STIDS 2015

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

Semantics in Cyber-Physical Systems

November 18-20, 2015

Johnson Center
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 Tenth International Conference on Semantic Technology for Intelligence, Defense, and Security (STIDS 2015) 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 2015 theme is Semantics in Cyber-Physical Systems. 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 2015

Ian Emmons and Kathryn Laskey
STIDS 2015 Technical Chairs

Paulo Costa and Alessandro Oltramari
STIDS 2015 General Chairs

STIDS 2015 Committees

Program Committee

Carl Andersen	BBN Technologies
Erik Blasch	AFRL
Rommel Novaes Carvalho	Brazil's Office of the Comptroller General
Werner Ceusters	SUNY at Buffalo
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
John Hebeler	University of Maryland, Baltimore County
Edward Huang	George Mason University
Terry Janssen	Quantum Cybersecurity Systems LLC
Gregory Joiner	BBN Technologies
Anne-Laure Jousselme	NATO Centre for Maritime Research and Experimentation (CMRE)
Mieczyslaw Kokar	Northeastern University
Dave Kolas	BBN Technologies
Kathryn Laskey	George Mason University
Louise Leenen	CSIR
David Mireles	Raytheon BBN Technologies
Ranjeev Mittu	US Naval Research Laboratory
Jeffrey Morrison	Office of Naval Research
Leo Obrst	MITRE Corporation
Alessandro Oltramari	Carnegie Mellon University, CyLab
Mary Parmelee	MITRE Corporation
Gregor Pavlin	Thales Group
Andrew Perez-Lopez	BBN Technologies
Plamen Petrov	Raytheon BBN Technologies
Setareh Rafatirad	George Mason University
Douglas Reid	Google
Patrice Seyed	Rensselaer Polytechnic Institute

Barry Smith	SUNY Buffalo
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

Amanda Vizedom

Tutorials Chair

Mary Parmelee

Classified Session Chair

Brian Haugh

Local Team (GMU)

Debra Schenaker (Administrative Chair)

Priscilla McAndrews

Tamara Day

Alexandre Barreto

Ricardo Fernandes

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:

- *2015 Michael Dean Best Paper:* Noam Ben-Asher, Alessandro Oltramari, Robert F. Erbacher, Cleotilde Gonzalez. Ontology-based Adaptive Systems of Cyber Defense.
- *Runner-up:* Kathryn Blackmond Laskey, Sudhanshu Chandekar, Bernd-Peter Paris. A Probabilistic Ontology for Large-Scale IP Geolocation.

2015 Michael Dean Award Committee

Erik Blasch	AFRL
Richard Haberlin	EMSolutions, Inc.
Peter Haddawy	Mahidol University
Anne-Laure Jousselme	NATO Centre for Maritime Research and Experimentation (CMRE)
Louise Leenen	CSIR
Ranjeev Mittu	US Naval Research Laboratory
Leo Obrst	MITRE Corporation
Brian Ulicny	Thomson Reuters
Amanda Vizedom (Chair)	Criticollab, LLC
Andrea Westerinen	Nine Points Solutions, LLC

Invited Speaker, November 19: Dr. Mark Hartong

Positive Train Control Critical Infrastructure

Positive Train Control (PTC) is a supervisory control and data acquisition system designed to protect against loss of locomotive crew situational awareness that could result in train-to-train collision, train derailments due to excessive speed, train incursion into roadway work zones, and train movements through misaligned switches. Multiple technological approaches are being implemented by different railroads. The complexity of these systems of systems, coupled with the lack of a common ontological description, complicates creation of interoperable systems and implementation of adequate critical infrastructure protections. This talk will introduce PTC, its operation, and associated implementation issues, which could be used as the basis for the development of an appropriate ontological model that supports formalization of interoperability and infrastructure security.

Biography: Dr. Mark Hartong

Dr. Mark Hartong is the Senior Scientific and Technical Advisor for Railroad Electronics at the Federal Railroad Administration, US Department of Transportation. The Federal Railroad Administration is the regulatory and enforcement agency responsible for promoting safe and secure railroad transportation within the United States. Mark serves as the agency's senior technical authority with respect to the application of safety and security critical electronics and software for use in the railroad environment. He also plays an integral part in the agency's regulatory and enforcement agenda in the creation of new federal regulations, determination of technical merits of requests for relief by railroads and vendors from existing federal regulations and evaluations of regulatory non-compliance.

Before joining the Federal Railroad Administration, Mark was a Staff Systems Engineer with Lockheed Martin Corporation providing systems engineering support for a wide range of classified and unclassified communications hardware and software development programs exploring state of the art communications systems and networking technologies. This included combat and communications systems for Virginia and Seawolf nuclear attack submarines as well as Trident class nuclear ballistic missile submarines, combat command and control systems for Kuwait, Saudi Arabia, and Qatar, and the US Defense Message System.

Prior to his employment at Lockheed Martin, Mark served on active duty in the US Navy as a Naval Engineering Duty Officer. A navy line technical specialist qualified in submarine warfare, he provided technical, acquisition and fleet industrial leadership to meet national defense needs for ships, submarines, and their associated warfare support systems. This covered hull systems, mechanical and electrical systems; combat systems; and command, communications and electronics.

Mark has a BS in Mechanical Engineering from Iowa State University, an MS in Computer Science from the Naval Postgraduate School, and both an MS in Software Engineering with a certificate in Information System Security as well as a PhD in Information Technology from George Mason University. He is also a Registered Professional Engineer.

Invited Speaker, November 19: Dr. Bruno Sinopoli

On the Security of Cyber-Physical Systems

Cyber Physical Systems (CPS) refer to the embedding of widespread sensing, computation, communication, and control into physical spaces. Application areas are as diverse as aerospace, chemical processes, civil infrastructure, energy, manufacturing and transportation, most of which are safety-critical. The availability of cheap communication technologies such as the Internet makes such infrastructures susceptible to cyber security threats, which may affect national security as some of them, such as the power grid, are vital to the normal operation of our society. Any successful attack may significantly hamper the economy, the environment or may even lead to loss of human life. As a result, security is of primary importance to guarantee safe operation of CPS.

In an offensive perspective, attacks of this sort can be carried out to disrupt the functionality of the enemy's critical infrastructures without destroying it or even being directly identified. Stuxnet, the malware at the root of the destruction of centrifuges employed to enrich uranium in Iran's nuclear facilities, is a clear example of how strategically important it is to gain a deep understanding of CPS security. In this talk, I will provide an introduction to CPS security, give an overview of recent results from our research group as well as directions for future work.

Biography: Dr. Bruno Sinopoli

Dr. Bruno Sinopoli received the DEng degree from the University of Padova in 1998 and his MS and PhD in Electrical Engineering from the University of California at Berkeley, in 2003 and 2005 respectively. After a postdoctoral position at Stanford University, Dr. Sinopoli joined the faculty at Carnegie Mellon University where he is an associate professor in the Department of Electrical and Computer Engineering with courtesy appointments in Mechanical Engineering and in the Robotics Institute and co-director of the Smart Infrastructure Institute, a research center aimed at advancing innovation in the modeling analysis and design of smart infrastructure. Dr. Sinopoli was awarded the 2006 Eli Jury Award for outstanding research achievement in the areas of systems, communications, control and signal processing at U.C. Berkeley, the 2010 George Tallman Ladd Research Award from Carnegie Mellon University and the NSF Career award in 2010.

Advances in very large-scale integration and micro-electromechanical system technology have boosted the development of micro sensor integrated systems. Such systems combine computing, storage, radio technology, and energy source on a single chip. When distributed over a wide area, networks of these embedded devices can perform a variety of tasks that range from environmental monitoring and military surveillance, to navigation and control of a moving vehicle.

A common feature of these systems is the presence of significant communication delays and data loss across the network. From the point of view of control theory, significant delay is equivalent to loss, as data needs to arrive to its destination in time to be used for control. In short, communication and control become tightly coupled such that the two issues cannot be addressed independently. Bruno Sinopoli's research interest focuses on the analysis and design of networked embedded control systems, with applications to sensor actuators networks.

Invited Speaker, November 20: Dr. Alexander Kott

The Unbearable Lightness in the Meaning of Cyber Risk

The term "cyber risk" aims to characterize a variety of phenomena where information assets are subject to a potential damage due to cyber attacks. Many attempts, almost unblemished by success, have been made to define cyber risk. In this talk we explore why the concept of cyber risk, as treated by both practitioners and researchers of cyber security, is largely inconsistent with definitions of cyber risk commonly offered in the literature. Unsurprisingly, an adequate ontology of cyber risk is lacking, and a rigorous re-conceptualization of cyber risk is needed.

A new formal treatment of cyber risk that should include an ontology of cyber risk-related concepts, a rigorous mathematical model, and practical definitions, all of which must align with common sense perceptions of cyber risk by cyber security practitioners. Furthermore, cyber risk belongs firmly to the realm of adversarial decision-making and has little meaning outside of a process geared toward decisions made under extreme uncertainty, time pressure, and under threat of an adversarial actions pre-empting and counteracting those of the defenders. Modeling of risk must be adversarial in nature, with game-theoretic and decision-theoretic perspectives duly considered. And to add to the confusion, we must discuss how inseparably connected yet different are cyber risk and cyber resilience.

Biography: Dr. Alexander Kott

Dr. Alexander Kott serves as the Chief, Network Science Division, Army Research Laboratory headquartered in Adelphi MD. In this position, he is responsible for fundamental research and applied development in performance and security of both tactical mobile and strategic networks. He oversees projects in network performance and security, intrusion detection, and network emulation. Research under his direction brings together government, industry and academic institutions working toward a fundamental understanding of interactions, interdependencies, and common underlying science among social/cognitive, information, and communications networks, including science for cyber. Prediction and control of the composite behavior of these complex interacting networks will ultimately enhance their effectiveness and security.

Between 2003 and 2008, Dr. Kott served as a Defense Advanced Research Programs Agency (DARPA) Program Manager responsible for a number of large-scale advanced technology research programs. His earlier positions included Technical Director with BBN Technologies, Cambridge, MA; Director of R&D at Logica Carnegie Group, Pittsburgh, PA; and IT Research Department Manager at AlliedSignal, Inc., Morristown, NJ. Dr. Kott received the Secretary of Defense Exceptional Public Service Award and accompanying Exceptional Public Service Medal, in October 2008.

He earned his PhD from the University of Pittsburgh, Pittsburgh PA in 1989; published over 80 technical papers; and co-authored, and edited nine technical books. Dr. Kott and his family reside in Silver Spring, Maryland. He can be reached at alexkott@yahoo.com.

Invited Speaker, November 20: Dr. James Momoh

Resilient Power Grids

Smart grid system deployment has been a major point of concern and interest in the development of the future electric grid both here in the US and abroad. Variety of definitions, semantics, interpretations of its functionality have been given by designers, implementers, end users, standard and security organizations and university communities. Several functions and applications have been proposed in the electric power industry and other related fields have yet to be properly measured by the designers.

To ensure that the capability of smart grid and its functionality are understood generally stakeholders need performance metric such as resilience, sustainability and reliability and efficiency to measure its deployment.

In this presentation we plan to provide a working definition, architecture and test beds in common use as well as define some of the functionality of smart grid for its deployment in the 21st century for the development of future electric grid.

Further, we will address the grand challenge problems and options for assessing the overall performance of the future grid, which will include identification of unified knowledge from different disciplines to address some of the challenge problems, will be highlighted in the presentation.

On going research activities and open research questions to further improve the high performance smart grid, which will be of value to university researchers, developers and policy makers will be discussed.

Biography: Dr. James Momoh

Dr. James Momoh, Professor and Director, Center for Energy Systems and Control (CESaC) Howard University, received a BS in Electrical Engineering from Howard University in 1975, a MS in Electrical Engineering from Carnegie Mellon University in 1976, a MS in Systems Engineering from the University of Pennsylvania in 1980 and a PhD in Electrical Engineering from Howard University in 1983. He was Chair of the Electrical Engineering Department at Howard University and Director of the Center for Energy Systems and Control. In 1987, Momoh received a National Science Foundation (NSF) Presidential Young Investigator Award. He was Program Director of the Power program in the Electrical and Communications Systems (ECS) Division at NSF from 2001-2004. Momoh is a Fellow at the Institute of Electronics and Electrical Engineering (IEEE) and a Distinguished Fellow at the Nigerian Society of Engineers (NSE). He was inducted as a Fellow Member of Nigerian Academy of Engineering (NAE) in 2004.

Momoh's current research activities for utility firms and government agencies span several areas in systems engineering, optimization and energy systems control of terrestrial, space and naval complex and dynamic networks. These include but are not limited to the development of multi-agent, intelligent optimization technologies; next-generation optimization for the design of future intelligent power grids; computational tools and algorithms for deregulated/restructured power economies; and advanced power management strategies for stressed power systems with uncertainty, dynamics and stochasticity of parameters. He has also led research and education outreach and collaborations in information technology, environment, energy and human capacity building to involve the United States and other countries worldwide. This has led to a number of international conferences, workshops and seminar series, and research and education in engineering programs that are sponsored by NSF, Howard University and several universities and public-private agencies.

Presently, he is developing interdisciplinary research and education programs in power, economics and environmental adaptive systems. The goal is to build cross-disciplinary partnerships among engineering, economics and other related disciplines that address socioeconomic issues, environmental issues, new teaching pedagogy and curricula to prepare the workforce of the future.

Momoh's research and professional activities have led to over 225 technical papers in refereed journals, transactions, proceedings and also production of several textbooks in his areas of expertise. These papers are presented at conferences, workshops, seminars, tutorial sessions and several other IEEE events to benefit the wider community of engineers, students and policy makers. He has contributed to and is engaged in the development of specialized computational applications of classical optimization, intelligent systems and advanced optimization techniques for the new tools needed by terrestrial, naval and space power systems. In particular, he has been developing special topical contributions in the area of Dynamic Stochastic Optimal Power Flow (DSOPF) using Adaptive Dynamic Programming (ADP) methods. His activities also extend to the development of Multi-Agent Systems (MAS) for coordination and control of complex power systems. His work continues to impact the research and innovations needed in optimization for planning and operational security, efficiency, reliability and stability, and autonomous control of sustainable energy systems.

Table of Contents

Technical Papers

Joint Doctrine Ontology: A Benchmark for Military Information Systems Interoperability

Peter Morosoff
E-Maps, Inc.,
Fairfax, VA
peter.morosoff@e-mapsys.com

Ron Rudnicki
CUBRC
Buffalo, NY
rudnicki@cubrc.org

Jason Bryant
Air Force Research Lab
Rome, NY
jason.bryant.8@us.af.mil

Robert Farrell
Air Force Research Lab
Rome, NY
robert.farrell.10@us.af.mil

Barry Smith
University at Buffalo
Buffalo, NY
phismith@buffalo.edu

***Abstract*—When the U.S. conducts warfare, elements of a force are drawn from different Services and work together as a single team to accomplish an assigned mission on the basis of joint doctrine. To achieve such unified action, it is necessary that specific Service doctrines be both consistent with and subservient to joint doctrine. But there are two further requirements that flow from the ways in which unified action increasingly involves not only live forces but also automated systems. First, the information technology that is used in joint warfare must be aligned with joint doctrine. Second, the separate information systems used by the different elements of a joint force must be *interoperable*, in the sense that data and information that is generated by each element must be usable (understandable, processable) by all the other elements that need them. Currently, such interoperability is impeded by multiple inconsistencies among the different data and software standards used by warfighters. We describe here the on-going project of creating a Joint Doctrine Ontology (JDO), which uses joint doctrine to provide shared computer-accessible content valid for any field of military endeavor, organization, and information system. JDO addresses the two previously-mentioned requirements of unified action by providing a widely applicable benchmark for use by developers of information systems that will both guarantee alignment with joint doctrine and support interoperability.**

Keywords—joint doctrine, military doctrine, ontology, Basic Formal Ontology (BFO), Common Core Ontologies (CCO), joint warfare, unified operations, interoperability, terminology, definition

I. Joint Doctrine

The publications of joint doctrine document fundamental principles and overarching guidance for the employment of the Armed Forces of the United States [1]. Joint doctrine applies to all military, from the joint staff to commanders of combatant commands, their supporting commands, and to the individual Services, each of which has its own Service-specific doctrinal publications. Joint doctrine is authoritative in the sense that, if conflicts arise between it and Service doctrine, then the former – absent more current and specific guidance from the Chairman of the Joint Chiefs of Staff – will take precedence.

Joint doctrine provides the benchmark for interoperability of the separate Service doctrines. And because all Service-level terminology is dependent on joint doctrine it is critical, if we are to prevent higher-level flaws cascading to domain-level doctrinal errors, that the terms of joint doctrine be defined correctly.

It is commonly supposed that doctrine provides not hard and fast rules but rather merely a loose and always revisable guide to action that is typically abandoned on first contact with the enemy. Doctrine is however authoritative also in the sense that it is to be followed in all cases except when, in the judgment of the commander, exceptional circumstances dictate otherwise. Moreover, there are many doctrinally acknowledged features of military action that survive through every engagement. Doctrine defines the shared frame of reference that remains active through every phase of every military engagement. This is because doctrine provides the principles that determine how to understand the authorized command relationships and the authority that military commanders can use. It establishes common ways of accomplishing military tasks and facilitates readiness by promoting coordination of training and planning. Most importantly for our purposes here, doctrine provides a common lexicon – a set of precise terms and precise definitions – expressed in a language that is designed to enable consistent understanding by military leaders, planners and educators. Doctrine thereby enables the sort of effective communication among warfighters that is needed for unified action.

II. Battle Management Language (BML)

While doctrine has been developed and used thus far to satisfy the needs of human beings, it is increasingly understood that it must also satisfy requirements that come into play when information systems are brought to bear in military action. The language used by warfighters and codified in field manuals and doctrinal lexica still involves some of the ambiguities characteristic of all languages used by human beings. But such ambiguities can be tolerated where human beings are involved because humans can easily disambiguate the meanings of ambiguous terms in everyday contexts of use.

The very human-friendliness of the language used by warfighters brings an equal and opposite weakness, however, when information systems are involved. Computers have difficulties in interpreting the common language of human beings and in using contextual cues to resolve ambiguities. Attempts to overcome these difficulties

led in around 2000 to the conception of the Battle Management Language (BML) [2] that was designed to allow the description of a commander's intent in the sort of context-free way that would support processing by automated systems. The initial goal of BML was to create a unified and unambiguous representation of command and control (C2) doctrine as a 'systematic data model' [2]. BML was seen as thereby providing a unified framework that would not only remove ambiguities but also rectify the terminological disunities created through the continued dominance of disparate Service cultures and of the numerous communities of interest within those cultures [3].

III. Interoperability

BML continues as an active project [4]-[5], especially in the modeling and simulation community. The promised unambiguous representation of the content of C2 doctrine using BML has, however, not been achieved. Here, we take up once again the idea of formalizing joint doctrine by drawing on more recent developments in the field of ontology. Our target, however, is more modest. It is not to provide the resources to capture formally a commander's intent. Rather, we seek to capture in a computer-usable form the terminological content of joint doctrine in a way that will support the sort of interoperability that is needed where live forces need to engage in unified action with information systems.

Interoperability is defined in the Glossary of DoD Instruction (DoDI) 8330.01 [6] as:

> The ability of systems, units, or forces to provide data, information, materiel, and services to, and accept the same from, other systems, units, or forces, and to use the data, information, materiel, and services exchanged to enable them to operate effectively together. IT interoperability includes both the technical exchange of information and the end-to-end operational effectiveness of that exchange of information as required for mission accomplishment.

Our hypothesis is that the creation of a Joint Doctrine Ontology (JDO) can provide a widely applicable benchmark for use by developers of information systems that will support rather than impede unified action by breaking down existing terminological silos of different Services and communities of interest.

In contrast to the BML, our alternative approach begins, not with defining a new language, but rather with the existing authoritative controlled vocabulary that is defined in Joint Publication 1-02, the *Department of Defense Dictionary of Military and Associated Terms* [7].

The JP 1-02 dictionary consists in its current version of some 2,803 terms drawn from some 81 approved doctrinal publications forming the Joint Doctrine Hierarchy (at http://www.dtic.mil/doctrine/doctrine/status.pdf). In effect, we are constructing JDO as a shadow of JP 1-02, incrementally adding definitional enhancements and further elements of logical regimentation, but in such a way that the ontology, and the dictionary that underlies it, remain synchronized with each other through future revisions of joint doctrine. In effect, JDO will provide a semantic enhancement of JP 1-02, and therefore also of the terminological content of the separate Joint Publications from which the terms and definitions of JP 1-02 are derived.

The Dictionary defines the standard U.S. military and associated terminology needed to enable the joint activity of the Armed Forces of the United States. As stated in the Preface signed by Vice Admiral William E. Gortney, Director of the Joint Staff, these military and associated terms, together with their definitions, constitute approved Department of Defense (DOD) terminology for general use by all DOD components. [7]

In multiple other joint publications, as well as in a series of DoD and Chairman of the Joint Chiefs of Staff instructions, it is required that all DoD initiatives, as well as all warfighters and warfighting organizations, should use a common terminology. In addition, instructions state that all IT intended for use in military operations should be designed from the beginning to be interoperable (paragraph 9b of Chapter 2, "Doctrine Governing Unified Direction of Armed Forces," JP 1 [1]). We believe that it follows from these instructions that all DoD IT efforts, insofar as they are intended for use in military operations, should be developed in such a way as to be interoperable with joint doctrine.

IV. Failures of Interoperability and Requirement for Effective Guidance to IT Developers in the Future

The need for interoperability of DoD information systems and for alignment of the data and information that enables military action has been recognized repeatedly and at the highest levels, and given today's and tomorrow's flood of digital data across networks this need is becoming ever more apparent.

