are in blue and the derived properties are in black. The 'derives from' indicates the subject has all of the same properties as the object. Therefore, 'Stragetic Bomber' and 'Tactical Bomber' each have a 'Fixed Wing' as a quality. The 'derives from' property is the only subsumption property in our model. The properties 'has object' and 'has subject' are used to indicate the subject and object of an LC relation. The properties 'has feature', 'has part', 'has value', and 'has code' all indicate a part–whole relationship between the subject and object. The difference between the three is the range of the properties. The range of 'has feature' is Information Content Entities, but the range of 'has part' is an LC Info Entity class. The range of 'has code' is LC Info Type Code. And the range of 'has value' is subclass of LC Ration Measurement Info Term. The property 'enumerated by' indicates the enumerations of a type code. The property 'has quality' indicates the object is a quality of the subject.

IV. Land Combat Logical Models

In this section, we describe how classes and individuals from the domain model created in Section III map to logical models in ECore and NIEM.

A. Mapping to ECore

ECore is a metal model for defining models in EMF (Eclipse Modeling Framework) [6]. Using Ecore, developers can create models similar to UML Class diagrams and automatically generate code from the models. Ecore contains constructs and features common in object-oriented design, such as classes, enumerations, and inheritance.

Mapping to an object model in ECore is straightforward. Each of the individuals of Type Code becomes an Enumeration class in ECore. The enumerations are determined by the 'enumerated-by' property. More specifically, if A 'enumerated by' X and A 'enumerated by' Y are triples, then X and Y are the enumeration literals of enumeration class corresponding to A.

Each LC Info Entity individual will be a class in ECore that extends the root class `InfoEntity`. The derived from property determines its subclasses and parent class. More specifically, if A 'specialization of' B or B 'generalization of' A is a triple, then the ECore class corresponding to A, will be a subclass of the ECore class corresponding to B. The attributes of the classes will be defined as follows. For each triple $S\ p\ O$, where S is a LC Entity Info Individual and p is one of the properties, 'has feature', 'has value', 'has attribute', or 'has part', there will be an attribute in the class corresponding to O whose type is the type corresponding to O. Each of these types will be created as classes using the same approach.

If the ECore class created from the Entity individual A does not have any attributes, then it can be made into an enumerated class. This will require the individual B in a triple A 'specialization of' B or B 'generalization of' A be converted into an enumeration literal.

Each A 'is record of' B triple will be converted into an association class. More specifically, it will be converted into a class that contains two attributes, `subject` and `object`. The type of `subject` will be the type corresponding to A. The type of `object` will be the type corresponding to B.

B. Mapping to NIEM

NIEM is a logical model developed by the U.S. Government to enable state and federal agencies to share data. The purpose of NIEM is to establish a common structured vocabulary for a set of terms used in all domains relevant to government activities, such as person and location, and a set of common terms used in specialized domains relevant to some government activities, such as hospital and unmanned vehicle. NIEM uses XSD and UML to define the terms so that it can be readily used in software.

In NIEM, terms are partitioned into *elements* and *types*. An element represent properties or attributes of objects. A type represents a set of objects that have the same properties and semantics.

Each Entity individual will be a NIEM type. Elements of the NIEM types are determined by the objects in triples. Objects of 'has feature', 'has attribute', and 'has part' will be come composite elements. Objects of 'has value' will be come scalar

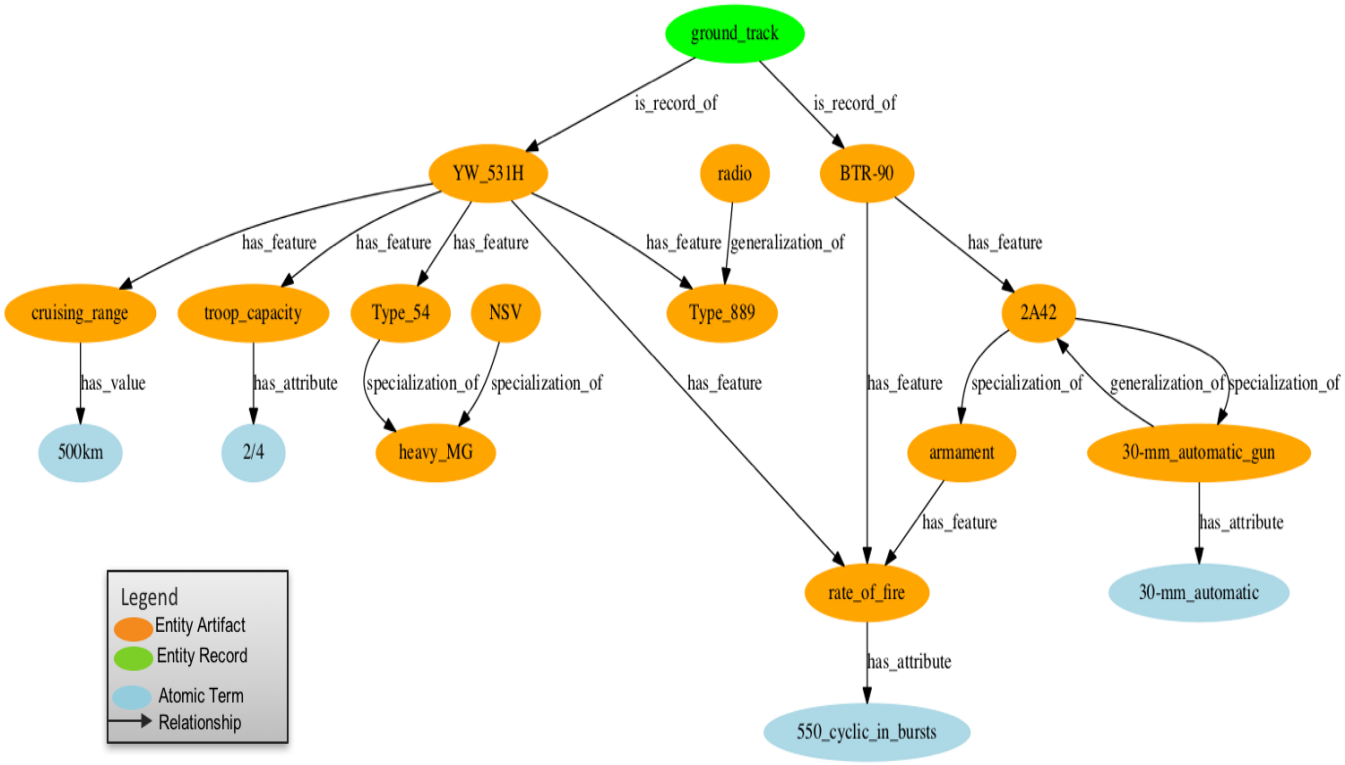


Fig. 4. Example illustrating use of meta–properties

elements. The 'generalization of' and 'specialization of' will determine inheritance.

Code Lists can be created in a similar fashion to how enumerated classes are created in ECore. Association Types can be created from 'is record of' triples.

A logical modeler determines whether an object of 'has attribute' should be considered Metadata. NIEM Augmentation and Extension Augmentation point and extensions are determined from 'derives from'. The logical modeler determines whether to create an augmentation point or an extension.

V. Conclusion

We described an approach to create a domain model in OWL for which logical models can be derived in a systematic way. Our approach is truly a domain model because it uses terminology from domain documents to create the ontology entities. In addition, the domain model contains the ontological relationships from the domain. For instance, it is able to specify that two concepts are related because one concept is a quality of another concept. In addition, it is able to capture role relationships.

We provided an overview of how we intend to use domain models created with our approach to generate logical models in Ecore and NIEM. We believe project managers will consider our approach suitable for their projects because it does not require expertise in ontologies and in-depth knowledge of CCO.

In the future, we plan to build a complete land combat domain model using the sources mentioned in Table I. We hope this domain model will be used as a common semantics for U.S. Army's initiative to use a single computing platform for multiple army battle command systems [7].

References

[1] *Interim report: ANSI/X3/SPARC Study Group on Data Base Management Systems*. Washington, D.C.: ACM, 1975.