For example, DoDI 8320.02, "Sharing Data, Information, and Information Technology (IT) Services in the Department of Defense" [8] requires that authoritative data sources (ADSs) be 'registered in the DoD Data Services Environment (DSE).'

The DoDI directs further that 'Data, information, and IT services will be made … interoperable throughout their lifecycles for all authorized users'. However, the instruction to achieve such interoperability – namely through 'enforcement of policy for DoD metadata that uses Government and industry metadata standards' – repeatedly fails in its goal. This is not only because the policy is formulated in a way that falls short of the required specificity, but also because, even where relevant standards exist, they have in almost all cases been created *ad hoc*, to address specific local needs. Thus they have not been built in the sort of coordinated, rule-governed way that would be needed to achieve interoperability.

The problem of overly weak requirements is illustrated also in the already mentioned DoDI 8330.01 on "Interoperability" [6], where it is stated that the information systems that DoD components use

> must interoperate, *to the maximum extent practicable*, with existing and planned systems (including applications) and equipment of joint, combined, and coalition forces, other U.S. Government departments and agencies, and non-

> governmental organizations, as required based on the operational context (italics added).

Because everything is by definition interoperable to 'the maximum extent practicable,' this instruction is without teeth.

DoDI 8320.02 suggests a further route to the achievement of interoperability through adherence to standards listed in the DoD IT Standards Registry (DISR). Unfortunately, it is difficult to determine the degree of interoperability of DISR standards, in part because the needed assessment must be applied simultaneously to the different portions of the DISR, and these often require different sorts of permissions (and thus, we assume, are accessible only to different sorts of people). Some of the resources contributed to the DISR that we were able to access, however, do not manifest – even when taken singly – the sort of minimal terminological consistency or formal regimentation that would be needed to meet the demands of interoperability. The terminology defined in [9], for example, was created by selecting terms and definitions from a wide range of sources. No common rules for definitions were employed, and so there is no way of checking even for simple logical consistency of the resulting artifact.

Achieving interoperability – both terminological and structural [10] – is of course difficult for a large organization like the DoD with a cumbrous history of information system development. However, in recent years a number of best practices for meeting the demands of inoperability have been established, some of them very simple to implement. Thus a first task would be to establish corresponding simple rules that must be satisfied by IT systems developed by the DoD in the future. We are concerned here only with issues of *terminological* interoperability, which we propose should be addressed through the creation of a benchmark ontology framework centered around the JDO. We envisage that the complementary *structural* interoperability might be tackled in part through the deployment of W3C standard resources such as RDF and the Web Ontology Language (OWL) [11]. The formulation of ontologies using OWL, in particular, would allow computational reasoners to be used in a way that provides automatic checking for consistency of definitions with each new revision of a terminological artifact such as JP 1-02. The ontology approach can thereby support agile development and coordinated maintenance of information systems in a way that does not sacrifice terminological interoperability [12]–[16].

V. The Solution

DoDI 8330.01 [6] already requires that *the content of joint operational concepts, and associated doctrine and operational procedures, address interoperability* of the IT used by the separate Services and also, where required, by joint and multinational forces and other U.S. government departments and agencies.

While DoD thus requires that joint doctrine addresses the need for IT interoperability, it crucially does not require – and has no effective strategy to ensure – that the IT systems and procedures themselves address the need for conformity with joint doctrine. We believe, however, that such conformity is not only indispensable if unified action between human warfighters and IT systems is to be achieved, but further that it would bring multiple significant benefits to military IT systems themselves, and thus also to the developers of such systems, because it would provide a benchmark for interoperability.

VI. Unified Action of Human Warfighters and Information Systems

JP 1, the Capstone Publication of Joint Doctrine [1], states that unified action demands 'maximum interoperability':

> The forces, units, and systems of all Services must operate together effectively, in part through interoperability. This includes joint force development; use of joint doctrine; the development and use of joint plans and orders; and the development and use of *joint and/or interoperable communications and information systems* (italics added).

Because a military organization includes its information systems, we believe that building the common language provided by doctrine into the information systems that will be used by warfighters is a vital need.

The DoD Manual (DoDM) 5120.01, "Joint Doctrine Development Process" [17], provides the guidance that steers DoD to consistent terminology across the joint publications governing different types of operational domains. Developers of doctrine are required to 'use, to the greatest extent possible, previously approved terminology contained in the text of other JPs or in … JP 1-02.' An information system needs more than well-trained and qualified people and high-quality equipment to provide effective support to unified action. It must be supported also by effective guiding principles and procedures rooted in an understanding of the requirements of unified action. Our proposal is that such support can be achieved in today's networked environment by extending the same guidance that is provided to doctrine developers also to IT developers. Those engaged in developing IT systems for military operations should be required to take the terminology and definitions of joint doctrine as their starting point. Increasingly, if this proposal is adopted, doctrine developers will come to be seen as constituting the first rank of information technologists, providing the core terminological content on which all DoD IT content will rest.

VII. Rules for Definitions in Instruction 5705.01D

To see how JDO will be constructed, we need first to understand some of the features of the Dictionary from which it will be derived. The idea for such a dictionary is expressed in DoDI 5025.12 of August 2009 [18], which states that it is DoD policy to improve communications and mutual understanding within the DoD, with other Federal Agencies, and between the United States and its international partners through the standardization of military and associated terminology.

This position is restated in the Chairman of the Joint Chiefs of Staff Instruction (CJCSI) 5705.01D of November

10, 2010 on the creation of a "Standardization of Military and Associated Terminology." [19] Specifically, the Chief of the Joint Education and Doctrine Division (JEDD), J-7, shall oversee the DoD Terminology Program and U.S. participation in the NATO Terminology Programme; serve as Joint Staff planner for terminology issues; and appoint and supervise the Joint Staff terminologist.

Enclosure C of this instruction provides a "Definition Writing Guide," which includes a specification of the scope of JP 1-02 and also simple rules for writing definitions. Such rules are of obvious importance for our needs here, since an ontological counterpart of JP 1-02 can be created only if the definitions contained in the latter are in good order from the point of view of logical consistency.

As concerns scope, the Guide specifies that the Dictionary will include terms of general military or associated significance. Technical or highly specialized terms may be included if they can be defined in easily understood language and if their inclusion is of general military or associated significance. The Guide requires further that the dictionary be non-redundant: thus, a term will be added to the dictionary only if '[an] approved joint term with similar definition does not exist.'

The Guide defines a definition as 'a formal statement of the exact meaning of a term that enables it to be distinguished from any other.' A definition is distinguished from a description by the fact that the latter 'is a narrative containing information about the term that is not constrained in format or content.'

Principles for the development of a definition require that it should be:

Clear – Address the meaning of the term only. A definition should not contain doctrinal or procedural information; i.e., it should focus on describing "what" a term is and not "how" or "why" it is used.

Concise – Be as brief as possible, including only information that makes the term unique. Limit the definition to one sentence whenever possible.

Complete – Include all information required to distinguish the term from those that are related or similar.

The Guide includes also a list of types of errors that are to be avoided when writing definitions. For example, a definition should not be *over-restrictive*; it should not be *circular*; it should be *positive* (state what is covered by a term rather than what is not covered); and it should contain no *hidden definitions* (where the definition of one term is embedded inside another).

The rules codified in the Guide conform very well to the best practices identified by terminologists who have studied the authoring of definitions [20]. That violations of these rules have slipped though the coordination process, however, is seen in the fact that errors of each of the mentioned kinds can still be found (see Table 1).

Avoiding these and other types of errors would not only make JP 1-02 more valuable to human users; it would also enable the construction of the formal representations of its content of the sort that are needed for use in computational systems. We have already proposed a series of supplementary rules for the formulation of definitions (summarized in [12]), rules which have been tested in some 150 ontology initiatives over a wide variety of domains (see under 'Users' at [14]). We are applying these rules in building the JDO, thereby providing a vehicle that can support the usage of joint terminology by computers without sacrificing understandability by humans. These definitions can also be of help in the process of revising joint publications in the future, allowing the content of JP 1-02 to be used as part of a computational process of quality assurance for the use of terminology in joint publications when successive revisions are made.

	U	N	C	H
operational area =def. An overarching term encompassing more descriptive terms (such as area of responsibility and joint operations area) for geographic areas in which military operations are conducted.	x	x		x
contingency operation =def. A military operation that is either designated by the Secretary of Defense as a contingency operation or becomes a contingency operation as a matter of law (Title 10, United States Code, Section 101[a][13])			x	
subordinate command =def. A command consisting of the commander and all those individuals, units, detachments, organizations, or installations that have been placed under the command by the authority establishing the subordinate command.		x	x	

Table 1: Examples of errors in JP 1-02 (from June 15, 2015)
U = **unclear,** N = **not concise,** C = **circular,** H = **hidden definitions**

VIII. Proposed Supplementary Rules for Definitions

We provide five examples of such rules, and illustrate their application to creating the JDO.

Rule 1: *Do not confuse the entity you are defining with the term used to represent that entity.*

(Failure to heed this rule is illustrated in the definition of **operational area** in Table 1 – an operational area is not a 'kind of overarching term'.)

Rule 2: *Distinguish between general terms and proper names.*

Almost every JP 1-02 term is a general term, which is to say, it is a term that refers to something general – a kind or type (as in all the cases listed in Table 1) – having multiple specific instances. A small number of JP 1-02 terms are proper names, which is to say, they refer to exactly one specific instance. Examples include the **Universal Joint Task List** and **Joint Doctrine Development System**. Such terms are standardly marked by use of initial capitals, but their treatment in JP 1-02 is sometimes uncertain. The definition of **Army air-ground system**, for instance, suggests that there is exactly one Army air-ground system, so that 'Army air-ground system' would be a proper name.

However, there may be a plurality of such systems used by the Army at any given time.

Rules 3–5 apply only to general terms, and are satisfied already by the definitions of many such terms in JP 1-02:

> Rule 3: *All general terms should be singular in number*.
>
> Rule 4: *Each general term should have at most one single parent term.*
>
> Rule 5: *The definition of each general term* A *should specify the associated parent term* B *and state what it is about the* A*s that marks them out from all other instances of* B *as instances of* A.

Thus a definition of a general term A should have the two-part form:

> An A =def. a B which Cs.

For example (from [16]):

> **artillery vehicle** =def. A *vehicle* which *is designed for the transport of one or more artillery weapons.*
>
> **artillery weapon** = def. A *device* which *is designed for projection of munitions beyond the effective range of personal weapons.*

Returning to JP 1-02 we can now, following rule 5, define:

> **operational area** =def. A *geographic area* in which military operations are conducted. (Contrast the first row of Table 1.)

Here 'geographic area' is the parent term; the specific difference is 'in which military operations are conducted.' The overwhelming majority of JP 1-02 definitions are already of this form. Consider for example:

> **theater of operations** =def. An operational area defined by the geographic combatant commander for the conduct or support of specific military operations.

Many of the remaining cases are easily converted to be of this form without any change of meaning. Starting, for example, from the definition:

> **cyberspace operations** = def. The employment of cyberspace capabilities where the primary purpose is to achieve objectives in or through cyberspace.

Here two conversion steps are needed. The first replaces the term to be defined with a singular noun following rule 2. The second, in accordance with rule 5, adds a representation of the appropriate parent term (here, trivially, **operation**) to yield:

> **cyberspace operation** =def. An operation that employs cyberspace capabilities and has primary purpose: to achieve objectives in or through cyberspace.

Such rules may seem trivial, and the effect of their application may be very slight when measured against the understandability and utility from the point of view of human beings of the definitions to which they give rise. But they bring two immediate benefits when IT systems are brought into play. First, because IT developers lack warfighters' experience and therefore context, they need definitions with as little ambiguity as possible. And second, the changes proposed bring aid not only to the formalization of joint doctrine terminology in the JDO – where adherence to rule 5 allows immediate generation of the backbone taxonomy of the ontology – but also to the quality assurance of joint doctrine definitions themselves, by allowing easier checking of logical consistency.

IX. Building the JDO

A. The OBO Foundry

Our strategy for building the JDO follows an approach to coordinated ontology development as a means to advancing interoperability across multiple domains that was first successfully applied in the life sciences in the context of the Open Biomedical Ontologies (OBO) Foundry initiative ([21]). The strategy rests on dividing the domain of biomedicine into a number of sub-domains (for genes, proteins, cells, and so forth) and creating ontology modules representing the corresponding general types of entities. Each ontology module consists of general terms organized hierarchically through the parent-child relation between types and subtypes. This relation then serves as the starting point for the formulation of the definitions of the terms in the hierarchy in accordance with Rule 5 above. This strategy is currently being applied in a series of DoD and intelligence community projects, in each case drawing on the Basic Formal Ontology (BFO) [12], which serves as a common upper level starting point for the creation of definitions of the terms used in the domain ontologies at lower levels.

The predominance of general terms in JP 1-02 reflects the purpose of military doctrine, which is to help warfighters understand the realities of war and their specific situations. It achieves these ends largely through the identification and explanation not of specific instances (such as a particular aircraft or IT system) but rather of important general categories. Doctrine is re-usable because it is applicable to many different instances and to many different sub-kinds of the same general categories that re-appear in ever new situations. This approach is effective because the basic realities of war are not changed by the fielding of new commanders, equipment, specialties, or tactics. A new IT system may provide a commander with more information in easier-to-understand formats; but the basic role of IT in supporting unified action remains unchanged. Because the developers of doctrine were so successful in identifying the high-level categories of C2, commanders and others continue to use these same categories when understanding how to employ each new IT system to create better operational capabilities.

The most general categories in military doctrine are:

(1) *thing* (people, equipment, organizations),

(2) *attribute* (capabilities, functions, roles, including relational attributes of command or support), and

(3) *process* (for example, the joint planning process).

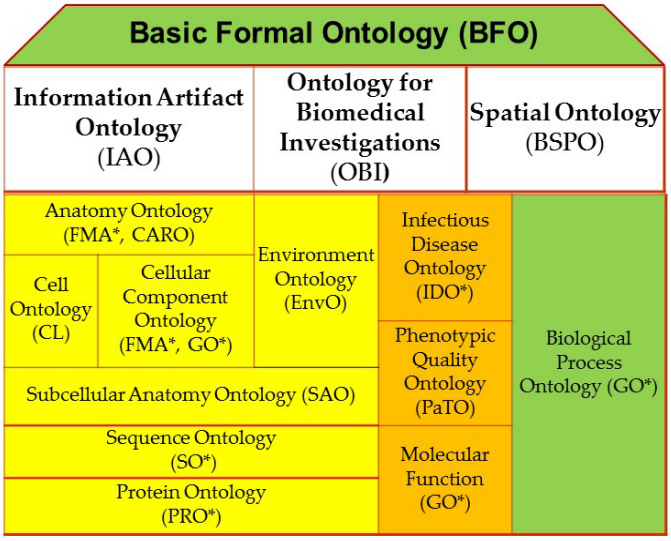


Figure 1: OBO Foundry strategy for modular coordination

Nowhere is it stated explicitly in military doctrine that these are the basic categories of the reality of war. Rather, the doctrinal publications are divided by area of warfare and by process (C2, intelligence, fire support, logistics, planning, and so forth). One of the virtues of joint doctrine is its consistent use of the same general terms representing sub-categories of *thing*, *attribute*, and *process* across all the joint publications. For example, every joint publication uses the term **commander** (*thing*) for the officer appointed to **command** (*process*) an **organization** (*thing*) and to exercise **authority** (*attribute*) over subordinates. It is impossible to understate the value of this achievement, which has not only diminished communications barriers among the warfighters of different specialties but also facilitated the application of IT in planning, training, and real world operations. What is remarkable is that the authors, managers, and terminologists of joint doctrine achieved this consistency with minimal documented theory and procedures for categorization and for the writing of definitions.

B. *BFO and the Common Core Ontologies*

In our view, BFO provides the documented theory needed to fill this gap [12]. BFO is architected around the same upper-level categories (of *thing*, *attribute*, and *process*) used by joint doctrine. More importantly, BFO serves as the starting point for a suite of associated resources – based on the Common Core Ontologies (CCO) (see [11] and Figure 2) – that have been purpose built to support IT applications in the military and intelligence domains.

The CCO and other domain-ontology modules are (1) defined in BFO terms and then (2) they are themselves extended through the addition of domain-specific sub-ontologies along the lines illustrated in Figure 2. The BFO community has refined and tested the needed theory and procedures for generating such sub-ontologies in agile fashion and for preserving their usability and mutual consistency across successive versions. [14]-[16].

C. *Building the JDO as Shadow JP 1-02*

Our strategy for building JDO is incremental. We proceed through the successive joint publications (JP n-m), moving from general to specific, for instance from JP 3-0 (*Joint Operations*) to JP 3-14 and JP 3-17 (*Space Operations* and *Air Mobility Operations*). The creation of an ontology for each JP n-m then follows three steps:

i. ENRICHMENT: create JP n-mE, a shadow version of those portions of JP 1-02 whose terms are defined in JP n-m but enriched (E) through the addition of new terms – for example, **commander** – that are not defined in JP 1-02 but used in JP n-m definitions;

ii. LOGICAL REGIMENTATION: create JP n-mLR, a logically regimented (LR) version of JP n-mE, in which definitions are formulated in humanly understandable English but with the logical regimentation sketched in our summary treatment of supplementary rules for definition writing in section VII, as supplemented by the further rules set forth in [12]);

iii. FORMALIZATION: create JP n-mF, in which the human-readable definitions in JP n-mLR are formalized (F) using the Web Ontology Language (OWL);

Content from CCO is incorporated in each stage as needed. Examples are provided at http://ncor.buffalo.edu/JDO-Oct-2015.

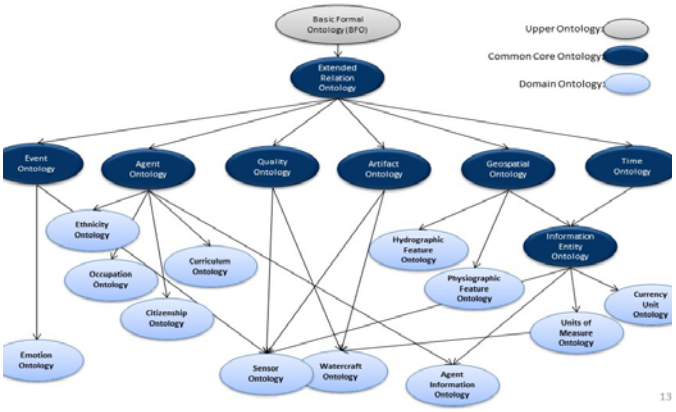


Figure 2: Common Core and associated domain ontologies

X. POTENTIAL BENEFITS OF THE JDO TO THE WARFIGHTER

We are developing the JDO to support efforts to extend the applicability of doctrine in those areas where commanders, planners, and other warfighters need to call upon information and information support in order to be effective. JDO will provide a computationally accessible counterpart of the content of JP 1-02 designed to support unified action by advancing terminological consistency and interoperability.

The major benefit of JDO should take the form of better C2 through improved communication, self-synchronization, and projection into the future, and in each stage of development of the JDO we will be testing its utility in supporting improvements along all of these dimensions.

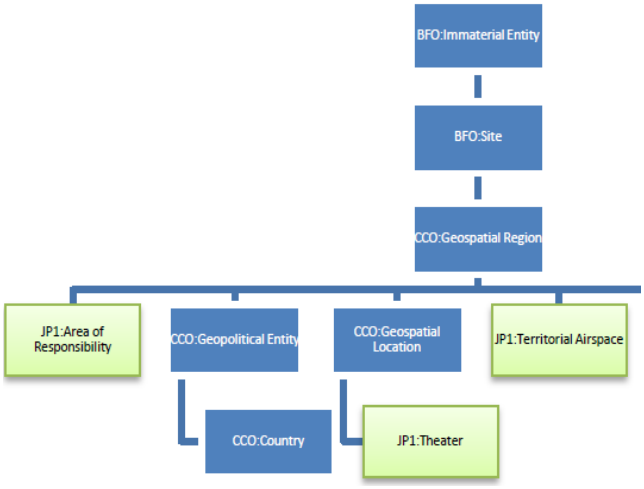


Figure 3: Examples of CCO and JP 1 terms descending from BFO in the JDO

As stated in JP 6-0 ("Joint Communications System") [13], a C2 system has two elements: (1) the people, who make decisions and accomplish missions, and (2) the facilities, equipment, communications, staff functions, and procedures essential to the commander for C2. People can conduct C2 without facilities, equipment and so forth, but the latter cannot perform C2 without people. Since unified action occurs not only between people and organizations, but also between IT systems and people, by advancing interoperability in the ways described above, a successfully developed JDO can facilitate moving past the low level of unity of action among people, organizations, and IT systems that has been achieved until now.

A subsidiary benefit takes the form of providing ways to extend the range of IT-supported uses of the content of doctrine, for example, by allowing the DoD Dictionary to serve as an entry point for web-based searches across multiple repositories of authoritative data; by facilitating greater coordination of training and operations; and by increasing automation of processes such as plan specification, course of action development, and operations and Blue Force Status assessment, particularly within highly contested environments.

We anticipate that the JDO will allow further enhancements of JP 1-02, for example, by providing for each term in the dictionary its own web page that can serve as a repository of usage and of revision history. This last benefit is part of our more general strategy to assist developers of the hundreds of IT systems that are developed for U.S. military operations to achieve the benefits of interoperability and to keep track of needed information.