[2] J. R. Schoening, D. K. Duff, D. A. Hines, K. M. Riser, T. Pham, G. H. Stolovy, J. Houser, R. Rudnicki, R. Ganger, and A. James, "PED fusion via enterprise ontology," in *Proceedings SPIE 9464,Ground/Air Multisensor Interoperability, Intergration, and Networking for Persistent ISR VI*. International Society for Optics and Photonics, May 2015.

[3] R. Arp, B. Smith, and A. D. Spear, *Building ontologies with basic formal ontology*. Mit Press, 2015.

[4] W. R. Thissell, R. Czajkowski, F. Schrenk, T. Selway, A. J. Ries, S. Patel, P. L. McDermott, R. Moten, R. Rudnicki, G. Seetharaman, I. Ersoy, and K. Palaniappan, "A Scalable Architecture for Operational FMV Exploitation," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2015, pp. 10–18.

[5] M. J. O'Connor and A. Das, "Acquiring OWL Ontologies from XML Documents," in *Proceedings of the Sixth International Conference on Knowledge Capture*, ser. K-CAP '11. New York, NY, USA: ACM, 2011, pp. 17–24.

[6] D. Steinberg, F. Budinsky, E. Merks, and M. Paternostro, *EMF: eclipse modeling framework*. Pearson Education, 2008.

[7] S. Lyngaas, "Four years on, Army common operating environment takes shape" *FCW*, Sep. 2015. [Online]. Available: https://fcw.com/articles/2015/09/22/army-mobile-computing.aspx

Semantic Cyberthreat Modelling

Siri Bromander
mnemonic
Norway
siri@mnemonic.no

Audun Jøsang
University of Oslo
Norway
josang@ifi.uio.no

Martin Eian
mnemonic
Norway
meian@mnemonic.no

***Abstract*—Cybersecurity is a complex and dynamic area where multiple actors act against each other through computer networks largely without any commonly accepted rules of engagement. Well-managed cybersecurity operations need a clear terminology to describe threats, attacks and their origins. In addition, cybersecurity tools and technologies need semantic models to be able to automatically identify threats and to predict and detect attacks. This paper reviews terminology and models of cybersecurity operations, and proposes approaches for semantic modelling of cybersecurity threats and attacks.**

I. Introduction

When security incidents occur there is typically limited understanding of who the threat agent is, why they attack and how they operate, which makes it difficult to make well informed decisions about countermeasures. Threat agents who are not identified and made responsible for their actions will continue their criminal behaviour. When we do not understand the attacker we can only see - if even that- the results of the attacker's actions. Improved cybersecurity requires digital threat intelligence - structured and semi-automated analysis and sharing of information. In order to make sense out of increasingly large and complex datasets related to cybersecurity we see the potential in developing models and tools for automated or semi-automated classification and discovery of cyberthreats based on ontologies.

Semantic technologies and ontologies are a relatively new logic-based landscape of technologies and tools aimed at giving better meaning to large and unstructured corpuses of data. Interesting research challenges are for example to investigate semantic representations of relevant concepts in the domain of cybersecurity big data, in order to facilitate advanced machine learning, search and discovery.

The potential benefit of this approach is that the developed tools and related technologies will provide a flexible framework for representing and structuring the large variety of data with which security analysts are confronted. The framework can further be used for the implementation of cybersecurity analytics tools.

II. Cybersecurity Threat and Risk Models

Cybersecurity is the body of technologies, processes and practices designed to protect networks, computers, programs and data from attack, damage or unauthorized access. Cybersecurity thus assumes that some actors, typically called *threat agents*, have the intent and capacity to produce attacks, gain unauthorized access and cause damage. The magnitude of the perceived potential damage caused by cyber attacks is typically interpreted as security risk.

A. Specific Security Risk Model

Cybersecurity risks are caused by threats. However, the concept of a threat can be ambiguous in the sense that it can mean the threat agent itself, or it can mean the thing that a threat agent (potentially) produces, typically called a *threat scenario*. Figure 1 illustrates a specific risk model which integrates the concepts of threat agent and threat scenario.

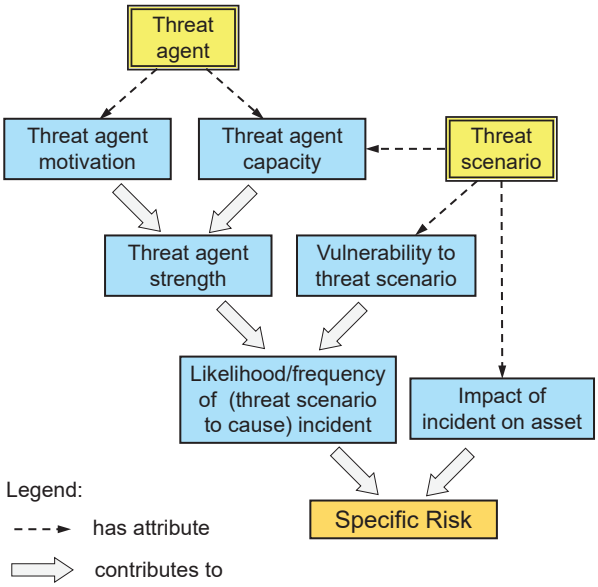


Figure 1. Specific risk model including threat agent and threat scenario

The specific risk model of Figure 1 emphasizes the risk dimension of threats, i.e. how threats lead to risk.

It can be seen that the threat agent and the threat scenario have very different attributes, but in combination they both contribute to risk. A threat agent can be modeled as a real agent with a motivation or goal as well as with a capacity to execute a specific threat scenario. Together, the motivation and capacity produce the strength of the threat agent. The threat agent strength can be modelled according to the weakest

This research was supported by the research projects TOCSA, ACT and Oslo Analytics funded by the Research Council of Norway.

link, i.e. the attacker is only as strong as the weakest of its motivation and capacity.

A threat scenario can be modelled as a sequence of attack steps which can be stopped by defence and security mechanisms. However, when the defence mechanisms fail to stop a specific threat scenario, we say that there are *vulnerabilities*.

The more severe the vulnerabilities and the greater the strength of the threat agent, the greater the likelihood that the threat scenario will cause a security incident and lead to damage, as illustrated in Figure 1. The actual risk of a specific threat scenario emerges by including the amplitude of the expected damage in case the security incident actually occurs. Risk assessment models such as in [1] are based on this interpretation of security risk.

There can of course be many different threat scenarios leading to the same goal when seen from the attacker's perspective. Each scenario represents the dynamic execution of a *tactic*. The attacker might consider multiple tactics, and then decide to use the one which is assumed to produce the greatest expected result with the least effort.

The threat scenario is an abstract set of steps executed in sequence, which from the victim/defender's perspective can cause damage to its assets. A threat scenario becomes a *cyber attack* when the scenario is actually executed. Behind every attack there is thus a specific threat scenario executed by an attacker or a group of attackers. However, a threat scenario by itself is abstract, and does not become an attack unless it is actually executed.

A threat scenario can therefore be interpreted as the blueprint for attacks. For cyber defenders there is thus a fundamental difference between detecting real attacks and identifying threat scenarios which only represent potential attacks.

B. Stillions' Detection Maturity Level Model

A model for the maturity of cyberthreat detection has been proposed by Ryan Stillions in several blogpostings [2]. A slightly extended version of Stillions' Detection Maturity Level (DML) model is illustrated in Figure 2. We have added the additional *DML-9 Attacker Identity* which can be important in certain contexts. We have also added *precision* and *robustness* to illustrate the qualitative aspects of features at each level. The DML model emphasizes the increasing level of abstraction in the detection of cyber attacks, where it is assumed that a security incident response team with low maturity and skills only will be able to detect attacks in terms of low level technical observations in a network, without necessarily understanding the significance of these observations. On the other hand, a security incident response team with high maturity and skills is assumed to be able to interpreted technical observations in networks in the sense that the type of attack, the attack methods used and possibly the identity of the attacker can be determined.