Access to detailed information on the usage and revision history of the vocabularies of the intended users of these hundreds of systems would facilitate unified action among IT developers, for example, by helping to rectify the current situation in which even the best-intentioned and conscientious IT developers must make assumptions on whether a warfighting term in a specification that is listed in joint doctrine is intended to be defined by the current or by some previous definition in JP 1-02. Integration of the JDO within the larger BFO–CCO framework would also help to resolve some of the problems that arise when expressions are used in JP 1-02 definitions but are themselves not defined in JP 1-02.

XI. Potential Benefits of JDO to the Doctrine Staff

Anticipated benefits of JDO to *doctrine authors* include the ability to apply standard ontology editing and visualizing software, for example, to create visualizations of how different parts of doctrine interact, of the doctrinal content relevant to particular types of operations or capabilities, or of the ways doctrine is used (and not used) in specific plans and operations. These benefits include opportunities for logical tracking of dependences among terms and definitions to identify (direct and indirect) circularities and thereby to help to ensure, when changes in definitions are made in the process of revision, that the effects of these changes cascade appropriately through all dependent definitions.

For example, imagine that revisions need to be made to the definition of a term such as **base defense** illustrated in Figure 4. The Figure tells us which definitions then need to be checked for continued validity, by showing the terms in JP 1-02 that are defined using **base defense** either directly or indirectly (by inheritance from a definition lower down the corresponding chain).

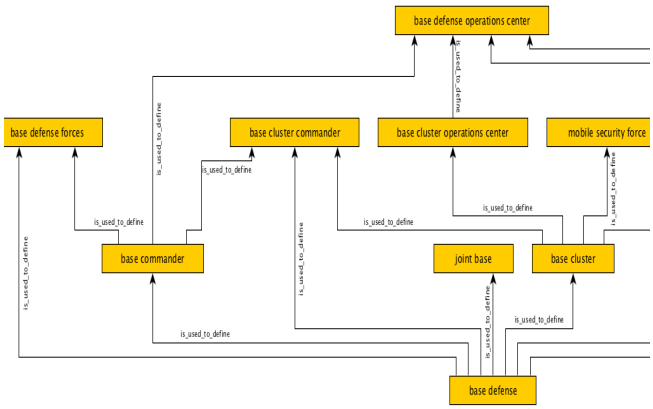


Figure 4: Fragment of the JP 1-02 network generated by the relation *is used to define.*

We believe that terminological interoperability can be achieved only where the terminologies involved are developed as part of, or are defined in terms derived from, a common benchmark ontology framework. Only such a framework can provide a basis for clearly formulated logical relations between terms, and only this will allow the sort of automated checking for consistency that is needed when the terminological content of multiple information systems is aggregated together in larger (actual or virtual) systems. This requirement for automated consistency checking becomes all the more urgent as terminological artifacts are revised over time. We believe that the value of the JDO will reveal itself not least in supporting consistent revision of JP 1-02 in tandem not merely with Service and coalition doctrines but also with information artifacts such as the Universal Joint Task List (UJTL), the Joint Lessons Learned Information System (JLLIS), and their Service counterparts.

Finally, JDO can also help the many teams of ontologists working on different military and intelligence community initiatives to advance information discovery and processing. The JDO will enable doctrine to serve as a new source of ground truth for ontologists across DoD and IC that can help to ensure mutual consistency and identify wasteful redundancies as well as gaps and errors in existing ontologies. It will contribute to consistent and yet agile development of IT technology while also counteracting current tendencies toward silo formation and failures of interoperation.

APPENDIX: EXAMPLES OF PRIOR WORK

The practical value of an ontology-based approach to supporting operational military IT has been demonstrated most conspicuously in the ICODES (Integrated Computerized Deployment System) load-planning system, a program of record employed by the DoD since 1997 [22].

A more recent example is the AFRL (Air Force Research Lab)/USTRANSCOM Mission Data and Transport Ontology project described in [23]. Here, the goal was to create a domain model of U.S. Transportation Command's operations, including operational processes, organizations, and Commander's Critical Information Requirements (CCIR), to be used to support the monitoring of information relevant to USTRANSCOM missions. Specifically, rules were used to modify terms and definitions of the Joint Mission Essential Task List (JMETL) in ways similar to those described in section VIII in relation to the definitions of **operational area** and **cyberspace operations**. When the resulting domain model was used with the Securboration MetaTagger application, there was a reduction 20% in the numbers of people required for monitoring for critical information and a reduction of 1–3 days in discovering such information.

Another AFRL effort used analogous rules in transforming a portion of the Joint Capability Areas (JCA) taxonomy into an ontology-based model that was then processed by a machine-learning algorithm to train an application. Formalizations of the JCA descriptions were used to allow comparisons of unstructured text against each of the formalized descriptions in order to determine matches. Initial attempts to disambiguate each of the JCA descriptions failed because of redundancy and ambiguity. Instead, a hybrid was created consisting of formalizations of JCA descriptions along with word bags of their respective contents. A machine learning algorithm was then used to compare historical user input against both to train the algorithm. Here, too, the implementation (described in [24]) shows a reduction in 1–3 days for discovering critical information that could affect USTRANSCOM operations, for example, in case of earthquake or other disaster and a 20% reduction in manpower required for monitoring the information.

ACKNOWLEDGEMENTS

Our thanks go to Lieutenant Colonel James McArthur (USMC JS J7), Lieutenant Colonel William D. Betts (USAF JS J7), Tatiana Malyuta (CUNY), Tony Stirtzinger (Securboration), Andreas Tolk (MITRE/Old Dominion University), and Erik Thomsen (Charles River Analytics).

REFERENCES

[1] Joint Publication (JP) 1, *Doctrine for the Armed Forces of the United States*, 25 March 2013.

[2] M. R. Hieb, S. Carey, M. Kleiner, M. Hieb, R. Brown, "Standardizing Battle Management Language – A Vital Move Towards the Army Transformation", *IEEE Fall Simulation Interoperability Workshop*, 2001.

[3] S. Lambert, M. R. Hieb, "Improving Unity of Effort in Command and Control Processes: An Operational Analysis of a Joint Doctrinal Language", *Spring Simulation Interoperability Workshop*, 2006, 743-752.

[4] Tolk, Andreas, and Curtis Blais. "Taxonomies, Ontologies, Battle Management Languages – Recommendations for the Coalition BML Study Group", Spring Simulation Interoperability Workshop 2005.

[5] Simulation Interoperability Standards Organization (SISO), Standard for: Coalition Battle Management Language (C-BML), Phase 1 SISO-STD-011-2012-DRAFT 4 April 2012.

[6] Department of Defense Instruction 8330.01, "Interoperability of Information Technology (IT), Including National Security Systems (NSS)", May 21, 2014.

[7] Joint Publication (JP) 1-02, *Department of Defense Dictionary of Military and Associated Terms*, as amended through 15 June 2015.

[8] Department of Defense Instruction Number 8320.02, "Sharing Data, Information, and Information Technology (IT) Services in the Department of Defense", August 5, 2013.

[9] Acquisition Community Connection (ACC) Practice Center, Terms and Definitions, Open Systems Architecture, https://acc.dau.mil/Community Browser.aspx?id=220108&lang=en-US, October 14, 2015.

[10] J.-F. Ethier, O. Dameron, V. Curcin et al., "A unified structural/terminological interoperability framework based on LexEVS", *Journal of the American Medical Informatics Association*, 2013, 20, 986-994.

[11] J. R. Schoening, et al., "PED fusion via enterprise ontology," *Proceedings of SPIE 9464,* Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR VI.

[12] R. Arp, B. Smith, A. D. Spear, *Building Ontologies with Basic Formal Ontology*, Cambridge, MA: The MIT Press, 2015.

[13] Joint Publication (JP) 6-0, *Joint Communications System*, June 2015.

[14] Basic Formal Ontology, http://ifomis.org/bfo, September 2015.

[15] D. Salmen, T. Malyuta, A. Hansen, S. Cronen, B. Smith, "Integration of Intelligence Data through Semantic Enhancement", *Semantic Technology in Intelligence, Defense and Security* (STIDS), 2011, CEUR 808, 6-13.

[16] B. Smith, T. Malyuta, W. S. Mandrick, C. Fu, K. Parent, M. Patel "Horizontal Integration of Warfighter Intelligence Data. A Shared Semantic Resource for the Intelligence Community", *Semantic Technology in Intelligence, Defense and Security* (STIDS), 2012, CEUR 996, 112-119.

[17] Chairman of the Joint Chiefs of Staff Manual (CJCSM) 5120.01, "Joint Doctrine Development Process," December 29, 2014

[18] Department of Defense Instruction 5025.12, "Standardization of Military and Associated Terminology," August 14, 2009.

[19] Chairman of the Joint Chiefs of Staff Instruction 5705.01D "Standardization of Military and Associated Terminology," Nov. 10, 2010.

[20] S. Seppälä. "An ontological framework for modeling the contents of definitions", *Terminology*, 21(1):23–50, 2015.

[21] B. Smith, et al., "OBO Foundry: Coordinated Evolution of Ontologies to Support Biomedical Data Integration", *Nature Biotechnology*, 25 (11), 1251-1255.

[22] K. Pohl and P. Morosoff, "ICODES: A Load-Planning System that Demonstrates the Value of Ontologies in the Realm of Logistical Command and Control (C2)", *InterSymp-2011*, 2011.

[23] J. M. Powers, Keith D. Shapiro, and David S. Monk, "Information Exchange and Fusion in a Collaborative Environment using Semantic Information Requirements", *International Conference on Collaboration Technologies and Systems* (CTS 2014). 597-601.

[24] Securboration, "Human assisted analysis for change alignment to an Enterprise Architecture", Requirements Analysis Portlet (RAP). May 2012.

Automated Ontology Creation using XML Schema Elements

Samuel Suhas Singapogu, Paulo C. G. Costa, J. Mark Pullen
C4I center, George Mason University
Fairfax, VA, USA
[ssingapo,pcosta,mpullen]@c4i.gmu.edu

***Abstract*—Ontologies are commonly used to represent formal semantics in a computer system, usually capturing them in the form of concepts, relationships and axioms. Axioms convey asserted knowledge and support inferring new knowledge through logical reasoning. For complex systems, the process of creating ontologies manually can be tedious and error-prone. Many automated methods of knowledge discovery are based on mining domain text corpus, but current state-of-the-art methods using this approach fail to consider properly semantic data embedded in XML schemata in complex systems. This paper proposes a mapping method for identifying relevant semantic data in XML schemata, automatically structuring and representing it in the form of a draft ontology. Concepts, concept hierarchy and domain relationships from XML schema are mapped to relevant parts of an OWL ontology. A part-of-speech tagging method extracts domain relationships from schema annotations. This mapping method can be applied to any system that has a well-annotated XML schema. We illustrate our process with the preliminary results obtained when creating a command and control to simulation (C2SIM) draft ontology from an XML schema.**

Keywords—OWL, XML Schema, Part of Speech tagging, Command and Control, Interoperability

I. INTRODUCTION

Knowledge discovery is essentially the process of extracting semantic concepts and relationships from domain resources within a particular domain. Ontologies are the *de facto* standard for representing knowledge of a system [1]. The framework and components of ontologies are based on established, yet evolving W3C standards. Ontologies usually are comprised of concepts, relationships and axioms. Concepts are abstractions of related attributes that form the basic building blocks of a semantic model. Relationships can be between two concepts or between a concept and a data-type. Axioms consist of asserted knowledge that can be represented as a `<subject, predicate, object>` triple. Logic-based reasoners can be used to infer new knowledge in an ontology.

Although ontologies are valuable assets in system modeling, testing and analysis, the process of manually creating an ontology for a complex system is inherently tedious and error prone. Existing methods of knowledge discovery and ontology creation usually are based on text mining of a data corpus for that domain. XML-based systems capture the structure and syntax of all necessary and meaningful elements in a XML schema, which therefore is a useful starting point for semantic analysis. In such systems, XML schemata have been shown to be a valuable resource of semantic data [2]. Command and Control systems in the military context support various functions, including commanding of forces and also receiving and interpretation of situational awareness reports. To perform these functions, most C2 systems are modeled using XML schemata e.g. Coalition Battle Management System (C-BML) [3], Military Scenario Definition Language (MSDL) [4], and National Information Exchange Model (NIEM) [5], which represent the systems' structural and syntactic framework. These systems use and exchange XML documents that are based on XML schemata.

XML often is used as the exchange mechanism in the command and control domain [6]. Given the increasing number of XML-based systems in the C2 domain, an automated framework that leverages semantic information in the XML schema and creates a draft ontology would be a useful tool. Domain experts can refine and populate the draft ontology using concept hierarchy and basic domain relationships. For large XML based systems, this process of creating a draft ontology saves valuable time and avoids errors common to the alternative tedious, manual process. In contrast to existing techniques used to map a XML schema to an ontology, the mapping proposed in this paper highlights the need to map schema element (`xs:element` from here forward) to an ontological concept. This avoids the usual approach of mapping XML complex types to concepts that could lead to unnecessary ontological complexity. In addition, this paper proposes a novel Part of Speech tagging

method to extract domain relationships from well-annotated XML schema.

II. Related Work

Most existing work on mapping a XML Schema to an ontology (e.g. [7], [8]) is based on mapping XML schema complexType (henceforth referred to as `xs:complexType`) to an `owl:class`. This mapping can lead to problems for the following reasons:

1. A "simpleType" definition is sufficient to define a semantic concept. In the command and control domain, for example, it is possible to define an element called "UnitName," which is of a "simpleType" string (with string restrictions). There is sufficient semantic information in this definition to create a distinct ontological concept. When this level of abstraction of concepts is ignored and only complex type definitions are considered as semantic concepts, the resulting ontologies will contain significant modeling gaps for any useful analysis.
2. By design, XML parsing only allows elements associated to a complex type to appear in valid XML files. XML schemata can contain complex type definitions that are never associated to an element definition. When concepts are mapped to complex types, the resulting ontology will likely include concepts that will never appear in the XML document. This unnecessary complexity is counterproductive to efficient semantic modeling in design and analysis.

Bohring et al. [9] recognize the value of semantic concepts being mapped to `<xs:element>` definition. However, this mapping is done only for `<xs:element>` definitions that are not leaf nodes and have at least one attribute definition. The approach in [9] fails to consider valid semantic concepts that are simple literal definitions. In addition, ontologies have been designed so that datatype properties can be mapped to XML schema datatypes [10]. Therefore, the resulting mapping of simple `xs:element` to `owl:DatatypeProperty` using that approach would be inconsistent with standard practice of ontology design. Yang, Steele, and Lo [11] describe an ontology-based mapping between XML and ontology (bi-directional) that focuses on limiting loss of information in the bi-directional mapping.

Existing work ignores semantic information pertaining to domain relationships that are present in well-annotated XML schema annotations. By design, the purpose of XML annotations is to capture description of elements, which are often described in relation to other elements. For instance, consider the XSD annotation for the element `<xs:EventStatus>` of the C2 domain in Figure 1.

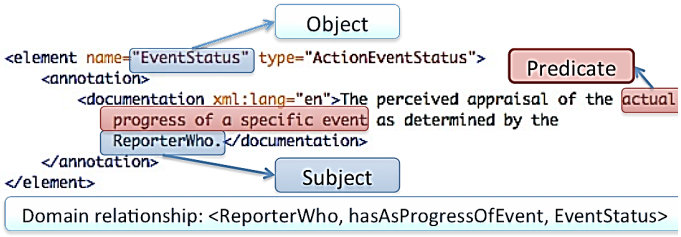


Fig. 1. Illustrating the presence of domain relationships in XSD annotations

This pattern of linking elements in annotations is common in domains with well-annotated XML schemata. Our approach leverages this pattern by employing a mapping from `xs:element` to `owl:class` and uses Part of Speech tagging of XSD annotations to extract domain relationships.

III. System Design

The mapping process takes as input a sufficiently annotated XML schema, which we define as any XML schema that contains the following:

a) The schema provides annotations for most elements using descriptive domain terminology
b) Annotations referencing elements defined in the schema use a consistent naming convention.

As an example of the latter, if the schema defines an element as "ReporterWho" then any annotation referring to this element must do so in a consistent way, i.e., the reference can be extracted by simple operations (e.g., removing spaces, pruning special characters, etc.)

The system components and relationships between the components are illustrated in Figure 2 below.

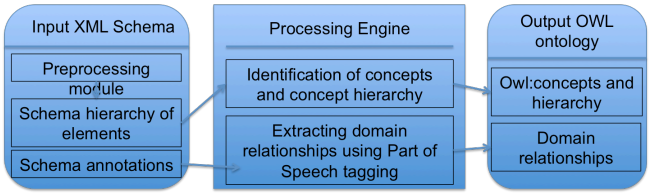


Fig. 2. Illustrating system components and relationships

Pre-processing the XML schema: Our approach maps the ontological concept name to the name of the element in the XML schema. XML schemata allow for multiple elements to have the same name. More often than not, the same name is used (e.g. `<xs:element name="ID"/>`), specially when defining identifiers and other common elements. Even though the same name may exist in different element definitions, as long as their context is different they are semantically different concepts. In order to disambiguate between elements with the same name, we propose a pre-processing step that concatenates the parent complex type name to the name of the element using a delimiter. For elements that do not have a parent complex type, an iterating place-holder name can be used (e.g. "Parent$_i$") Because XML schema rules require that all `xs:complexType` have unique names in the schema, this pre-processing step ensures that element names (and therefore concept names in the ontology) are unique in the XML schema. The pre-processing step performs the disambiguation technique to all elements, and not only to the redundant elements. This is specifically convenient because capturing the schema structure in the ontology could be useful to other ontological processes (e.g. ontology matching that uses XML schema structure). The pre-processing step is illustrated in Figures 3 and 4 below.

```
<complexType name="GDCLightType">
    <annotation>
        <documentation> Provides the coordinates for a point using the GDC system.
        </documentation>
    </annotation>
    <sequence>
        <element minOccurs="0" name="OID" type="OIDType"/>
```

Fig. 3. XML schema before pre-processing

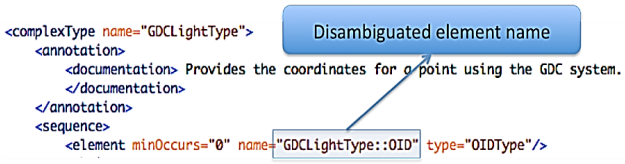


Fig. 4. XML schema after pre-processing

After pre-processing, the following mappings are established while parsing through the XML schema:

Mapping 1: `'xs:element'` to `owl:class`: Commonly, each definition of an element contains a name and an associated complex type. An `owl:class` is created with the class name equal to the name of the element. Attribute definitions for the element are mapped to the `owl:datatype` property. Cardinality of concepts is defined according to the `xs:minOccurs` and `xs:MaxOccurs` in the XML schema as explained in [9]. Therefore, if an element "element1" is defined as having type "complexType1" and the definition of "complexType1" includes an "element2" with `max-occurs=unbounded`, then the cardinality between the concepts "element1" and "element2" is 1…∝

Mapping 2: Element hierarchy to concept hierarchy: Nayak and Wina [12] have noted that XML schema contains element definitions in a hierarchical structure. They employ structural information in XML schemata to define clusters based on semantic similarity. Varlamis and Michalis [13] note that XML schema relationships support inheritance relationships between elements. The most common method to create an inheritance relationship in a XML schema is to use `xs:extension` so that one element can extend another. This mapping step also identifies all occurences of "`abstract=true`" in the definition of an element, computing it as indication of a inheritance relationship. It should be noted that existing mapping methods ignore the presence of inheritance using of "`abstract=true`" because `xs:element` is not mapped to ontological concept.

Mapping 3: Schemata composition to 'partOf' and 'kindOf' relationships: In XML schemata an element defined as a complex type is composed of other elements. Elements can be composed using '`All`', '`Sequence`' and '`Choice`' indicators. All elements in '`All`' and '`Sequence`' groupings are mapped to a '`isPartOf`' OWL object property and all elements in the '`Choice`' composition are mapped to a '`isKindOf`' OWL object property. The following two examples illustrate the mapping from schema composition to OWL object properties.