The levels of the DML model are briefly explained below. The focus is on what the IR team (incident response team) is

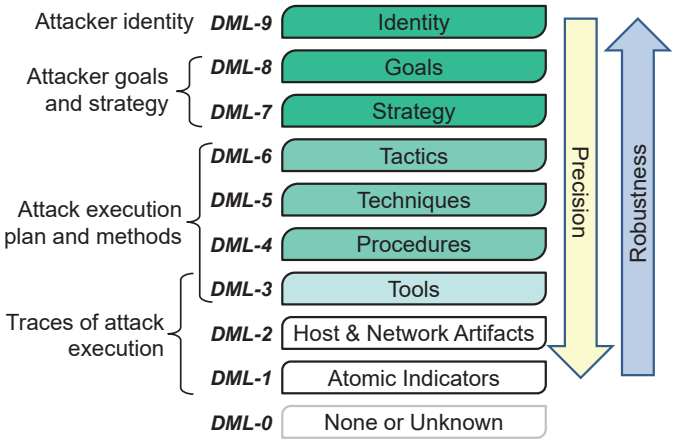


Figure 2. Detection Maturity Level Model [2]

capable of doing at each level. Our description is a summary and interpretation of Stillions' description [2].

- **DML-0 None or Unknown**. There is no IR team, or they are totally clueless.
- **DML-1 Atomic indicators of compromise (IOCs)**. These are elementary pieces of host & network artifacts, which might have been received from other parties. The value of atomic IOCs is limited due to the short 'shelf life' of this type of information.
- **DML-2 Host & Network Artifacts**. This is the type of information which can be collected by network and endpoint sensors. With high capacity links the amount of information collected can be overwhelming and requires good analytical tools to analyse and understand the attack at higher levels of abstraction.
- **DML-3 Tools**. Attackers install and use tools within the victim's network. The tools often change, so that a tool detected and analysed in a previous security incident might be similar but not exactly the same in new attacks. DML-3 means that the defender can reliably detect the attacker's tools, regardless of minor functionality changes to the tool, or differences in the artifacts and atomic indicators left behind by the tool.
- **DML-4 Procedures**. Detecting a procedure means detecting a sequence of two or more of the individual steps employed by the attacker. The goal here is to isolate activities that the attacker appears to perform methodically, two or more times during an incident. In the military jargon, procedures mean *"Standard, detailed steps that prescribe how to perform specific tasks"* [3].
- **DML-5 Techniques**. Techniques are specific ways of executing single steps of an attack. In the military jargon, techniques mean *"Non-prescriptive ways or methods used to perform missions, functions, or tasks"* [3].
- **DML-6 Tactics**. To detect a tactic means to understand how the attack has been designed and executed in terms the techniques, procedures and tools used. In the military jargon, tactics mean *"the employment and ordered*

arrangement of forces in relation to each other" [3].

- **DML-7 Strategy**. This is a non-technical high-level description of the planned attack. There are typically multiple different ways an attacker can achieve its goals, and the strategy defines which approach the threat agent should follow.
- **DML-8 Goals**. The motivation for the attack can be described as a goal. Depending on how the attacker is organised, the goal might not be known for the attack team executing the attack, the team might only receive a strategy to follow.
- **DML-9 Identity**. The identity of the attacker, or the threat agent, can be the name of a person, an organisation or a nation state. Sometimes, the identity can only be linked to other attacks without any other indication of who they are or from where they operate. The attacker identity might not be relevant to the defender if they only want to get the attacker out of the network. However, it is often important to be able to connect multiple attacks to the same actor in order to predict strategy, tactics, techniques and procedures expected to be used. This is an additional level defined by us, the original DML model [2] only consists of the levels 0–8.

The challenge is to leverage observed attack features detected at low levels to determine derivative causes at higher levels.

Assume that a given company *B* has as goal to beat company *A* in the open market. This goal might cause company *B* to use unethical means, with a strategy to steal secret information from company *A* in order to improve their own products and market position. Company *B*'s tactics may be to gain access to company *A*'s internal servers based on an attack plan with techniques, procedures and tools. Finally, the execution of the plan causes traces of the attack to be left in the network of victim *A*.

The cyber incident response team will first detect the traces, and from there must try to figure out what has happened and then decide the appropriate response. The traces are indicators, and the task of determining what really happened is a form of abductive reasoning which consists of using the indicators as classifiers to determine the nature and origin of the attack.