Example 1 - Consider the definition of a Task below:

```
<xs:element name="Task" type="taskType">
```

The definition of 'taskType' is composed of other elements as follows:

```
<xs:complexType name="taskType">
 <xs:sequence>
   <xs:element  name="Who"
                type="whoType"/>
  <xs:element   name="When"
                type="whenType"/>
  <xs:element   name="Where"
                type="whereType"/>
 </xs:sequence>
</xs:complexType>
```

In the definition above, the element 'Task' is composed of 'Who', 'When' and 'Where' using the sequence composition. The mapping proposed in the paper will leverage the sequence composition to establish the following properties:

- 'Who' isPartOf 'Task'
- 'When' isPartOf 'Task'
- 'Where' isPartOf 'Task'

Example 2 - Consider the definition of a Where below:

```
<xs:element name="Where"
type="whereType>
<xs:complexType name="whereType">
 <xs:choice>
   <xs:element  name="AtWhere"
                type="AtWhereType"/>
  <xs:element   name="RouteWhere"
                type="routeType"/>
 </xs:choice>
</xs:complexType>
```

The element 'Where' is composed of 'AtWhere' and 'RouteWhere' using the choice composition. The mapping proposed in this paper will leverage the choice composition to establish the following properties:

- 'AtWhere' `isaKindOf` 'Where'
- 'RouteWhere' `isaKindOf` 'Where'

Mapping 4: Mining XSD annotations for domain relationships: Annotations in XML schemata are designed to provide documentation in the form of free text for elements being defined. It is common in the C2 domain to have annotations in XML schemata describing an element often in relationship to other elements. Existing published research in XML schemata to ontology mapping does not check for semantic relationships in XSD annotations. We propose the novel use of Part of Speech (POS) tagging to extract domain relationships from XSD annotations. Part of Speech tagging is a well-developed natural language technique that parses text and determines the part of speech for each word in the text. The common process is to determine the tag based on a probabilistic modeling of the word and its context (preceding and succeeding words). The current standard involves use of the Penn Treebank tokenization that categorizes into thirty-six possible parts of speech [14]. Extensive work has been done to identify how POS tagging can be used to determine relationships embedded in text, such as those described in [15][16][17][18]. Wang, Ting, et al. [19] use a support vector method and accompanying relationship ontology to determine semantic relationships embedded in text. The following steps are used to map XSD annotations to domain relationships (`owl:objectProperty`):

Step1: Identifying all concepts in the annotation: This is done by identifying all words tagged as nouns or proper nouns (NN, NNS, NNP, NNPS) by the POS tagger. The concepts are added to a vector as follows:

$$V_{concepts} = \{C_i \mid C_i \text{ is the } i^{th} \text{ concept in the annotation}\}$$

Step 2: Identifying predicates for the relationship: Starting at the beginning of the annotation, a concatenation of adjectives (JJ), Pronouns (WP), and prepositions (IN) is created until concept C_i is encountered. This concatenation forms the predicate of the domain relationship. These predicates are added into the vector as:

$$V_{predicates} = \{pred_i \mid pred_i \text{ is the concatenated predicate before the } i^{th} \text{ concept in the annotation}\}$$

Each concept C_i will now have an accompanying predicate. If $pred_i$ has only one word and is a coordinating conjunction (e.g. "and") then $pred_{i-1}$ is assigned to $pred_i$. This is due to the presence of a coordinating conjunction between

two concept names C1 and C2, meaning that whatever predicate applied to C1 also applies to C2.

Step 3: Creating the domain relationship: If the annotation is for element E1, then for each concept in $V_{concepts}$ the following domain relationship (`owl:objectProperty`) is created:

```
<subject, predicate, object> =
< Ci, str_concat(hasAs,predi), E1  >
```

```
<element name="TaskWhatRef" type="TaskWhatRefType">
    <annotation>
        <documentation xml:lang="en">Specifies a reference to task.</documentation>
    </annotation>
</element>
```

Fig 5. An example schema element with annotation

Based on the technique described above, the following domain relationship is created:

```
<Task hasAsReferenceTask TaskWhatRef>
```

Note: For the sake of clarity of illustration the schema as it appears before pre-processing is shown in Figure 5.

The mappings described in the steps above are illustrated in Figure 6.

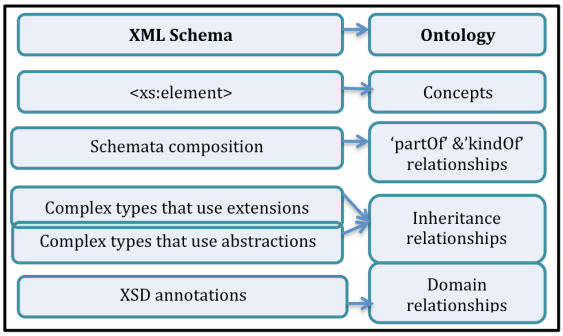


Fig. 6. Mapping from XML schema components to ontology components

IV. Application to C2SIM.

C2-to-simulation interoperability (C2SIM) is a standard under development by the Simulation Interoperability Standards Organization (SISO) to facilitate interoperability between C2 and simulation systems [20]. The current phase of standardization effort involves work on developing a formal semantic model as part of a model-driven framework. The goal is to use C2SIM to interoperate between multiple C2 and simulation systems.

In order to provide interoperability on the semantic level, C2SIM will require ontological support for formalizing semantics and for the design and analysis of networked C2 and simulation systems. Early work on the need for and future of semantic C2SIM is described in [21]. C2SIM development is based on complex XML schemata that have been developed in Phase 1 standardization effort of C-BML [3] and MSDL [4]. These XML schemata have been found to be complex [22] because the design intended to capture the full expressivity of the underlying sophisticated data model. Adopting a manual process to create an ontology from these schemata can be tedious and error-prone. The method proposed in Section 3 has been applied to the C-BML Phase1 schema [3]. Statistics of the XML schema used to create the draft ontology are presented in Table 1.

Table 1. Statistics regarding C2SIM XML schema

Metric	Value
Number of complex type definitions	531
Number of element definitions	1115
Number of annotations	1604
Number of unbounded elements(Number of elements that have "maxoccurs=unbounded")	40
Fanning Index [23](Number of relationships/number of elements)	9.67
ComplexityMeasure (based on the formula in [23]):	738

V. Preliminary Results and Discussion

A software prototype was built, based on the proposed method using OWL-API [24] to create the draft ontology.

The pre-processing step, described in section 2, disambiguated element names so that concepts can be accurately mapped to elements in the XML schema. The draft ontology created by this method captures a conceptual hierarchy consistent with an intuitive understanding of C2SIM. The domain relationships are descriptive and useful for capturing business rules. The statistics of the ontology created are shown in Table 2. At the time of this writing, we are conducting an evaluation of the draft ontology to validate these preliminary results. The evaluation involves the use of subject matter experts (SMEs) to evaluate the draft ontology by checking it against domain documents and their own expertise. The initial results, while still anecdotal, suggest that the resulting ontology is consistent with SME evaluation of domain documents.

TABLE 2. C2SIM ONTOLOGY METRICS

Ontology metric	Value
Number of Concepts	1115
Number of Inheritance relationships	765
Number of domain relationships	73

Figures 7, 8 and 9 provide snapshots of the ontology as viewed in the ontology editor Protégé 4.3 [25].

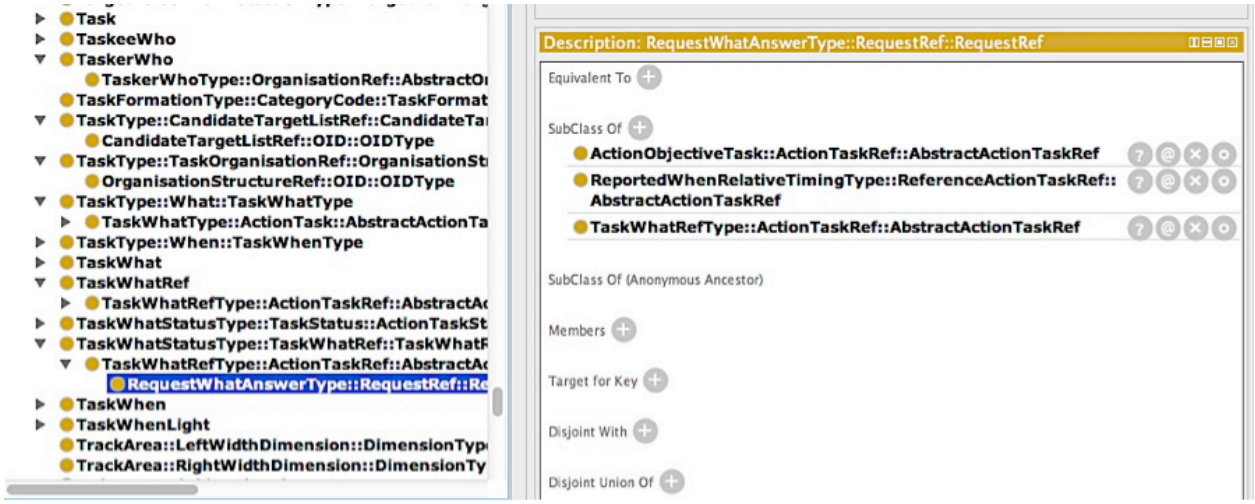


Fig. 7. Snapshot of class hierarchy in C2SIM ontology

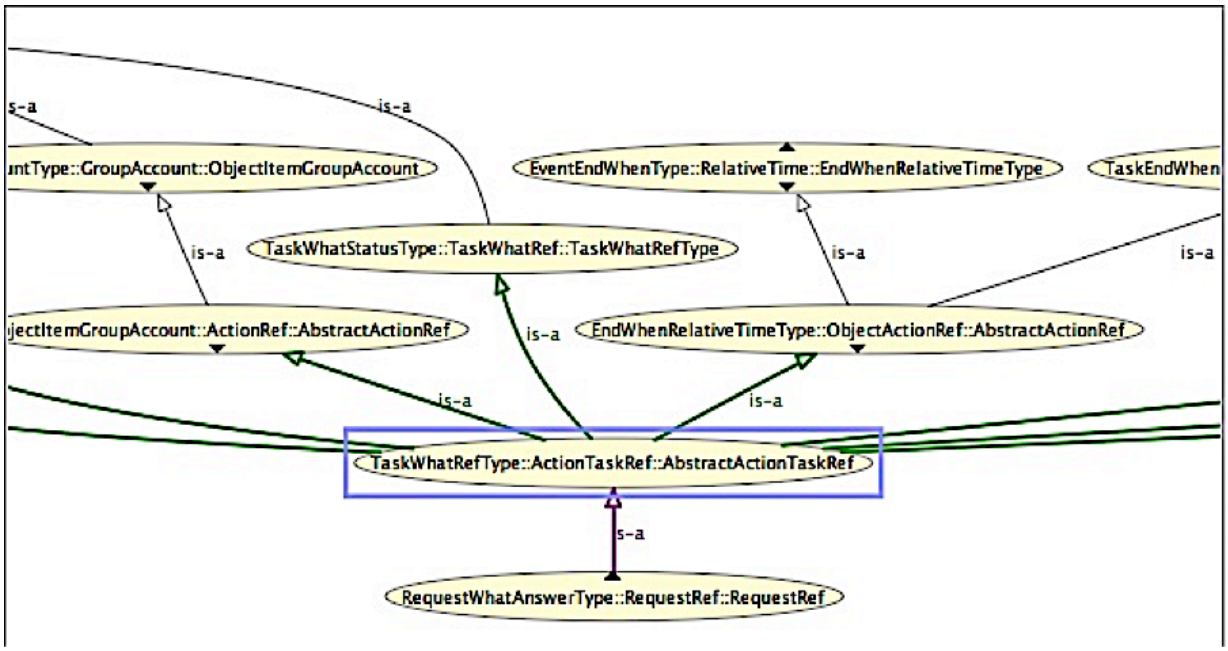


Fig. 8. Snapshot of C2SIM ontology subset visualized in OWL Viz.

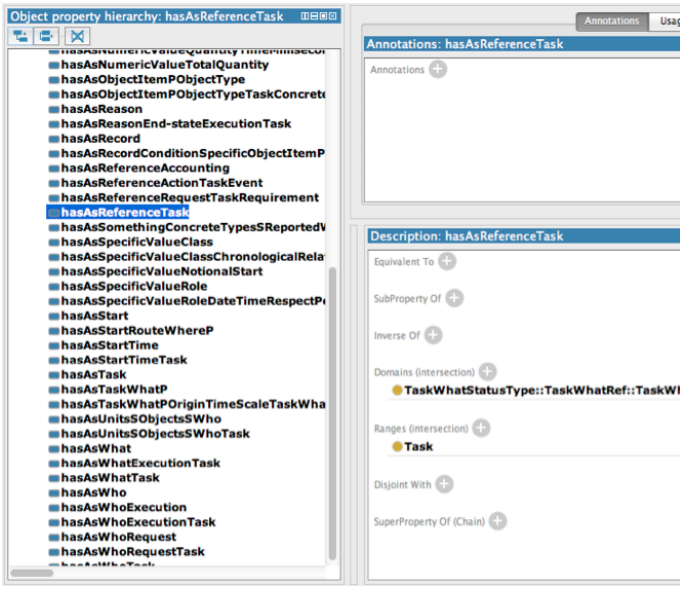


Fig. 9. Snapshot of domain relationships in C2SIM ontology extracted through POS tagging of XML schema annotations

VI. Future Work

The proposed method extracts domain concepts and relationships from well-annotated XML schema. Potential approaches to improve the quality and resolution of the resulting ontology include the use of domain synonym tables as a means to support identifying concept names in schema annotations. We believe this can account for variants of a name that may be used in the schema annotation. Future work improving our methodology also includes capturing C2 doctrine in the form of axioms, as well as evaluating the use of reasoning to both infer and hypothesize knowledge. In ongoing work [26], we are also investigating the use of structural information in XML schemata to perform ontology matching between two XML based ontologies. Ontology matching of two complex ontologies that have accompanying complex schemata suffers from high computational cost. Finally, we are exploring a tighter coupling between ontology creation and ontology matching by embedding basic XML schema structure in auxiliary ontology artifacts (e.g. annotations).

REFERENCES

[1] Adda, Mehdi. "A Pattern Language for Knowledge Discovery in a Semantic Web context." *Models for Capitalizing on Web Engineering Advancements: Trends and Discoveries: Trends and Discoveries* (2012): 59.

[2] Li, Luochen, et al. "Discovering semantics from data-centric XML." Database and Expert Systems Applications. Springer Berlin Heidelberg, 2013.

[3] Blais, C., Brown, D., Chartrand, S., Diallo, S., Heffner, K., Levine, S., Singapogu, S., St-Onge, M., and Scolaro, D.: "Coalition Battle Management Language (C-BML) Phase 1 Information Exchange Content and Structure Specification," Paper 10S-SIW-002, Proceedings of the Spring Simulation Interoperability Workshop, Simulation Interoperability Standards Organization, Orlando, FL, April 2010.

[4] Simulation Interoperability Standards Organization. 2008. Standard for: Military Scenario Definition Language. SISO-STD-007-2008, 14 October.

[5] https://www.niem.gov/ (last viewed: September 10 2015)

[6] Beardsworth, Robert, et al. "XML standards as the basis for data interoperability among military C2 systems and beyond." MILITARY COMMUNICATIONS CONFERENCE, 2010-MILCOM 2010. IEEE, 2010.

[7] Matthias Ferdinand, Christian Zirpins, and D. Trastour. Lifting XML Schema to OWL. In Nora Koch, Piero Fraternali, and Martin Wirsing, editors, Web Engineering - 4th International Conference, ICWE 2004, Munich, Germany, July 26-30, 2004, Proceedings, pages 354–358. Springer Heidelberg, 2004.

[8] Bedini, I., Gardarin, G., and Nguyen, B. Deriving Ontologies from XML Schema. In Proceedings 4émes Journées francophones sur les Entrepôts de Données et l'Analyse en ligne (EDA 2008). Invited paper. 5 - 6, June 2008 Toulouse, France

[9] Bohring, Hannes, and Sören Auer. "Mapping XML to OWL Ontologies." Leipziger Informatik-Tage 72 (2005): 147-156.

[10] http://www.w3.org/TR/2004/REC-owl-guide-20040210/#Datatypes1 (last viewed: September 10, 2015)

[11] K. Yang, R. Steele, A. Lo, '"An ontology for XML schema ontology mapping representation"' *in Proceedings of the 9th International Conference on Information Integration and Web-based Applications & Services (iiWAS 2007)* (2007)

[12] Nayak, Richi, and Wina Iryadi. "XML schema clustering with semantic and hierarchical similarity measures." Knowledge-Based Systems 20.4 (2007): 336-349.

[13] Varlamis, Iraklis, and Michalis Vazirgiannis. "Bridging XML-schema and relational databases: a system for generating and manipulating relational databases using valid XML documents." Proceedings of the 2001 ACM Symposium on Document engineering. ACM, 2001.

[14] http://www.comp.leeds.ac.uk/ccalas/tagsets/upenn.html (last viewed: September 10 2015)

[15] Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In Proceedings of HLT-NAACL 2003, pp. 252-259.

[16] Ruiz-Casado, Maria, Enrique Alfonseca, and Pablo Castells. "Automatising the learning of lexical patterns: An application to the enrichment of wordnet by extracting semantic relationships from wikipedia." Data & Knowledge Engineering 61.3 (2007): 484-499.

[17] Medelyan, Olena, et al. "Mining meaning from Wikipedia." International Journal of Human-Computer Studies 67.9 (2009): 716-754.

[18] Sánchez, David. "A methodology to learn ontological attributes from the Web."Data & Knowledge Engineering 69.6 (2010): 573-597.

[19] Wang, Ting, et al. "Automatic extraction of hierarchical relations from text." The Semantic Web: Research and Applications. Springer Berlin Heidelberg, 2006. 215-229.

[20] Pullen, J. Khimeche, L. "Advances in Systems and Technologies towards Interoperating Operational Military C2 and Simulation Systems", 19th International Command and Control Research and Technology Symposium (ICCRTS). Alexandria, VA, 2014.

[21] Singapogu, Samuel. "Opportunities for Next Generation BML: Semantic C-BML". 19th International Command and Control Research and Technology Symposium (ICCRTS). Alexandria, VA, 2014.

[22] Abbott, Jeff, J. Pullen, and Stan Levine. "Answering the Question: Why a BML Standard Has Taken So Long to Be Established?." *IEEE Fall Simulation Interoperability Workshop, Orlando FL*. 2011.

[23] McDowell, Andrew, Chris Schmidt, and Kwok-bun Yue. "Analysis and Metrics of XML Schema." Software Engineering Research and Practice. 2004.

[24] http://owlapi.sourceforge.net/ (last viewed: September 10 2015)

[25] http://protege.stanford.edu/ (last viewed: October 25 2015)

[26] Singapogu, Samuel Suhas. "Ontology Matching Using Structure and Annotations in XML Schema". 20th International Command and Control Research and Technology Symposium (ICCRTS), 2015.

A Probabilistic Ontology for Large-Scale IP Geolocation

Kathryn Blackmond Laskey
Department of Systems Engineering and Operations Research
George Mason University
Fairfax, VA 22030
Email: klaskey@gmu.edu

Sudhanshu Chandekar and Bernd-Peter Paris
Department of Electrical and Computer Engineering
George Mason University
Fairfax VA 22030
Email: [schandek, pparis]@gmu.edu

***Abstract*—Mapping IP addresses to physical locations is important for a host of cyber security applications. Examples include identifying the origin of cyber attacks, protecting against fraud in internet commerce, screening emails for phishing, and enforcing restrictions on commerce with sanctioned countries. Simultaneous geolocation of large numbers of IP hosts is needed for cyber situation awareness. Explicit formal representation of the geospatial aspects of the cyber domain is necessary for interoperation with other cyber security capabilities. Formally representing the uncertainty inherent in geolocation supports increased accuracy via information fusion, as well as integration of geospatial inference with inference about other aspects of the cyber landscape. This paper presents a probabilistic ontology (PO) for IP geolocation. The geolocation PO is represented in the PR-OWL language, which allows an OWL ontology to be augmented with information to support uncertainty management. We show how the PR-OWL ontology supports automated construction of a Bayesian network for simultaneously geolocating a large number of IP hosts. The ultimate aim is to integrate our probabilistic ontology into a comprehensive cyber security probabilistic ontology to support cyber situation awareness, predictive modeling, and response strategy definition.**

I. Introduction

Recognition is growing of the need to establish a common vocabulary for and a shared understanding of the cyber security domain (e.g., [1]). Explicit, formal representation of entity types, properties and relationships is a key means to this end ([2], [3], [4]). Among the advantages of such a cyber domain ontology include increasing interoperability of cyber security tools and methods, improving tools to support situation awareness among cyber security operators, and enhancing information sharing among domain experts (c.f., [5]). Anticipating, diagnosing and responding to increasingly sophisticated cyber threats requires drawing on and fusing information from diverse sources. Automated fusion of hard and soft, structured and unstructured information requires semantic as well as syntactic interoperability among information providers and consumers. Ontologies are a key enabler of semantic interoperability.

The cyber security domain is fraught with uncertainty. Support for uncertainty management is a key requirement for cyber situation awareness and decision support tools. Probabilistic ontologies augment traditional ontologies with the ability to represent uncertainty associated with properties of and relationships among domain entities, supporting semantically aware automated uncertainty management [6].

This paper presents a case study of the use of a probabilistic ontology to represent and reason about a key problem in the cyber security domain, mapping IP addresses to physical locations. Example applications of IP geolocation include identifying the origin of cyber attacks, protecting against fraud in internet commerce, screening emails for phishing, and enforcing restrictions on commerce with sanctioned countries. Most IP geolocation methods focus on identifying the location of a single IP host. To support cyber situation awareness, a useful capability is simultaneous geolocation of a large number of hosts, with a reduced requirement for geographic resolution.

As an essential component of an overall cyber security strategy, geolocation services need to interoperate smoothly with other elements of a cyber security tooklit. For this purpose, an ontology of the geospatial aspects of the cyber domain can form a useful module in a cyber domain ontology. Available information for IP geolocation is fraught with uncertainty. Representing the uncertainty inherent in geolocation can support more accurate geolocation through information fusion, as well as integration of geospatial inference with inference about other aspects of the cyber landscape.

This paper presents a probabilistic ontology (PO) for IP geolocation and describes its application to the simultaneously IP node geolocation problem. The geolocation PO is represented in the PR-OWL language, which provides constructs for augmenting an OWL ontology with information to support uncertainty management. We show how the PR-OWL ontology supports automated construction of a Bayesian network for simultaneously geolocating a large number of IP hosts. The ultimate aim is to integrate our probabilistic ontology into a comprehensive cyber security probabilistic ontology to support cyber situation awareness, predictive modeling, and response strategy definition.

The paper is organized as follows. Section II gives a brief overview of previous research on IP node geolocation. Section III presents a factor graph model [7] for simultaneous IP node geolocalization that forms the basis for our PO. Section IV makes the case for explicitly representing the semantics of the model, presents a probabilistic ontology for IP node localization, and shows how the probabilistic ontology can be

queried using a generic reasoner to construct a geolocation model that is formally equivalent to the model of [7]. Section V presents a summary and discussion.