Most incident response teams of today are working on DML-1 and DML-2. Some are working on DML-3 and partly DML-6. However, the further up the stack you get the more seldom you find machine readable results from the analysis and work that is done. Defining semantic models for the type of information gathered in the higher levels of the DML model and the relations between them will enable more teams to increase their maturity level. Information sharing will also be facilitated by this development.

III. Elements of Semantic Threat Modelling

Discovering the real nature of a threat given a set of data or information requires a semantic model to represent all aspects of the threats with no room for ambiguous input. The further down the DML model you get, the more precise an identification can be done. The further up, the more costly a change is for the attacker and the more robust your conclusion of identity may become. Both aspects are useful for different roles and situations throughout a security incident. SIEM (Security Incident and Event Management) tools typically use semantic representation of host & network artifacts at the lower levels of the DML model, but rarely provide semantic representations of high level aspects. It is thus necessary to standardise the semantic representations of high level aspects in the DML model. This will allow automated reasoning to leverage the potential of machine learning and classifiers to do advanced cybersecurity analytical reasoning.

A. A Semantic Threat Classification Model

The primary focus of the DML model is to indicate levels of maturity in cyberthreat detection. However, the same model can be used as a basis for the design of cyberthreat classifiers, and we call this new model the *semantic threat classification model* (STCM).

Figure 3 shows the STCM which consists of a compact representation of the DML model combined with *classifiers* representing the analytical relationships from low level features to high level features.

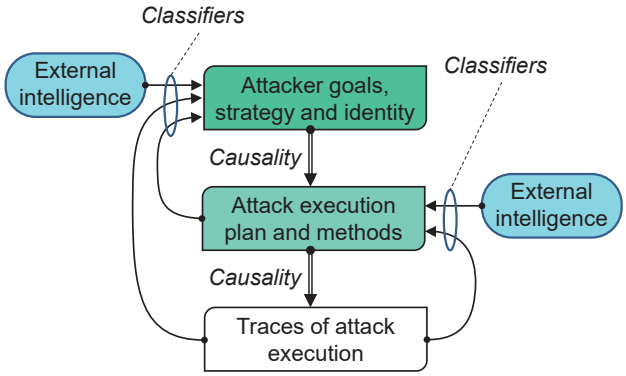


Figure 3. Semantic Threat Classification Model

Note that there are causal relationships from high level features to low level features. Hence, classifiers are used to reason in the opposite direction to that of causal relationships.

In machine learning and statistics, classifiers are used to determine categories to which some observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership is known. For cybersecurity analytics, a classifier can e.g. be used to determine which type of attack a set of network artifacts belong to (i.e. are caused by), the goal of the attacker or even the identity of the attacker.

Note that contextual information can also be used as input indicators for classifiers. Contextual intelligence can e.g. be political events covered by the media. A political conflict between nation states can make it more likely that states launch specific types of cyberattacks against each other.

The challenge for developing reliable classifiers is to identify appropriate semantic features and their variables at each

level of abstraction, and to have available sufficient amount and type of data in order to give the classifiers sufficient training for reliable detection and classification.

The design of classifiers for machine learning is heavily dependent on statistical methods, and several authors have pointed out the importance of mathematics for cybersecurity [4].

B. Semantic Feature Extraction

Stillions' DML model [2] uses English prose to informally define each level of abstraction. The use of classifiers, however, requires formal definitions of the features at each level of abstraction. Our approach is to gather informal descriptions of goals, strategies, tactics, techniques and procedures from the literature. Through analysis of these informal descriptions, we derive tuples that describe each level of abstraction. In the following, we illustrate this process for the abstraction level "Goals".

Stillions mentions the following goal as an example:

> Replicate Acme Company's Super Awesome Product Foo in 2 years or less [2]

If we ignore the time dimension of this goal, then we can derive the 2-tuple ("Replicate", "Product") from the informal description.

From Mandiant's APT1 report [5], we can derive the following goals: ("Replicate", "Product"), ("Replicate", "Manufacturing process"), ("Obtain", "Business plan"), ("Obtain", "Policy position").

Another goal can be derived from Symantec's blog post on the "Cadelle" and "Chafer" APT groups [6]: ("Monitor", "Individuals").