II. Background: IP Geolocation

Although most work on IP geolocation focuses on identifying the physical location of a single IP host, the problem of simultaneous geolocation of many IP hosts is beginning to receive attention ([7], [8]). The challenge for large-scale IP geolocation is three-fold. The first is to expand the scope of IP geolocation by taking into account not only hosts at the network edge but also hosts in the network core, such as routers. The second is to achieve scalability to large numbers of IP hosts, which requires the ability to simultaneously infer the location of many hosts. The final challenge is to improve geolocation accuracy while not sacrificing the first two objectives. To tackle these challenges, we introduce a model that uses Bayesian inference to fuse information from multiple sources to simultaneously geolocate a set of IP addresses to within a discrete set of geographic regions. The simultaneous geolocation model underlying our IP geolocation PO was presented in [7], and is reviewed briefly in this section.

By itself, an IP address provides no information about a host's location. Therefore, information from external sources is required to map an IP address to a physical location. Available information comes from different sources, each subject to uncertainty. Information sources for geolocation can be classified into three broad categories: database-based, name-based and measurement-based [9].

Geolocation databases [10] contain mappings between IP addresses and locations. These providers tend to focus on geolocating end-hosts, and consequently tend to be unreliable in geolocating routers. Moreover, it has been observed that databases tend to geolocate blocks of IP addresses to the location where they were initially registered — often the business address of the network provider. As a result, some geographically distributed blocks of IP addresses that may be geolocated to the same location. Location information about devices in the core of the network, such as routers, can often be inferred from the names assigned to them.

Name-based geolocation [11] uses information embedded in a hostname, such as an airport code or a city abbreviation, to infer the location of the host. For example, the hostname `ip68-100-3-241.dc.dc.cox.net` indicates a device located in Washington D.C. When available, hostname information tends to be fairly reliable, but it is not always available.

Measurement-based geolocation [12] uses network information such as delay and topology to estimate the location of nodes. When location of and connectivity to "landmark" hosts is available, measurement data can be used to infer the location of other nodes in the network. However, such techniques depend not only upon active landmarks that conduct delay measurements among themselves and the target but also on passive landmarks that are used for approximating the target's location. Also, due to factors discussed later, some delay measurements may be biased significantly. As a result, delay-based geolocation errors may be large, sometimes on the order of several hundred kilometers. The size of the error has been shown [13] to be correlated with the number of distributed landmarks and with the number of probes between landmarks and the unknown target. The dependence on many distributed hosts with known locations, coupled with the focus on pinpointing the location of individual target hosts, renders such techniques impractical for geolocating large numbers of IP hosts.

III. A Model for IP Geolocation

From the discussion above, it is clear that geolocating a large number of hosts requires coping with missing and/or imperfect information. Fusing information from the geolocation database and the location hints obtained from hostnames admit information about mutually exclusive sets of hosts, thereby increasing the number of hosts that can be geolocated. However, this does not guarantee improvement in accuracy due to the aforementioned uncertainties in the respective information sources. Because past studies [14], [15] have shown a strong relationship between measured delay and physical distance, we incorporate evidence about link delay into our model as a means to improve accuracy.

Our link delay measurements are taken from the DIMES database [16], [17], constructed as part of an ongoing, distributed, open-source project to map the structure and topology of the Internet. The DIMES database contains a set of `traceroute` measurements. Each `traceroute` measurement includes the measured round trip time (RTT) from a source node along a path toward a destination node, along with the IP addresses of the intermediate nodes along the path. Host-to-host or link delays can be inferred by subtracting RTTs for consecutive hosts. In addition, path information can be used to infer the network topology.

Our model assumes that IP nodes are to be geolocated into a set of M discrete disjoint geographic regions. A joint probability distribution is defined over the random variables defined in Table I. The random variable R_n denotes the region in which IP node n is located; G_n represents the result of a geolocation database query for IP node n; H_n represents the result of a hostname lookup for IP node n. The random variable Y_{mn} represents a measurement of relative host-to-host propagation delay between host m and host n.

TABLE I
Random Variable Definitions

Random Variable	Definition
R_n	Region in which node n is located
G_n	Region returned by geolocation database query for node n
H_n	Region returned by hostname lookup for node n
Y_{mn}	Delay measurement for signal transit between nodes m and n

A joint probability distribution for the random variables $\{R_n, G_n, H_n, Y_{mn}\}$ is defined in factored form as follows.

In the absence of prior information on the distribution of hosts across regions, independent uniform distributions are assumed for the node locations R_n. That is, for each region r,

$$\Pr(R_n = r) = \frac{1}{M}. \tag{1}$$

Of course, if information were available about the relative density of nodes in different regions, it could be encoded as an informative prior distribution on the locations R_n.

Conditional on the region in which n is located, the geolocation database and hostname evidence are independent with distributions given by:

$$\Pr(G_n = s|R_n = r) = \begin{cases} \alpha & r = s \\ \frac{1-\alpha}{M-1} & r \neq s \end{cases} \tag{2}$$

and

$$\Pr(H_n = t|R_n = r) = \begin{cases} \beta & t = r \\ \frac{1-\beta}{M-1} & t \neq r \end{cases} \tag{3}$$

That is, the model assumes there is a probability α that the geolocation database query returns the correct region, with the remaining probability distributed uniformly among the remaining regions. Similarly, there is a probability β that the hostname lookup returns the correct region, with the remaining probability distributed uniformly among the remaining regions. Again, this simple model could be modified to incorporate additional information if available.

Finally, the distribution of a link delay measurement depends on the distance between the starting and ending point of the link. A measurement probe packet traveling from a source to a destination may encounter other delays at each node. While propagation delay is directly proportional to distance, the true linear relation is distorted by the presence of other delays, including queueing, transmission and nodal processing delay. For Y_{mn} measuring link delay between nodes m and n, let D_{mn} denote the distance between the regions R_m and R_n where nodes m and n are located. Chandekar and Paris [7] considered a normal distribution for Y_{mn} given R_m and R_n:

$$\begin{aligned} f(Y_{mn} = y|R_m = r, R_n = s) = \\ f(Y_{mn} = y|D_{mn} = d) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y-ad-b)^2}{2\sigma^2}} \end{aligned} \tag{4}$$

with mean $ad + b$ a linear function of the distance between the starting and ending point, and standard deviation σ. They also noted that delay measurements may be biased due to the fact that some internet routers delay ICMP (Internet Control Message Protocol) replies. Thus, they also considered a Gaussian mixture model with different components for zero, positive and negative bias.

The distributions (1 - 4) are combined into a factored representation for the joint distribution of $\{R_n, G_n, H_n, Y_{mn}\}$. The joint probability mass / density function is:

$$\prod_{n \in \mathcal{N}} \Pr(R_n) \Pr(G_n|R_n) \Pr(H_n|R_n) \prod_{(m,n) \in \mathcal{L}} f(Y_{mn}|R_m, R_n) \tag{5}$$

where $\mathcal{N}$ is the set of IP nodes to be localized and $\mathcal{L}$ is the set of links, or pairs of nodes connected by signal transmission measurements.

Fig. 1, adapted from [7], depicts a factor graph for the joint distribution (5) when there are two nodes connected by a single link. A factor graph is a graphical probability model for a joint distribution represented in factored form [18]. The figure shows a Forney-style factor graph [19], [20], in which nodes are labeled by factors of the joint distribution and edges connect pairs of factors that share a random variable. Edges are labeled by the random variable shared between the factors at either end of the edge. If a random variable is shared by more than two factors, equality constraint nodes are inserted into the graph to "clone" random variables so each random variable is shared by no more than two factors. Evidence is shown as labels at the end of edges extending from the random variables whose values are observed. For example, evidence that G_m has value s_m is depicted at the terminus of an edge extending from the factor $\Pr(G_m|R_m)$.

From Fig. 1 it can be observed that the underlying physical topology determines the connectivity between random variables in the factor graph. The graph shows a cluster for each of the two hosts. Each of these clusters contains a factor for the prior distribution over the node's location, as well as a factor for evidence from the geolocation database query and a factor for evidence from the hostname lookup. These evidence items local to each host are henceforth referred to as node-local evidence. Fig. 1 also contains a link between the two clusters, which is labeled by a factor representing the delay measurement.

Extending this model to an arbitrary network results in a factor graph containing a node-local evidence cluster for each host in the network and an edge connecting any two clusters for which there is a link delay measurement. Thus, the factor graph structure mirrors the topology of the physical network. Each node-local evidence cluster corresponds to a physical IP node and the delay evidence edge corresponds to a connection (IP link) between nodes.

This mapping between network topology and factor graph allows for the systematic and simultaneous geolocation of a set of interconnected nodes using the joint probability distribution (5). This can be achieved by finding the joint posterior distributions of the node regions $\{R_n\}_{1 \leq n \leq M}$ conditional on database, hostname and delay evidence. For general network topologies, solving for the joint posterior distribution is intractable. However, the well-known sum-product algorithm [18] can be applied to estimate the joint posterior distribution. This algorithm operates by passing messages along edges of the factor graph to propagate evidence through the network. The factor graph model allows for the systematic update of

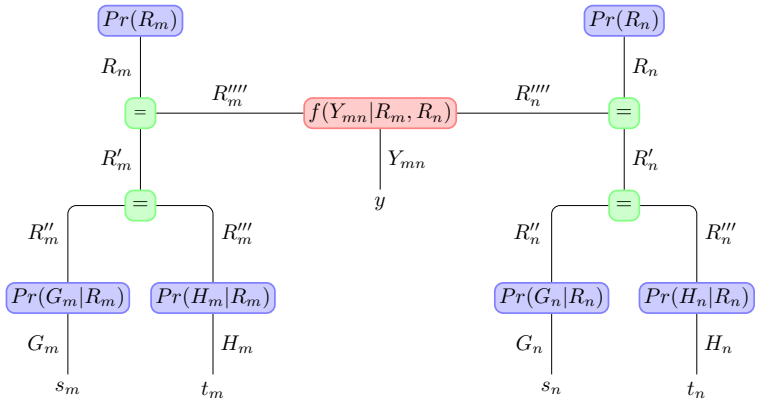


Fig. 1. Factor Graph representation for combining delay and node-local evidence to simultaneously geolocate a pair of nodes m,n

directly connected nodes, thereby reasoning about the location of a host based on the location of its directly connected hosts that may have inaccurate or missing node-local evidence.

An algorithm for automatically constructing factor graphs for arbitrary IP topologies is given in [7]. The authors applied their factor graph construction method to several test cases, applied the sum-product algorithm to find posterior distributions, and compared their results against ground truth. On both simulated and real-world test cases, they reported improved accuracy in geolocation from fusing delay data with node-local information.

IV. IP Geolocation Probabilistic Ontology

IP address geolocation is an important component capability for a broad variety of applications in cyber security and other information technology domains. Explicitly representing the the semantics of the IP address geolocation model can support model reuse across applications and interoperability with other kinds of models. For example, IP geolocation can contribute to predicting, diagnosing and responding to large-scale cyber attacks. Incorporating a geolocation capability into a cyber situation awareness and response system is facilitated by semantic awareness of the system and the component module.

As the factor graph in Fig. 1 makes clear, the geolocation model consists of modular elements that are assembled into a larger model to reason about a given network topology and patterns of available evidence. Additional evidence types can be added in a modular way by augmenting the graph with additional nodes and edges representing the new evidence. In a similar manner, the geolocation model could be used within a more comprehensive system. For example, a graphical probability model library for cyber attack plan recognition (*c.f.* [21]) could be augmented with elements representing geolocation information, which could then be referenced by attack plan models.

It is worth noting that the model of Section III does not consider intentional efforts by users to thwart attempts at geolocation. Reasons for evading geolocation are diverse, including privacy concerns, overcoming geographical restrictions on content access, and disguising the source of cyber-attacks. To address the problem of IP spoofing, a technique common in denial of service (DoS) attacks, it is important to draw a distinction between user geolocation and IP geolocation. As pointed out by [22], user geolocation seeks to identify the location of a user who requests content or attempts to connect to a specific resource, whereas IP geolocation seeks to identify the geographic location of a device given its IP address. An IP geolocation capability such as the one presented above can support user geolocation by tracing the path of a spoofed packet to the network edge and geolocating the device nearest to the origin. Other techniques exist to extract the actual IP address of the attacker, which can be geolocated using an IP geolocation method. Again, our IP geolocation could be combined with additional modular components to form a user geolocation capability.

Representing the model as a probabilistic ontology supports this kind of model interoperability and reuse. Ideally, such a probabilistic ontology would be built on an existing ontology of the cyber domain (*e.g.*, [2]). As such, many of the random variables in the model should already be represented in the ontology, and probabilistic ontology development would largely involve augmenting the existing ontology with information about uncertainties. For the purpose of illustrating the approach, we constructed a limited, partial ontology consisting of entities, properties and relationships needed to reason about IP geolocation and augmented that ontology with uncertainty information. Clearly, a comprehensive ontology of the cyber domain would represent additional general knowledge and specific domain knowledge not included here.

Our probabilisitic ontology is represented in the PR-OWL language [6] and implemented in the UnBBayes-MEBN open-source PR-OWL reasoning tool [23]. Our representation includes some workarounds to overcome limitations of the current version of UnBBayes-MEBN. These limitations will be addressed in future releases. The model encoded in the probabilistic ontology is equivalent to the IP geolocation model presented above.

Table II lists the entities in the partial ontology and properties used by the node geolocation probabilistic ontology. The ontology has four types of entity: IP nodes, regions where IP

nodes can be located, probe packets for measuring link delays, and evidence items. Because this is an OWL ontology, all four types are subtypes of `Thing`.

TABLE II
ENTITIES AND ATTRIBUTES IN GEOLOCATION PROBABILISTIC ONTOLOGY

Entity	Property	Description
`IPNode`	`Location`	Region in which IP node is located
`Region`	`RegionID`	Unique identifier for a region
`ProbePacket`	`StartingNode`	Starting node for a link delay measurement
	`EndingNode`	Ending node for a link delay measurement
`EvidenceItem`	`ReportedNode`	IP node to which a database query or hostname lookup refers
	`GeoIPReport`	Region returned by database query on IP node
	`HostnameReport`	Region returned by hostname lookup on IP node
	`ReportedProbe`	Probe packet to which a link delay measurement refers
	`DelayReport`	Measured delay for a probe packet sent across a link

A property of an IP node is its location. A property of a region is its region ID, a unique identifier used to refer to the region. Properties of a probe packet include its starting and ending nodes. Properties of an evidence item include the IP node to which it refers for node local evidence, the content of a GeoIP query response, the content of a hostname lookup result, the probe packet measured by a link delay report, and the measured delay for a probe packet sent across a link.

Table III shows the relationships represented in the ontology. The entity types participating in the relationship are shown. The `IsA` relationship relates an entity and a type if the entity is of the given type. The ontology includes the relationships `NodeDistance` and `RegionDistance` to represent the distance between nodes and regions, respectively. Ideally, there would be only one `Distance` attribute to represent the distance between two spatial entities. However, the current UnBBayes-MEBN implementation does not yet support polymorphism; this capability is slated for the next release. Thus, the ontology uses two different terms to accommodate the limitations of the reasoning tool.

Fig. 2 shows the Node Geolocation probabilistic ontology. The probabilistic ontology consists of five MFrags (Multi-Entity Bayesian Network Fragments). Each MFrag defines a local probability distribution for its resident random variables, depicted by yellow ovals, conditional on their parents in the MFrag. The context random variables, depicted by green pentagons, represent conditions that must be satisfied for the local distribution definitions to be meaningful. Finally, the gray trapezoids are input random variables, which are parents of resident random variables whose distribution is defined in another MFrag.

The random variables in the MFrags define a joint probability distribution over properties and relationships in the ontology. A random variable with a single argument corresponds to a property, and a random variable with two arguments corresponds to a relationship. The arguments are placeholders (called ordinary variables to distinguish them from random variables) that can be filled in by the identifiers of individuals of the appropriate types. For example, if `N1` and `N2` are individuals of the `IPNode` type, the random variable $\texttt{Location}(\texttt{N1})$ represents the uncertain location of `N1`, and $\texttt{NodeDistance}(\texttt{N1}, \texttt{N2})$ corresponds to the distance between IP nodes `N1` and `N2`. The second argument of the `IsA` random variable is always a type name, indicating the type of its first argument. Thus, $\texttt{IsA}(\texttt{N1}, \texttt{IPNode})$ has value *True* and $\texttt{IsA}(\texttt{N1}, \texttt{Region})$ has value *False*. Multiple instances of these MFrags can be constructed by filling in the ordinary variables with different entity instances. The MFrag instances can then be assembled into a Bayesian network called a situation-specific Bayesian network, or SSBN.

TABLE III
RELATIONSHIPS IN GEOLOCATION PROBABILISTIC ONTOLOGY

Relationship	Entities	Description
`IsA`	`Thing`, `Type`	Indicates that an entity is of the referenced type
`NodeDistance`	`IPNode`, `IPNode`	Distance between two IP nodes (real number)
`RegionDistance`	`Region`, `Region`	Distance between two regions (real number)

The MFrags and local distributions are described as follows.

- *Node Location*: This MFrag defines a distribution for the *Location* random variable, representing the region in which an IP node is located. This random variable corresponds to the `Location` property of the `IPNode` entity. It also corresponds to the random variable R_n in the factor graph of Fig. 1. Its possible values are regions and it is given a uniform distribution, meaning that all regions are equally likely locations for any given node.
- *Distance*: This MFrag defines distances between regions. The *RegionDistance* random variable is initialized to a uniform distribution (or a Gaussian distribution with mean zero and very large variance). When region instances are defined, their respective *RegionDistance* random variables are set to the actual distance between each pair of regions. The *NodeDistance* random variable has a deterministic distribution, being equal to the distance between the regions in which its arguments are located. These random variables define distributions for the *RegionDistance* and *NodeDistance* relationships from Table III.
- *Probe Packet Definition*: This MFrag defines random variables *StartingNode* and *EndingNode* for probe packets sent across links. The distributions are initialized as uniform. When a delay measurement is received, they are set to the starting and ending node for the the probe packet. These random variables define distributions for the `StartingNode` and `EndingNode` properties of a

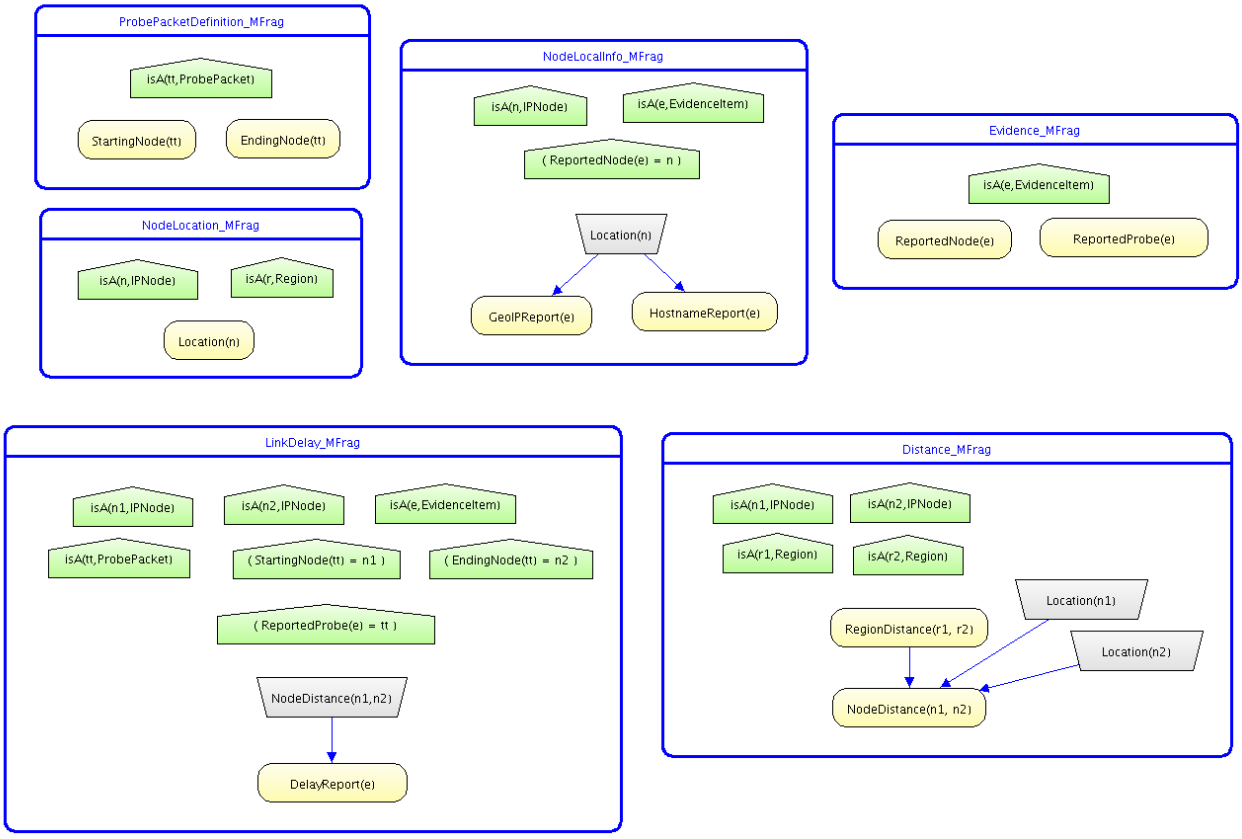


Fig. 2. Node Geolocation Probabilisitic Ontology

`ProbePacket` entity as defined in Table II.