By generalising the examples above, we get the following definition of a goal: (Action, Object). When we observe the 2-tuples from the examples, we identify two challenges. The first challenge is that we use strings to describe each element of the tuple. If we use 2-tuples of strings in a system where a multitude of analysts and classifiers identify and record new goals, then the result will be duplicated by synonyms resulting in an explosion of features. In order to avoid this, our goal is to define a formal taxonomy of goals, where each tuple contains references to the taxonomy.

The second challenge is that the second element of the 2-tuple is too general. To alleviate this, we must define sub-elements that are more specific, e.g. that the "Product" in the first example is manufactured by "Acme company", and that the specific product is "Super Awesome Product Foo". In the last example, "Individuals" could have a sub-element "Iranian Citizens". Note that in some cases we will not be able to determine these sub-elements due to insufficient data.

Applying this approach to all the layers of abstraction in the extended DML model requires a monumental amount of effort. We believe that in order to achieve this, a community effort is needed. Thus, one of our primary goals is to lay the foundations for such an effort. Furthermore, re-using existing standards and taxonomies where applicable can significantly reduce the amount of work needed. A good example of such re-use can be observed for the abstraction level "Techniques". The MITRE ATT&CK taxonomy [7] has already defined more than 100 techniques used by adversaries in the post-compromise phases of an attack.

C. Current Initiatives for Cyberthreat Representation

There are several initiatives currently being used for representation and sharing of data on the different levels of the DML model. The following initiatives are seen as useful and may be used when selecting features for representation on the different levels:

- **INTEL Threat Agent Library** (TAL) [8] was suggested in 2007 and provides a consistent reference describing the human agents that pose threats to IT systems and other information assets. This library may serve as a feature of "Identity" in our semantic threat modelling.
- **STIX** [9] is a language for having a standardized communication for the representation of cyberthreat information. It is well known in the incident response community, but not serving the purpose of describing all aspects of cyber threats. The main shortcoming in the current version is the lack of separation between tactics, techniques and procedures.
- **CAPEC** The objective of the Common Attack Pattern Enumeration and Classification (CAPEC) [10] effort is to provide a publicly available catalog of common attack patterns classified in an intuitive manner, along with a comprehensive schema for describing related attacks and sharing information about them. CAPEC is run by MITRE and is openly available for use and development for the public. For our semantic threat modelling it may be used when describing 'Tactics' and 'Techniques'.
- **ATT&CK** is a common reference for post-compromise tactics, techniques and tools [7] run by MITRE. ATT&CK and CAPEC are related and do not exclude use of each other.

IV. Example applications of semantic cyberthreat models

In this paper, we argue that semantic cyberthreat models can help cybersecurity professionals to be more effective and efficient. This section presents some concrete examples from our own experience that support this hypothesis.

A. Incident response

Breaches due to attacks from advanced persistent threats (APTs) are often detected post-compromise. APTs quickly initiate lateral movement after the initial compromise, so assessing the scope of the breach can be challenging. In order to assess the scope of the breach, we need to know how the threat agent operates and what kind of indicators, artifacts, tools, tactics, techniques and procedures (TTPs) we should search for. The incident response analysis process typically consists of the following steps:

1) Evidence collection
2) Analysis of evidence

3) Identification of new indicators, artifacts, tools and TTPs
4) Threat agent attribution

Steps 1-3 are performed in an iterative fashion. The analysis results may indicate that we need to collect more evidence, or that we should search the existing evidence for new indicators. If we are able to perform step 4 and attribute the breach to a known threat agent, then we can leverage our historical knowledge of this threat agent. We can use this knowledge to guide our evidence collection and analysis. We have used the MITRE ATT&CK taxonomy [7] to be able to quickly compare our evidence to known threat agents during incident response. By manual analysis, we found threat agents that used tools and techniques very similar to what we observed in our evidence. The ATT&CK taxonomy [7] has a loose semantic model connecting threat agents, tactics, techniques and tools. It does not model procedures, artifacts or indicators. In order to automate the analysis of threat agent similarities, we implemented a simple semantic model using a graph database. The model linked threat agents to observed indicators, artifacts, tools and TTPs. We then used the graph database to find all subgraphs that connected the findings from our incident to known threat agents. The result enabled us to attribute the evidence from our incident to a known threat agent, and the results helped guide our evidence collection and analysis. Another great advantage of using such a model is that the attribution hypothesis can be re-tested as more knowledge is added to the graph, in order to avoid confirmation bias. Our experience from this incident was that we were able to attribute the evidence to a known threat agent much more rapidly than by using manual analysis. We were also able to fully document all relations between our evidence and the threat agent by issuing a simple graph query.