- *Evidence*: This MFrag defines random variables *ReportedNode* and *ReportedProbe* for evidence items. The former defines the IP node to which a GeoIP query or a hostname lookup refers. The latter defines the probe packet to which a given link delay measurement refers. These random variables are initialized to uniform distributions and are set to the appropriate values when reports are received. These random variables define distributions for the `ReportedNode` and `ReportedProbe` properties of `EvidenceItem` entities as defined in Table II.
- *Link Delay*: This MFrag defines the distribution for a link delay measurement conditional on the distance between the starting node and ending node for the corresponding probe packet. The *DelayReport* random variable corresponds to the random variable Y_{mn} in the factor graph of Fig. 1. It has the normal distribution given by (4), or a mixture of normal distributions if router delay is being considered in the model. This random variable defines the distribution for the `DelayReport` property of an `EvidenceItem` entity as presented in Table II.
- *Node Local Information*: This MFrag represents evidence local to a given node. The resident nodes *GeoIPReport* and *HostnameReport* correspond to the random variables G_n and H_n, respectively, in the factor graph of Fig. 1. The local distributions for *GeoIPReport* and *HostnameReport* are given by (2) and (3), respectively. These random variables define distributions for the `GeoIPReport` and `HostnameReport` properties of an `IPNode` entity shown in Table II.

The probabilistic ontology is applied to a given network topology and set of measurements as follows. Assume that we are given a set of nodes, a set of regions, a network topology defining node connectivity, link delay measurements for nodes connected by the topology, and GeoIP query and hostname lookup results for some or all of the nodes. Inference about node locations proceeds as follows.

1) Create an instance of `Region` for each region. Define the regions as mutually exclusive. Give each region an ID, and set the value of `RegionID` to the region's ID. For each pair of regions, set the value of `RegionDistance` to the distance between the regions.
2) Create an instance of `IPNode` for each node in the network. Define the nodes as mutually exclusive.
3) Create an instance of `ProbePacket` for each probe packet for which the propagation delay has been measured. Set the properties `StartingNode` and `EndingNode` to the instances of `IPnode` corresponding

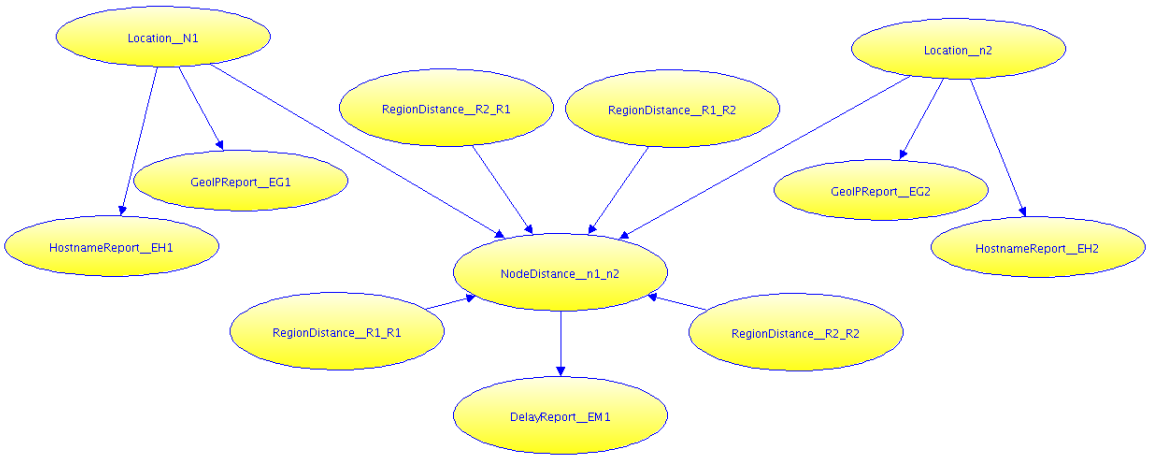


Fig. 3. Constructed Bayesian network for geolocating a pair of IP nodes

to the endpoints of the link.

4) Create an instance of `EvidenceItem` for each GeoIP query result. For each report, set the property `ReportedNode` to the `IPnode` instance to which the report refers, and the property `GeoIPReport` to the region indicated by the report.
5) Create an instance of `EvidenceItem` for each hostname lookup result. For each report, set the property `ReportedNode` to the `IPnode` instance to which the report refers, and the property `HostnameReport` to the region indicated by the report.
6) Create an instance of `EvidenceItem` for each link delay measurement. For each report, set the property `ReportedProbe` to the instance of `ProbePacket` to which the report refers. Set the property `DelayReport` to the measured delay across the link.
7) Run a query to find the posterior distribution of the `NodeLocation` properties. This involves assembling a situation-specific Bayesian network containing the random variable instances created in the above steps. The MFrags containing the random variable instances are retrieved and instantiated, and then combined by unifying on common random variables. The result is a Bayesian network to reason about IP node locations.

Fig. 3 shows the situation-specific Bayesian network produced by the UnBBayes software for the case of a network with two nodes connected by a single link. The Bayesian network of Fig. 3 and the factor graph of Fig. 1 encode the identical joint distribution for node locations and evidence. The model of (5) and the probabilistic ontology of Fig. 2 encode formally equivalent joint distributions over node locations, node local evidence and transmission delay evidence.

The SSBN of Fig. 3 is a hybrid Bayesian network containing both discrete and continuous random variables. The distance random variables are real-valued and continuous; the other random variables are discrete. The graph of Fig. **??** is a polytree, and exact inference is possible using an algorithm for conditional linear Gaussian (CLG) Bayesian networks. For larger networks with complex network topologies, the graph will contain cycles and exact inference is intractable. Approximate inference algorithms such as the one presented in [24] can be applied.

V. Conclusion

Explicitly representing domain semantics in computable form supports maintainability, interoperability and extensibility of systems. For problems characterized by reasoning under uncertainty, a semantically rich representation for sources of uncertainty should be appropriately integrated with the domain ontology. This paper presented a case study of a probabilistic ontology for large-scale IP address geolocation. The probabilistic ontology integrates an existing factor graph model for IP geolocation with a domain ontology representing geolocation knowledge. Random variables in the factor graph model correspond to uncertain properties and relationships in the domain ontology. The model is represented as a PR-OWL probabilistic ontology that augments an OWL domain ontology by defining probability distributions for uncertain properties and relationships. Reasoning with the probabilistic ontology is performed by creating instances of the relevant entities, instantiating copies of the random variables by filling in their arguments with appropriate entity instances, and assembling them into a Bayesian network to reason about the particular problem instance. The model can be used to reason about arbitrary numbers of IP nodes and regions, arbitrary network topologies, and arbitrary numbers of evidence items.

Acknowledgment

The authors thank Shou Matsumoto for assisting with development of the UnBBayes representation of the probabilistic ontology.

REFERENCES

[1] A. Kott, "Towards Fundamental Science of Cyber Security," in *Network Science and Cybersecurity*, R. E. Pino, Ed., New York, 2014, vol. 55.

[2] A. Oltramari, L. Cranor, R. Walls, and P. McDaniel, "Building an Ontology of Cyber Security," in *Proceedings of the Ninth Conference on Semantic Technologies for Intelligence, Defense, and Security (STIDS 2014)*, ser. CEUR Workshop Proceedings, K. B. Laskey, I. Emmons, and P. C. G. Costa, Eds. Aachen: George Mason University, 2014, pp. 54–61. [Online]. Available: http://ceur-ws.org/Vol-xxx/

[3] R. Dipert, "The Essential Features of an Ontology for Cyberwarfare," in *Conflict and Cooperation in Cyberspace - The Challenge to National Security,*, P. A. Yannakogeorgos and A. B. Lowther, Eds. Taylor and Francis, 2013.

[4] B. Barnett and A. Crapo, "A Semantic Model for Cyber Security," in *Proceedings of the Fifth Grid-Interop Forum*. Gridwise Architectural Council, 2011. [Online]. Available: http://www.gridwiseac.org/pdfs/forum_papers11/barnett_paper_gi11.pdf

[5] N. F. Noy and D. L. Mcguinness, "Ontology development 101: A guide to creating your first ontology," Knowledge Systems Laboratory, Stanford University, Tech. Rep., 2001.

[6] K. Laskey, P. Costa, and T. Janssen, "Probabilistic ontologies for knowledge fusion," in *2008 11th International Conference on Information Fusion*, Jun. 2008, pp. 1–8.

[7] S. Chandekar and B.-P. Paris, "Large-scale, discrete IP geolocation via multi-factor evidence fusion using factor graphs," in *18th International Conference on Information Fusion*, Jul. 2015.

[8] R. Koch, M. Golling, and G. D. Rodosek, "Advanced Geolocation of IP Addresses," in *International Conference on Communication and Network Security (ICCNS)*, 2013, pp. 1–10. [Online]. Available: http://www.waset.org/publications/16111

[9] M. Crovella and B. Krishnamurthy, *Internet Measurement infrastructure, traffic and applications*. England: John Wiley and Sons, 2006.

[10] MaxMind LLC, "GeoIP." [Online]. Available: http://www.maxmind.com

[11] V. N. Padmanabhan and L. Subramanian, "An investigation of geographic mapping techniques for internet hosts," in *ACM SIGCOMM Computer Communication Review*, vol. 31, no. 4. ACM, 2001, pp. 173–185.

[12] B. Gueye, A. Ziviani, M. Crovella, and S. Fdida, "Constraint-based geolocation of internet hosts," *IEEE/ACM Transactions on Networking*, vol. 14, no. 6, pp. 1219–1232, 2006.

[13] A. Ziviani, S. Fdida, J. F. de Rezende, and O. C. Duarte, "Improving the accuracy of measurement-based geographic location of internet hosts," *Computer Networks*, vol. 47, no. 4, pp. 503–523, 2005.

[14] M.J.Arif, S.Karunasekara, S.Kulkarni, A.Gunatilaka, and B.Ristic, "Internet host geolocation using maximum likelihood estimation technique," *IEEE International Conference on Advanced information Networking and Applications*, 2010.

[15] I. Youn, B. L. Mark, and D. Richards, "Statistical geolocation of internet hosts," in *Computer Communications and Networks, 2009. ICCCN 2009. Proceedings of 18th Internatonal Conference on*. IEEE, 2009, pp. 1–6.

[16] DIMES, "Ip topology," http://www.netdimes.org.

[17] Y. Shavitt and E. Shir, "Dimes: Let the internet measure itself," *ACM SIGCOMM Computer Communication Review*, vol. Vol 35, October 2005.

[18] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, 2001.

[19] H.-A. Loeliger, J. Dauwels, J. Hu, S. Korl, L. Ping, and F. R. Kschischang, "The factor graph approach to model-based signal processing," *Proceedings of the IEEE*, vol. 95, no. 6, pp. 1295–1322, 2007.

[20] G. D. Forney Jr, "Codes on graphs: normal realizations," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 520–548, 2001.

[21] C. W. Geib and R. P. Goldman, "A probabilistic plan recognition algorithm based on plan tree grammars," *Artificial Intelligence*, vol. 173, no. 11, pp. 1101–1132, Jul. 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0004370209000459

[22] J. A. Muir and P. C. V. Oorschot, "Internet geolocation: Evasion and counterevasion," *ACM Computing Surveys (CSUR)*, vol. 42, no. 1, p. 4, 2009.

[23] P. C. G. d. Costa, M. Ladeira, R. N. Carvalho, K. B. Laskey, L. L. Santos, and S. Matsumoto, "A First-Order Bayesian Tool for Probabilistic Ontologies," in *FLAIRS Conference*, 2008, pp. 631–636.

[24] W. Sun, K.-C. Chang, and K. Laskey, "Scalable inference for hybrid Bayesian networks with full density estimations," in *2010 13th Conference on Information Fusion (FUSION)*, Jul. 2010, pp. 1–8.

Towards a Human Factors Ontology for Cyber Security

Alessandro Oltramari
Carnegie Mellon University
Pittsburgh, USA

Diane Henshel & Mariana Cains
Indiana University
Bloomington, USA

Blaine Hoffman
Army Research Laboratory
Aberdeen, USA

***Abstract*— Traditional cybersecurity risk assessment is reactive and based on business risk assessment approach. The 2014 NIST Cybersecurity Framework provides businesses with an organizational tool to catalog cybersecurity efforts and areas that need additional support. As part of an on-going effort to develop a holistic, predictive cyber security risk assessment model, the characterization of human factors, which includes human behavior, is needed to understand how the actions of users, defenders (IT personnel), and attackers affect cybersecurity risk. Trust has been found to be a crucial element affecting an individual's role within a cyber system. The use of trust as a human factor in holistic cybersecurity risk assessment relies on an understanding how differing mental models, risk postures, and social biases impact the level trust given to an individual and the biases affecting the ability to give said trust. The Human Factors Ontology illustrates the individual characteristics, situational characteristics, and relationships that influence the trust *given* to an individual. Furthering the incorporation of ontologies into the science of cybersecurity will help decision-makers build the foundation needed for predictive and quantitative risk assessments.**

Keywords*— *cyber security, risk assessment, human factors, cyber operations

I. INTRODUCTION

A. The Holistic Cybersecurity Risk Framework

The science of cybersecurity risk assessment has been reactive, narrow in focus, and based on a business risk assessment approach. More recently, the National Institute of Science and Technology (NIST) responded to the 2013 "Improving Critical Infrastructure Cybersecurity" Executive Order with the development of the 2014 NIST Cybersecurity Framework [1,2]. The NIST framework aims to provide organizations and businesses with best risk management practices that can be implemented to improve the security and resilience of critical infrastructure. NIST recognizes that risk management is an iterative process of risk identification, risk assessment, and risk mitigation. While the NIST framework provides businesses and organizations with a neatly organized account of their cybersecurity efforts, the framework fails to capture the concept that humans are an inherent risk to any system in which they directly or indirectly participate.

To go beyond the current risk framework promulgated by NIST [1,2], risk assessment needs to be more holistic. In order to enable cybersecurity risk assessment to become more predictive, the process and models need to incorporate humans and risk factors together in a single model and use metrics that go beyond the direct assessment of classical vulnerabilities (confidentiality, integrity, accessibility, or CIA).

First, when considering CIA, the actual measurement or evaluation of these vulnerabilities will depend on the situation being modeled. Situations requiring cybersecurity risk assessment can include baseline assessments of network protection, but must also include situations in which the network is being used actively. The actual metrics for, say, protection of an SQL database containing personal information (social security numbers, for example) may be very different than the metrics needed to be assessed when evaluating risk related to a field operation using radios, walkie talkies or cell phones to convey information.

Second, other variables beyond CIA may be the relevant risk variables that need to be analyzed in a risk model. Take, for example, a situation in which information being used, generated in, or relayed by one network needs to be received in a specific time window either for another operation to begin or so that the information can be used maybe by the human who will receive the information. Within a military or other time critical context, the evaluation goes beyond time to access information; it must include time to act on the accessed information and can include time for completion of actions within a critical time window. In this example, time to completion of a task is the critical metric that must be tracked, and so must be incorporated into the risk model.

Third, humans are a part of virtually all networks, whether as users, defenders (and IT personnel) or attackers. All humans can introduce risk into the network, not just attackers, a consideration acknowledged when users are asked how they use the system (and system components) as part of the NIST risk management and risk assessment process. Defenders or IT personnel can also increase cyber risk if they are, for example, less skilled, or tired, or inside threats. Humans can also reduce risk in a cybersecurity system. Defenders put in place baseline protections, and then track attacks on the system to assess whether the protections have been breached and what needs to be done to increase system hardening (protections), counteract

malware that may have introduced access to the system (or otherwise compromised the system and system assets), and repair damage to the system. Users can decrease risk by being aware of (and not being hooked by) spam or phishing efforts, ensuring their personal system assets are appropriately protected, and by not downloading infected files or accessing malware-linked websites. Therefore, human-dependent metrics must be included in a holistic risk analysis of cyber security.

A fully predictive cyber security risk assessment model will take into account humans as risk factors, and as risk mitigators, and will enable the incorporation of metrics that go beyond the classic CIA vulnerabilities. In order to develop such a model, we have been characterizing the universe of cybersecurity by framing the characteristics, attributes and, ultimately, metrics that can be use to describe the risks associated with any cyber network. The framework has multiple pieces, and metrics that are assessed at different levels.

Three main parts to the Cybersecurity Risk Framework identifies system level metrics, policy related metrics, and asset related metrics. System level metrics are evaluated at the full system level, such as probability of completion of a mission or a system level task. Policy level metrics evaluate the risks associated with the policies that govern the network and network assets. Asset level metrics are evaluated at the asset level, such as metrics to assess risks associated with specific machines, a virtual network, or an operating system. One piece of the asset level framework characterizes the Human Factors that introduce or mitigate risk in a cyber network [3], which is then being incorporated into an ontology. One goal of this framework and ontology is to identify the factors that contribute to a key aspect of human-related cyber risk, trust.

B. An ontological approach to risk modeling

A recent report on quantification of cyber threats highlights the intrinsic complexity of the cyber domain [4]: in this document experts pinpoint the bottleneck of cyber threat assessment on the lack of "standardization and benchmarking of input variables", as conversely accomplished – they add – "by the car insurance industry" (p.16). But if agreeing on the meaning of notions like 'age' and 'gender' of drivers, 'weight' and 'year of built' of cars, 'claims history', etc. seems mostly straightforward, specifying the semantics of concepts like 'system vulnerability', 'software usability', 'trust', 'password strength', etc. requires advanced technical knowledge, fine-grained modeling primitives, and non-trivial metrics.

Little effort has been put into this standardization process. For instance, Fenz and Ekelhart propose an ontology based on four parts, i.e. security and dependability taxonomy, the underlying risk analysis methodology, the concepts of the IT infrastructure domain and a simulation enabling enterprises to analyze various policy scenarios [5]. Notwithstanding the comprehensive investigation, the work presented in [5] is affected by an underspecified notion of risk, conceived as "the probability that a successful attack occurs", which clearly fails to account for the mutual dependence between profiles of attackers, system vulnerabilities, level of expertise of the defenders, monetization of information loss resulting from data breaches, etc. In general, a too-coarse representation of risk is a pervasive problem in the state of the art on ontologies of cyber security: it's the case of [6] and [7] where the in-depth conceptual distinctions adopted to model cyber attacks are not matched by a corresponding level of detail in defining cyber threats and risk assessment procedures.

The most popular modeling solution in risk-related ontology research seems to be the reification of risk-assessment and threat-quantification into the process of 'rating', whose attributes are expressed either qualitatively (e.g., by means of high, medium and low dimensions in the Likert scale) or quantitatively (measuring the probability of a risk). Note that in ontology modeling, reification of properties is commonly adopted as a method to bypass language expressivity limits: in RDF, for instance, a relation with arity $n > 2$ can be represented with a statement about those n entities. Thus, for instance, we could represent the fact that a set of n cyber vulnerabilities exposes a system to a certain risk factor, by asserting a risk-rating statement about those known n vulnerabilities [8]. An alternative approach comes from Enterprise Risk Management (ERM), an area that concerns the identification, assessment and mitigation of operational risk: for instance, Lykourentzou and colleagues focus on seven subclasses of events, i.e. 'Failure', 'Infrastructure disruption', 'Occupational incident', 'Fraud', 'Disaster', 'Attack', binding each of these event types to a wide spectrum of 'Root causes' and 'Treatment plans' to address risk factors [9]. ERM's approaches can be effective not only to identify risk-related event patterns, but also to elicit the behavioral patterns in the adoption of risk management practices. In this context, ontologies supply an axiomatic infrastructure to mental models of risk-related patterns.

The rest of the paper is organized as follows: Section II makes the case for a holistic approach to risk in cyber security, introducing the role of trust ontologies; Section III focuses on the Human Factors Ontology (HUFO); finally, Section IV draws preliminary conclusions and sets an agenda for future research.

II. RELATED WORK

A. Ontologies of cyber security

The U.S faces cyber attacks by rogue states and terrorist organizations on a daily basis. While greatly increased use of information systems has contributed enormously to economic growth, it has also made the U.S. vulnerable to a variety of cyber threats that are difficult to contrast and prevent. There are numerous factors that make cyber defense, and cyber security in general, especially problematic. The kinds of threats are diverse and span a wide spectrum of private and public interests: destruction or

theft of data, interference with computer networks and information systems, disruption of the power grid and telecommunications, denial of services, etc. The legal and ethical status of cyber attacks or counterattacks by states are also unclear, at least when deaths or permanent destruction of physical objects does not result. It is still an open question what U.S. policy is or should be, and how cyber threats are analogous to traditional threats and policies—for example whether "first use" deterrence, and in-kind responses apply, and whether a policy of pure cyber defense does not put the far greater burden on attacked rather than attacking nations [10].

As these arguments suggest, untangling the complexity of cyber security does not solely depend on pinning down the computational elements into play, but demands a thorough analysis of the human factors involved. In this regard, cyber security must be studied in the context of "sociotechnical systems" [11], where the interaction between people and technology in workplace is central. Ontology analysis has recently proved to be an effective tool for investigating the defining aspects of that interaction [12].

Informed decisions emerge when a cyber analyst projects her observations into a broad context that factors in threat and attack types, space of defensive maneuvers, system vulnerabilities, risk assessment and mitigation under time constraints. Obrst and colleagues [13] provide the most systematic description of a wide-ranging ontology of cyber security, but only a small portion of this large-scale project is devoted to the human component. Various agencies and corporations (NIST [1,2], MITRE [14], and Verizon [15]) have formulated enumerations of types of malware, vulnerabilities, and exploitations: MITRE, which has been very active in this field, maintains two dictionaries, namely CVE (Common Vulnerabilities and Exposure[1]) and CWE (Common Weakness Enumeration[2]) and a classification of attack patterns (CAPEC - Common Attack Pattern Enumeration and Classification [3]). Regardless of the important issues covered by these initiatives, they have two major problems: 1) machine-readability is not supported, making them ineffectual as computational models of cyber security; 2) the human component is mostly overlooked, making the resulting models partial in scope.

In order to overcome these problems, in the context of the Cyber Collaborative Research Alliance we are developing CRATELO, a three-level modular ontology of cyber security. In the next section we are going to describe the general features of CRATELO, focusing on the Human Factors Trust Ontology module (HUFO).

B. *Trust ontologies*

Ontology-based models of trust have been studied in various domains [16]. In [17], the authors propose an intelligent and dynamic Service Level Agreement (SLA) based on a probabilistic ontology that detects warnings in a cloud computing environment. A generic service-oriented framework of trust ontologies is described in [18]. A trust ontology aiming at improving the semantic specification of trust networks in the context of social institutions and ecosystems is discussed in [19]. In [20], the author focuses on six general areas to derive trust for a system, namely user, hardware, software, network, machines, and the applications, mapping trust associated with each area to specific attributes. An ontology-based approach to integrate semantic web based trust networks with provenance information to evaluate and filter a set of assertions is presented in [21]. In [22], a reference ontology to develop privacy preserving negotiation systems is delineated.

III. The Human Factors Trust Ontology

A. *The Human Factors Trust Ontology*

Adopting a standard understanding and definition of terms and concepts is a foundational requirement for good cyber security practice, owing to the nature of the space and the need for rapid, efficient decision-making. Cyber security is an adversarial space, where defenders must project possibilities and be ahead of their opposition in order to be successful. Enacting strategies favors selecting a suitable course of action in minimal time over exhaustively searching [23,24]. Furthermore, the data available is not always straightforward, requiring collection and parsing in order to construct an understanding of the situation(s) at hand. Numerous sources of relevant information are often applicable, including network monitoring tools, logs, system statuses, and hardware monitors. Analysts are situated at the center of a large-scale data fusion process, identifying and defining information through patterns and relationships to perceive the ground truth of the cyber systems and assets they are defending and monitoring [25,26,27]. Once collected, the information must be appropriately combined, categorized, and communicated in order to provide a useful and accurate picture of the world on which future strategies can be based. Simply stated, cyber defense is heavily focused on the human analysts and agents involved in a data fusion and situation awareness process.

Through processing of data, defenders can draw conclusions and decide how to respond to evolving scenarios. Implicit within the workload is a desire and preference for information that can be *trusted*, a concept that requires a lot of unpacking to properly understand. In fact, conceptualizing trust in order to evaluate its role and presence within a system is itself a difficult problem; there are literally hundreds of definitions of trust covering interpersonal trust, trust in automation (system trust), and human-machine interaction [28]. However, that variety only strengthens the argument for constructing and supporting an ontological representation of cyber security. The core similarities of cyber security and the tasks involved are essentially the same [29], which also supports the creation

[1] https://cve.mitre.org/
[2] https://cwe.mitre.org/
[3] https://capec.mitre.org/

of a standard ontology. Thus we should be able to describe the human factors that influence trust in a way that can be applicable regardless of the specific cyber environment or organization involved and that will help explicate the role of trust in risk assessment and evaluation.

Assessing cyber security risks is a multi-component, multi-tiered problem that involves hardware, software, environmental, and human factors. Effective and successful efforts must consider impacts beyond the computer assets and network, taking a more holistic approach that considers the users, defenders, and attackers involved [3]. Exploring the differences among human roles and human factors includes exploring how trust permeates risk assessment, such as trust in information, in people, or in security policies. Information is not uniformly trusted and incorporated into situation awareness and defender responses automatically, but it is built over time as those involved develop relationships, progress through training, and gain experience [30]. Individuals grow trust in one another through working together, and people gain trust in systems as they continue to demonstrate consistent behavior. Previous definitions of trust aggregate characteristics into a whole sum, including concepts such as competence, benevolence, integrity, predictability, attitude, intention, behavior, reliability, dependability, and faith [31] [32] [20]. The human factors trust ontology aims to map these concepts into understood and explicit relationships that tie together risk assessment across the human and human-system interactions within the cyber security space.

As part of an ongoing development of holistic cyber security risk assessment, we have been creating a framework that enables predictive and proactive defenses [33,34,35]. A critical component of this process has been the characterization of human factors, such as trust, and mapping the relevant risk attributes to the risk spaces involved in cyber security. Overall, this is a process of creating, enumerating, and solidifying risk characteristics and factors, and in many cases refining them and relating them to the human factors. The latter are broken into three main categories of attacker, defender, and user with a shared core of spaces (their *behavioral characteristics*, *knowledge and skill characteristics*, *situational characteristics*, and traits that influence behavior) that create the definition of each [3]. The framework (see Figure 1) can be navigated from top to bottom, the lower tiers breaking out into the more specific metrics and concepts that, collectively, describe and detail these core spaces, which allows for the mapping of attributes to measures and data that can be used to create risk evaluations.

Situational Characteristics focus on where in the system/network the individual is positioned and the level of insider access they possess, denoting when this access is authorized or unauthorized. A person's situational characteristics also influence the knowledge they can access and may influence the attention they bring to a situation. For example, a user who is an executive of a company may have significant authorized access to assets but lack the same level of attentiveness to security concerns and information that a network analyst possesses. *Knowledge and skill characteristics* call to attention the experience, expertise, and situational awareness capabilities of the individual, including demographics such as years working in a position and training as well as their proficiency with relevant tools and techniques. *Behavioral Characteristics* are split into spaces such as motivation, rationality, malevolence vs. benevolence, and integrity. For example, a defender who is rational, benevolent, and has a record of following through with work and being accountable for his or her responsibilities will likely exhibit persistence in defending assets and building appropriate situational awareness. We have expanded the framework to include traits that influence the behavioral characteristics, including ideology, ethical attributes, risk averseness, and personality traits. Each of these may scale the behavioral characteristics in some fashion or serve as the driving force behind a person's integrity, benevolence, or rational approach to cyber security situations. Collectively, these characteristics and traits impact the individual's interactions with mission assets and play a role in determining risk. For example, defender with poor motivation and integrity, insufficient knowledge, and appropriate insider access can present a higher risk, whereas an attacker with high motivation and knowledge despite limited insider access also poses higher risk.

Trust also comes through across these spaces. The predictability and reliability of an individual generates a sense of trust in his or her actions and creates a reputation for that individual. The expertise and knowledge possessed can instill a faith or confidence in the work a defender will do, and users with sufficient integrity will be trusted to follow security policy and not act maliciously within the network. In effect, the human factors of trust directly associates with risk evaluation of cyber situations, and we can explore the relationships across the human factors of cyber security to discover *where* risk manifests and how trust is generated and influenced. Integrating the human factors framework into a cyber security ontology provides a logical means to explicate relationships both obvious and unintuitive, follow their connections, and evaluate trust's presence and impact on the risk present within a given network.

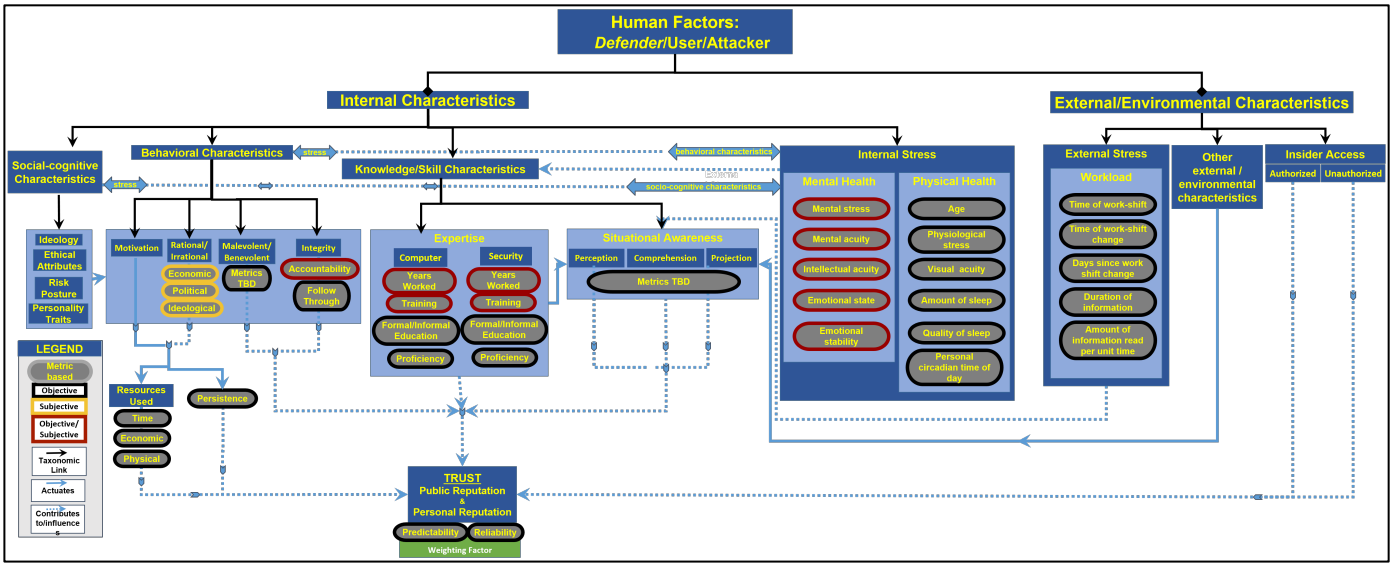


Figure 1 – Trust Framework of Human Factors in Cyber Security.

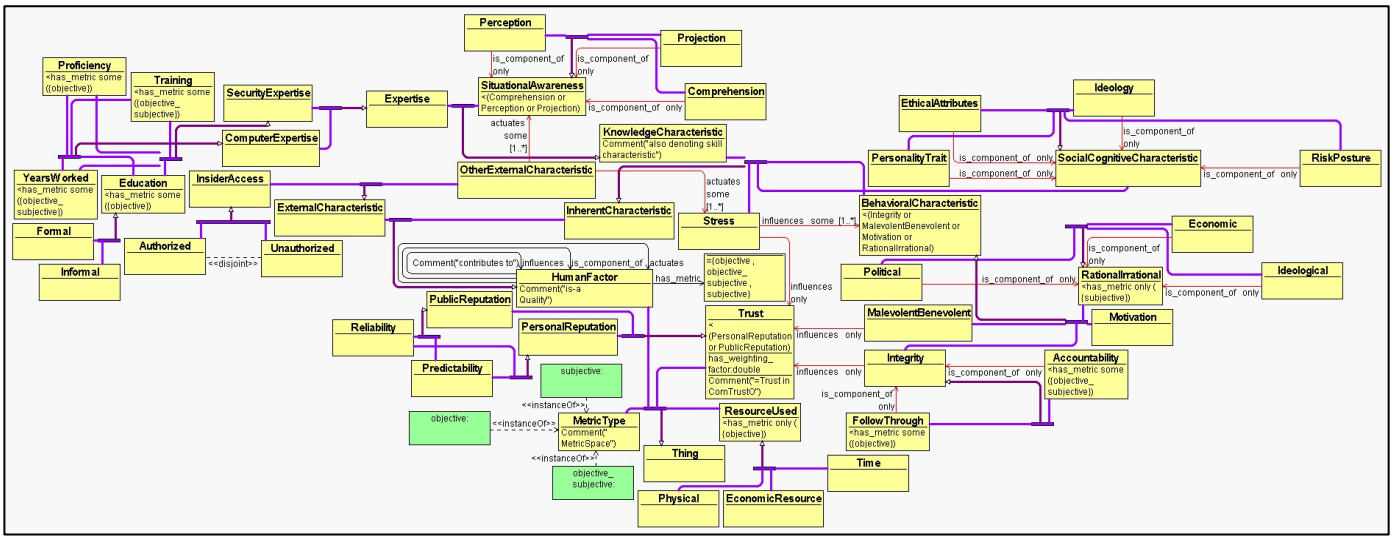


Figure 2 – A visualization of HUFO.

B. *HUFO and Trust: an overview*

HUFO (see Figure 2 above) is part of CRATELO [36], a suite of integrated ontologies of cyber security, designed on the basis of DOLCE top level [37], extended with a security-related middle ontology. These top, middle and domain level ontologies currently add up to 330 classes, connected by 162 relationships (132 object properties and 30 datatype properties) and encoded in OWL-DL. The logical expressivity of CRATELO is SRIQ, a decidable extension of the description logic SHIN (for more details see [38]).

The relation holding between the human factors and the metrics used to assess them is captured by the semantic characterization of 'qualities' and 'quality spaces', which has been originally formulated by [39] and subsequently formalized in DOLCE ontology [37]. Intuitively, a quality corresponds to an individual attribute of a specific entity, as 'predictability' or 'reliability' can be considered attributes of 'trust'; a quality space is the abstract representation of an attribute's semantics, e.g. a boolean space that denotes the 'reliable/unreliable' dichotomy. An important topological property of quality spaces is that their dimensional structure can vary. For instance, the 'reliability space' can be more complex than a bidimensional configuration: in particular, this is the case when reliability is conceptualized as probabilistic distribution between maximum reliability (100%) and complete unrealibility (0%). The atomic parts of a quality space, which collectively denote the range of

values used to specify an attribute's semantics, are called 'quality regions'. Note that quality regions of a linear space reduce to points.

As mentioned above, 'predictability' and 'reliability' are conceived in HUFO as components of 'trust', a complex factor that is influenced by inherent and external characteristics, in combination with measures of human performance in a given situation. Hence, trust is not only associated to human characteristics, but emerges as an essential aspect of sociotechnical systems: the hybrid nature of trust is particularly evident in the cyber security domain, where a trustworthy interaction with computer network systems is the '*conditio sine qua non*' for a defender/attacker to accomplish a mission in cyberspace[4].

Figure 2 represents an overview of HUFO generated using OWLGrEd[5]: the purple links represent subsumption relationship between classes, whereas the dotted arrows indicate either the 'component-of' or the 'influenced-by' property (textual labels in the figure disambiguate the equivalent graphical notations); classes are depicted as yellow boxes, instances as green boxes. The object property 'component of', holding between attributes and qualities, is modeled as a generic 'part-of' relation [40], whereas the 'influenced-by' relation reflects DOLCE's characterization of general dependence, to highlight the strong connection between the assessment (existence) of proper internal and external characteristics and the computation of the derived trust level. Note that *objective*, *subjective*, and *objective-subjective* designate the sorts of metrics that can be predicated to each human factor (represented in Figure 1). An *objective* metric represents characteristics that are based in quantifiable and unbiased facts such as highest level of education completed. A *subjective* metric represents characteristics based in human decision-making and assumptions such as political rationality. An *objective-subjective* metric represents characteristics that are based in fact while also influenced by human decision-making such as emotional state. These metrics types are modeled as instances in HUFO: the use of meta-classes would have required OWL-Full, which is the undecidable fragment of OWL, and therefore unfit for reasoning. Consequently, we opted for modeling the three types of metrics as a collection of individual instances (range) associated to human factors classes (domain) through the object property 'has metric'.

IV. Conclusions And Future Work

In this paper we examined the effort of building a human factors ontology (HUFO) as part of a broader ontology of cyber security (CRATELO). In particular, we focused on the notion of trust, showing its ties with the inherent and external characteristics of humans interacting with computer networks. In the long term, we envision to apply HUFO in support of risk assessment and risk prioritazion in cyber operations.

The semantic model outlined in this paper is only a first, preliminary step in the process of porting a larger model of the cyber security ecosystem into a computational ontology. The holistic nature of our approach makes the task exceptionally challenging and, to the best of our knowledge, uniquely systematic in cyber security research. Despite the complex problems we are trying to solve, we're also convinced that, in the forward-looking vision of the ARL Cyber Security Collaborative Research Alliance, our approach sets a realistic and crucial milestone toward the foundation of a science of cyber security.

[4] This is the case, for instance, when a cyber analyst uses a network-based intrusion prevention system (or NIPS) to monitor and protect a given network environment from cyber attacks.

[5] http://owlgred.lumii.lv/

Acknowledgments

This research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-13-2-0045 (ARL Cyber Security CRA). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

Bibliography

[1] Technology, National Institute of Standards and, "Framework for Improving Critical Infrastructure Cybersecurity", Dept. of Commerce, NIST, Ver. 1 2014.

[2] Technology, National Institute of Standards and, "Guide for Conducting Risk Assessments", US Dept. of Commerce, NIST, Special Publication 800-30 2012.

[3] M., Hoffman, B., Kelley, T., and Henshel, D. Cains, "Trust as a Human Factor in Holistic Cyber Security Risk Assessment", in *6th International Conference on Applied Human Factors and Ergonomics (AHFE)*, 2015.

[4] World Economic, Deloitte Forum. (2015) weforum.org.[Online]. http://www3.weforum.org/docs/WEFUSA_QuantificationofCyberThreats_Report2015.pdf

[5] S., Ekelhart, A. Fenz, "Formalizing Information Security Knowledg" in *the International Symposium on Information, Computer, and Communications Security (ASIACCS '09)*, New York, pp. 183-194.

[6] D. B., Prakash, M., & Shepherd, M. Lenat, "CYC: Using Common Sense Knowledge to Overcome Brittleness and Knowledge Acquisition Bottlenecks", *Artificial Intelligence*, vol. 6, no. 4, pp. 65-85, 1985.

[7] A., Lenne, D., Debray, B. Assali, "Ontology Development for Industrial Risk Analysis", in *IEEE*

International Conference on Information & Communication Technologies: from Theory to Applications., Damascus, 2008.

[8] B. McBride, "Jena: Implementing the RDF Model and Syntax Specification", in *SemWeb*, Chicago, 2001.

[9] I. , Papadaki, K. Lykourentzou and A., Djaghloul, Y., Latour, T., Charalabis, I., Kapetanios, E. Kalliakmanis, "Ontology-based Operational Risk Management", in *13th Conference on Commerce and Enterprise Computing (CEC).*

[10] R. Dipert, "The Essential Features of an Ontology for Cyber Warfare", in *Conflict and Cooperation in Cyberspace: The Challenge to National Security*, A. Lowther and P. Yannakogeorgos, Eds.: Air Force Press (by Taylor & Francis), 2013.

[11] K. B. De Greene, *Sociotechnical systems: factors in analysis, design, and management*.: Prentice-Hall, 1973.

[12] N. Guarino, E. Bottazzi, R. Ferrario, and G. Sartor, "Open Ontology-Driven Sociotechnical Systems: Transparency as a Key for Business Resiliency", in *Information Systems: Crossroads for Organization, Management, Accounting and Engineering*, 2012, pp. 535-542.

[13] L. Obrst, P. Chase, and R. Markeloff, "Developing an Ontology of the Cyber Security Domain", in *STIDS 2012*, Fairfax, VA, 2012.

[14] MITRE. Common Malware Enumeration list. [Online]. http://cme.mitre.org/data/list.html

[15] Verizon. (2015) Data Breach Investigation Report. [Online]. http://www.verizonenterprise.com/DBIR/2015/?utm_source=pr&utm_medium=pr&utm_campaign=dbir2015

[16] L. Viljanen, "Towards an Ontology of Trust", in *Trust, Privacy, and Security in Digital Business*. Berlin-Heidelberg: Springer-Verlag, 2005, vol. 3592, pp. 175–184.

[17] O, Hafid, A. and M.A. Serhani Jules, "Bayesian network, and probabilistic ontology driven trust model for sla management of cloud services", in *3rd IEEE International Conference on Cloud Networking*, 2014.

[18] E., Dillon, T. S., Hussain, F. Chang, "Trust ontologies for e-service environments", *International Journal of Intelligent Systems*, vol. 22, pp. 519-545, 2007.

[19] N. and Matskin, M. I, pages Papeete, France, 4-9 Nov. 2007. Dokoohaki, "Structural determination of ontology-driven trust networks in semantic social institutions and ecosystems", in *International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, 2007, pp. 263–268.

[20] E. Blasch, "Trust metrics in information fusion", in *SPIE 9119 - Machine Intelligence and Bio-inspired Computation: Theory and Applications VIII,* 2014.

[21] J., Parsia, B. Goldbeck, "Trust network-based filtering of aggregated claims", *International Journal of Metadata, Semantics and Ontologies* , vol. 1, no. 1, pp. 58-65, 2006.

[22] A.C., Bertino, E. Ferrari Squicciarini, "Achieving privacy in trust negotiations with an ontology based approach", *IEEE Transactions on Dependable and Secure Computing*, vol. 3, no. 1, pp. 13-30, Jan-Mar 2006.

[23] G. A. Klein, "Recognition-primed-decision". In W.B. Rouse (Ed.), Advances of Machine-System Reserch. Greenwich, CT: JAI Press, 1989, vol. 5, pp. 47-92.

[24] G.A., Calderwood, R., & Clinton-Cirocco, A. Klein, "Rapid decision making on the fire ground", in *Human Factors Society 30th Annual Meeting,* pp. 576-580.

[25] E. Blasch, "Introduction to Level 5 Fusion: the Role of the User", in *Handbook of Multisensor Data Fusion*, D. Hall, and J. Llinas. M. E. Liggins, Ed.: CRC Press, 2008, pp. 503-535, .

[26] N. A. Giacobe, "Application of the JDL Data Fusion Process Model for Cyber Security" , 2010.

[27] E.P., Breton, R., and Valin, P. Blasch, "User Information Fusion Decision Making Analysis with the C-OODA Model", in *14th International Conference on Information Fusion*, 2011, pp. 2082-2089.

[28] D. R. Billings, K. E. Schaefer, N. Llorens, and P. A. Hancock, "What Is Trust? Defining the Construct Across Domains", in *American Psychological Association Conference (Division 21)*, Orlando, FL, 2012.