B. Requests for information

A common task for threat intelligence analysts is to find all information related to a single data point, e.g. an IP address, a malware sample or a threat agent. Having a semantic model implemented as a graph makes it possible to complete such a task quickly and reliably by issuing a single graph query.

C. Intrusion detection

Current intrusion detection systems operate at DML-1, DML-2 and/or DML-3. One of the challenges with operating at DML-4 and above is that TTPs are commonly described using English prose, i.e. as unstructured data. This makes it challenging to translate the description to intrusion detection signatures, and signature development must be performed manually. Defining formal models for TTPs makes it possible to automatically generate signatures from structured data when a new TTP is defined. One concrete example is the procedure described in [11]:

> An example would be an adversary running **net time**, followed by the **AT.exe** command to schedule a job to kick off just one minute after the current local time of the victim system. [11]

Given an endpoint security solution that logs process execution with arguments and command inputs/outputs, a human analyst could write a signature to detect this procedure. The signature would have to detect the following:

1) Execution of **net.exe** with **time** as the first argument and **victim system** as the second argument
2) Timestamp returned by the command in step 1
3) Execution of **at.exe** with **victim system** as the first argument and ((timestamp from step 2) + 1 minute) as the second argument

Interpreting the description "to schedule a job to kick off just one minute after the current local time of the victim system" is easy for a human, but very difficult for a computer. A formal definition of this procedure would make it possible for a computer to automatically generate signatures for the procedure by applying transformation rules.

V. Conclusion

Semantic modelling of threats is a promising approach for automated threat and attack detection at multiple levels of abstraction. A semantic model of threats will enable security analysts to work faster and more efficiently in terms of identifying threat agents and take advantage of previous experience and gathered intelligence when handling incidents caused by known or unknown threat agents. The task of extracting semantic features for all levels of abstraction in our suggested extended DML model is an undertaking of daunting proportions. In order to make this task manageable the reuse of related standards and taxonomies is required.

References

[1] ISO, *ISO/IEC 27005:2011 - Information technology – Security Techniques – Information security risk management (second edition)*. ISO/IEC, 2011.

[2] R. Stillions, "The DML Model," http://ryanstillions.blogspot.com/2014/04/the-dml-model_21.html, 22 April 2014.

[3] U. DoD, *Department of Defense Dictionary of Military and Associated Terms*. Joint Chiefs of Staff, 2010.

[4] A. Pinto, "Secure because of Math: A deep-dive on Machine Learning-Based Monitoring," Black Hat Briefing. BlackHat Conference, 2014.

[5] Mandiant, "Mandiant APT1," https://www.fireeye.com/content/dam/fireeye-www/services/pdfs/mandiant-apt1-report.pdf, 18 February 2013.

[6] S. S. Response, "Iran-based attackers use back door threats to spy on Middle Eastern targets," http://www.symantec.com/connect/blogs/iran-based-attackers-use-back-door-threats-spy-middle-eastern-targets, 7 December 2015.

[7] MITRE, "Adversarial Tactics, Techniques and Common Knowledge (ATT&CK)," https://attack.mitre.org/.

[8] T. Casey, "Threat agent library helps identify information security risks," *Intel White Paper, September*, 2007.

[9] S. Barnum, "Standardizing cyber threat intelligence information with the Structured Threat Information eXpression (STIX)," *MITRE Corporation*, vol. 11, 2012.

[10] MITRE, "Common Attack Pattern Enumeration and Classification (CAPEC)," https://capec.mitre.org/.

[11] R. Stillions, "On TTPs," http://ryanstillions.blogspot.com/2014/04/on-ttps.html, 22 April 2014.