[29] A., Whitley, K. D'Amico, "The real work of computer network defense analysts," in *Workshop on Visualization for Computer Security*, 2008, pp. 19-37.

[30] A. Jøsang, J. Dezert, P.C.G. Costa, and A.-L. Jousselme. E. Blasch, "URREF self-confidence in information fusion trust", in *In 17th International Conference on Information Fusion (FUSION'2014)*, Salamanca, Spain, 2014, pp. 1-8, .

[31] D.H., and Chervany, N.L. McKnight, "Trust in Cyber-societies: Integrating the Human and Artificial Perspectives", in *Lecture Notes in Computer Science*, M. Singh, Y.-H. Tan R. Falcone, Ed. New York: Springer, 2001, pp. 27-54, .

[32] B., Moray, N. Muir, "Trust in automation: Part II. Experimental studies of trust and human Intervention in a process control simulation", *Ergonomics*, vol. 39, no. 3, pp. 429-460, 1996.

[33] D.S. Henshel, A. Alexeev, P. Rajivan M.G. Cains, "Human Actors' Roles in Holistic Cyber Security Risk Assessment", in *World Congress on Risk* , Singapore, 2015.

[34] A. Alexeev, M.G. Cains, P. Rajivan. D.S. Henshel, "Risk Parameters in Holistic Cyber Security Risk Assessment", in *World Congress on Risk* , Singapore, 2015.

[35] D. S. Henshel M.G. Cains, "Holistic Cyber Security Risk Assessment", in *Society for Risk Analysis, Denver*, Denver (CO), 2014.

[36] Oltramari, A., Cranor, L.F, Walls, R., McDaniel, P., "Building an Ontology of Cyber Security", in *STIDS 2014 (9th International Conference on Semantic Technology for Intelligence, Defense, and Security*, 2014.

[37] Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A., Schneider, L., "The WonderWeb Library of Foundational Ontologies and the DOLCE ontology," Laboratory For Applied Ontology, ISTC-CNR, Technical Report 2002.

[38] Kutz, O., Lücke,D., and Mossakowski, T., "Heterogeneously Structured Ontologies—Integration, Connection, and Refinement", in *Knowledge Representation Ontology. Workshop*, 2008, pp. 41-50.

[39] Gärdenfors, P. "Conceptual Spaces: The Geometry of Thought", p. 2004.

[40] Simons, P. "Parts: a study on ontology" , 1987.

Ontology-based Adaptive Systems of Cyber Defense

Noam Ben-Asher*‡, Alessandro Oltramari†, Robert F. Erbacher*, Cleotilde Gonzalez†
*U.S. Army Research Laboratory Adelphi, MD, USA
nbenash@us.ibm.com, robert.f.erbacher.civ@mail.mil
†Carnegie Mellon University Pittsburgh, PA, USA
aoltrama@andrew.cmu.edu, coty@cmu.edu
‡IBM T.J.Watson Research Center, Yorktown Heights, NY

***Abstract*—In this paper we outline a holistic approach for understanding and simulating human decision making in knowledge-intensive tasks. To this purpose, we integrate semantic and cognitive models in a hybrid computational architecture. The contribution of the paper is twofold: first we describe a packet-centric ontology to represent network traffic. We show how the ontology is used to describe real-world network traffic and also serve as a basis for higher level ontologies of cyber operation, threat and risk. Second, we demonstrate how the combination of the packet-centric ontology with an adaptive cognitive agent with learning capabilities, can be used to understand the human defender reasoning processes when monitoring network traffic. Through simulation experiments we evaluated the proposed hybrid computational architecture and demonstrate its ability to successfully detect malicious port scanning within legitimate network traffic. We discuss the implications of these findings for improving our understanding of the cognitive processes and knowledge requirements of the cyber defender, as well as the possible use of the hybrid architecture as a cognitively inspired decision support tool.**

I. Introduction

Disruption of computers and the loss of sensitive information through cyber-attacks are becoming a widespread threat and a critical concern for citizens, organizations, and governments. Even with recent advances in information and network security and the development of new monitoring and threat detection tools, many of the tasks performed by cyber-defenders (i.e., security analysts) remain challenging, resulting in weak and uncertain cyber-defense. The analytical capabilities of the human decision maker are needed and indispensable for the process of cyber-defense [1]. Security analysts transform network traffic data into cyber situation awareness, a high level of processing that is difficult to automate [2]. This process may be seen as analogous to the Data-Information-Knowledge-Wisdom (DIKW) hierarchical model that is central for information and knowledge management [3]. Within this context, cognition serves as the driver that governs the transitions between the different levels of information representation [4]. While there is a large body of research on technologies that detect port scanning [5], there is a limited understanding of the cognitive processes cyber security analysts use to detect port scanning and specifically how these cognitive abilities interact with and information representation. In this regard, the contribution of this paper is twofold: first we describe a packet-level ontology that represents network traffic. Second, we demonstrate how the integration of this ontology with a computational cognitive agent can be used to understand the human analyst reasoning process, which may then serve as guide to develop decision support technology for the analyst.

II. Knowledge Model

From a cyber security standpoint, variations in network traffic are the primary prompts of analyst's behavioral responses; nevertheless, full situational awareness can emerge only from a projection of observations and decisions into a more comprehensive context that includes knowledge about threat and attack types, executable defensive maneuvers, system vulnerabilities, risk mitigation and time constraints, among others. In this regard, building a rigorous model of this complex context is a key requirement for the study of human decision making in cyber security. Computational ontologies are the knowledge component in this *holistic* approach, as they can provide a machine-readable semantic representation of cyber scenarios. In virtue of their logical properties and schematic structure, ontologies can be used by automatic reasoners in dynamic tasks: in particular, in our work we apply ontology-based reasoning to a detection task, where an agent simulates a human analyst's cognitive capabilities, including the capability of using domain knowledge and temporal information to reason about perceived events [6]. To this purpose, we engineered a packet-centric ontology of network traffic, a module of a larger ontology framework called `CRATELO` [7], the suite of modular ontologies under development in the U.S. Army Research Laboratory Cyber Security Collaborative Alliance. `CRATELO` is constituted of several domain ontologies (collectively indicated as `OSCO`), integrated on the basis of `DOLCE` top level [8] extended with a security-related middle layer. These top, middle and domain level ontologies currently add up to 330 classes, connected by 162 relationships (132 object properties and 30 datatype properties) and encoded in OWL-DL. The packet-centric ontology presented in this paper, henceforth abbreviated to `PACO`, is a partition of `OSCO`[1].

Our reseach efforts in developing `CRATELO` are inspired by Obrst and colleagues's proposal of a wide-ranging ontology framework of cyber security [9], that spans from top-level, system-oriented ontologies and human factors ontologies. In

[1]`CRATELO` stands for 'Three Levels Ontology for the ARL Collaborative Research Alliance'. `OSCO` stands for ontology of cyber operations. For more details about the program see also: http://www.arl.army.mil/www/default.cfm?page=1417

this long-term endeavour, we have been working with ARL domain experts and cyber analysts to distill the necessary knowledge of the cyber domain. As the state of the art shows, a preliminary step in understanding any new domain is to produce accessible definitions and classifications of entities [10]: discussions on cyber security often begin with the difficulties created by misused terminology (such as characterizing cyber espionage as an attack). In this regard, the Joint Chiefs of Staff created a list of cyber term definitions (allegedly extended and refined for a classified version). None of these definitions, however, were formulated as an ontology. Likewise, various agencies and corporations (NIST, MITRE, Verizon) have formulated enumerations of types of malware, vulnerabilities, and exploitations. In particular MITRE, which has been very active in the field, maintains two dictionaries, CVE (Common Vulnerabilities and Exposure) and CWE (Common Weakness Enumeration), a classification of attack patterns (CAPEC - Common Attack Pattern Enumeration and Classification), and an XML-structured language to represent cyber threat information (STIX - Structure Threat Information Expression).

Despite of the important role played by these and further initiatives, the lack of a shared formal semantics make terminologies hard to define, sustain, and port into a machine-processable format: here we try to overcome these problems, embracing a holistic approach to model cyber security factors. In fact, if the ontology outlined in this paper is tailored to a packet-centric model of network traffic, it can be framed at a higher level of conceptualization by means of the integration with `CRATELO`: for instance, when modeling the behavior of a cyber analysts during an attack, packets can be seen as parts of the evidence collection process, and specific attributes of packets (e.g. internal or external IP addresses, low or high packet rate, etc.) may hint to specific intentions of the adversary (also called *anti-goals*). As mentioned at the beginning of the section, ontologies can serve as knowledge bases to agents: conversely, the dynamics of the agent's decision process and learning from experience are captured by an Instance-based Learning (IBL) cognitive model [11], which is a computational representation of the processes that guide human behavior. Next section reviews what cognitive models are, and how they can be used to study human decision making.

III. Cognitive Model

In a dynamic decision making setting, cognitive architectures, such as ACT-R [12], SOAR [13] and others, have been commonly used to provide an integrated representation of human cognition. Cognitive models, constructed using these architectures, allow for a careful examination of various cognitive processes that drive human decision making [11]. Cognitive models based on IBL theory (IBLT) focus on decision making and learning from experience in dynamic settings [11]. IBLT emerging from ACT-R, proposes a generic decision-making process that recognizes decision situations, generates instances through the interaction with the decision task, and finishes with reinforcement of the instance leading to desired outcomes. According to IBLT, the decision maker represents decision making situations as instances stored in memory. An instance is composed of three parts: (1) *situation* (S) a set of attributes representing a situation; (2) *decision* (D) that is made in the particular situation; and (3) *utility* (U) that is the experienced outcome from a decision. The IBLT decision cycle includes several stages: recognition, judgment, choice, and execution. In the *Recognition* stage, a decision maker identifies relevant attributes for a specific decision situation. *Judgment* stage determines the relevancy of past experiences (instances) in current decision making situation. The activation of instances in memory is a representation of relevancy. Activation is influenced by the recency and frequency an instance occurred in the past and the similarity between the current decision situation and the situation stored in the instance. This activation mechanism is a simplification of the mechanism originally proposed in the ACT-R architecture. Memory activation determines the probability that an instance will be retrieved from memory and participate in the next phase. In the absence of previous experiences that may be relevant to the current situation, pre-defined heuristics are triggered for decision making. In the *Choice*, the retrieved instances and their retrieval probability are used to calculate the expected utility for each of the decision options, and the option with the highest expected utility is chosen. Finally, in the *Execution*, feedback regarding the last decision is provided to the decision maker [11]. In this work, we chose IBL to model the decision making as it captures the adaptive human decision making and learning processes in dynamic environment as well as the transition between exploration and maximization.

Agents based on IBL models successfully account for human decision making and behavior in a variety of tasks. Lejarraga *et al.* [14] demonstrate that a single IBL model constructed for a specific repeated binary choice task can be generalized to different variants of repeated tasks requiring a binary decision as well as to probability learning tasks. More specifically, IBL models can reflect human behavior in simple *stimulus-response* practice and skill acquisition tasks and training. Furthermore, the experience-based learning process of an IBL model was successfully extended to include descriptive information and biases as risk aversion [15]. A pair of IBL models successfully consider the dynamics of cooperation in iterated Prisoner's Dilemma as well as reciprocity and other complex social interactions [16], [17].

IV. A Packet-Centric Network Ontology

In this section we describe the structure of `PACO`, and how it can be used to instantiate thousands of packets generated by capturing actual network traffic. As Fig. 1 shows, the class 'PacketTransmission' is considered the atomic element of a 'NetworkSession'. Intuitively, this means that without an actual exchange of packets between a source and a destination node, no network session can be deemed as properly complete. In fact, there are additional features of network sessions: for instance, when considering TCP connections, a complete handshake with SYN, SYN+ACK and ACK packets transmis-

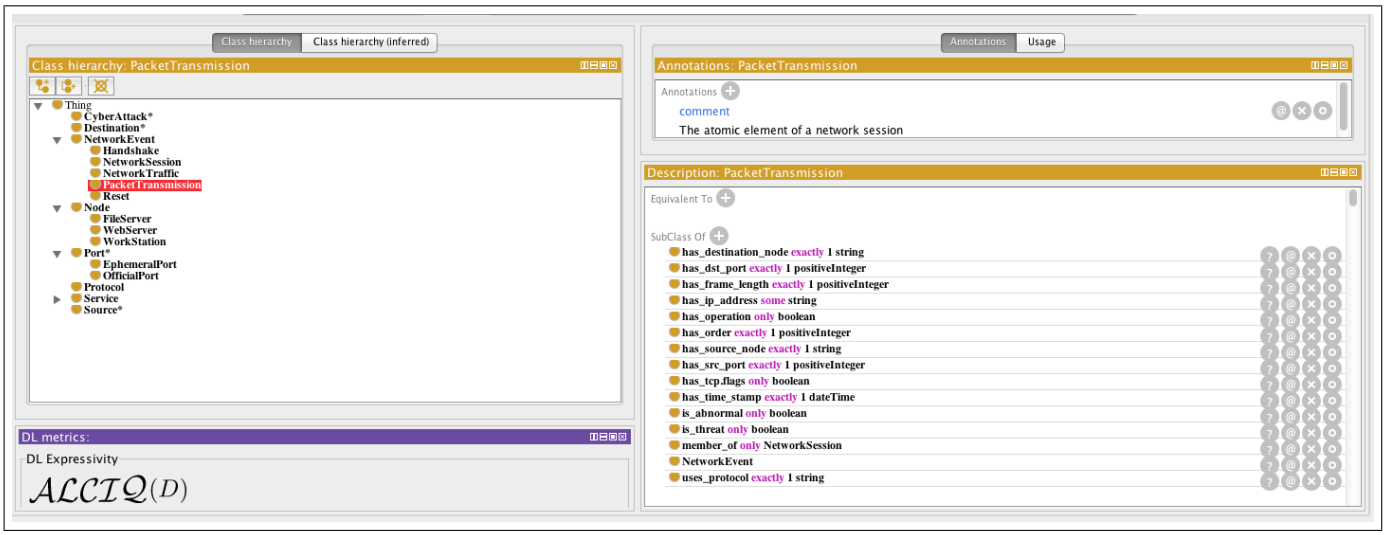


Fig. 1. A Protégé visualization of PACO. From the bottom-left corner (clockwise): 1) The DL expressivity derived by the HermiT 1.3.8 reasoner; 2) the backbone taxonomy of classes; 3) an informal definition of packet transmission as value of annotation property; 4) property restrictions.

sion is necessary to enable a packet transmission between two nodes, though this is not the case for communication protocols like UDP, where handshake dialogues are not supported. Following the actual packet transmission between the two network nodes and after the data are exchanged, a session is usually resetted (although this final stage is not essential to qualify it as complete - and session can also end due to a timeout). In summary, when a communication between a source and a destination node is established, a complete network session consists of the transmission of a unit of data from source A to destination B, and of the transmission of a unit of data from source B to destination A. From the ontological standpoint, this constraint is represented by the cardinality restriction 'min 2' on the object property 'has_member' holding between 'NetworkSession' and 'PacketTransmission' classes, respectively the domain and the range of 'has_member'.

Apart from network-specific information associated to source and destination nodes, like IP and port numbers, communication protocols, packet size, etc., we have introduced a data property 'has_time_stamp' that assigns a specific time stamp to each network event and a data property 'has_order' that binds each individual network event to its relative position in a given sequence (the first event, the second event, etc.). This twofold modeling choice provides us with a flexible model of temporal knowledge: 1) it pinpoints the discrete temporal coordinates of each event according to a universal time format (based on the XML schema specifications[2]); 2) it allows for representing and reasoning over qualitative temporal relations like 'before', 'after', and 'overlap', as defined by Allen's temporal axioms [18]). Figure 2 shows a situation where the ordinal scale of the packet is captured (i.e., the 1024th packet) but the time stamp is not represented: the reason is that the former is more appropriate than the latter for the simulation experiment reported in the next section, since the dataset was collected with a rate of about 83 packets per second. In other words, in our specific cyber scenario knowing the sequence of events is more meaningful than knowing the real time stamps from the defender's perspective, although - to be general enough - the ontology has to support both representational formats. As depicted in Fig. 2, the role of a packet in the handshake sequence can be captured by three booleans data properties, respectively 'has_tcp.flags.syn', 'has_tcp.flags.ack' and 'has_tcp.flags.reset'. In the 'PacketTrasmission1024' case, however it is unclear whether this packet represents the first stage of a handshake or is part of a port scanning [19]. This can be resolved by evaluating the properties of the proceeding packet exchange (i.e., session) between the two nodes. As the next section will show, we conducted an experiment to elicit relevant information from instantiated ontology, and make the resulting knowledge chunks available to the cognitive model of a cyber defender. This process of knowledge elicitation from PACO is driven by a set of SPARQL queries[3], properly designed to extract and present relevant information that an agent can use to decide whether a specific event is a threat or not. For instance, the query in Fig. 3 is designed to collect all the pairs of distinct source and destination ports in the dataset of network events: on the basis of the retrieved information, an analyst can gauge the volume of network traffic on a per unique port basis; moreover, Fig. 4 represents a query built to assess how many times a given source has sent a packet to a closed port. In the latter case, the returned result, around one thousand times, can be used as a clue of the maliciousness of the source: so many attempts of communication with closed ports may, in fact, suggest a port scanning attack. Note that

[2]http://www.w3.org/TR/xmlschema11-2/

[3]http://www.w3.org/TR/rdf-sparql-query/

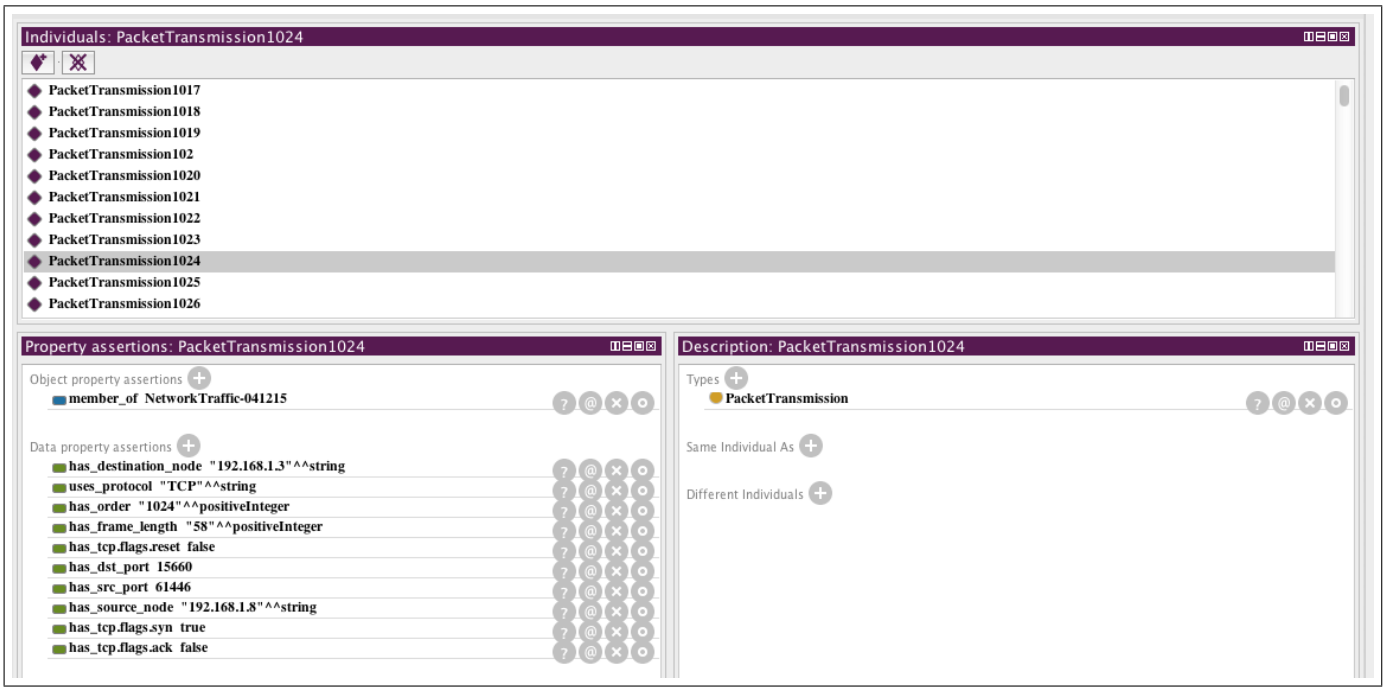


Fig. 2. A Protégé visualization of a specific instance of the 'PacketTransmission' class.

both queries have been used dynamically in the experiment described in the next section, where the goal is to replicate the analyst's incremental understanding of the considered cyber scenario.

Following a basic modeling strategy, in PACO we directly assign specific data sizes to each network event through the data property 'has_frame_length': an alternative option would have been to introduce the class 'Packet' (a subclass of 'information object' in DOLCE), and use the object property 'participation' to link 'Packet' and 'PacketTransmission', switching the domain of the data property 'has_frame_length' from PacketTransmission' to 'Packet'. At the current stage of development, representing the data contents of packet transmissions doesn't add any fundamental benefit to our modeling framework, although we don't exclude this option in the future.

Additional semantic structures of PACO concern network topology and services: for instance, every network node runs a set of services, and each service uses an official communication port and a specific protocol to establish a network session with another node. It follows that when a port is open, a service is running on a node, and if a port is closed, no services are currently running for that particular node. Thanks to the interoperability between PACO and CRATELO, services can be modeled in the context of user's actions: for instance, a system administrator can decide to start or stop an HTTP service, or access to the event log service on a server. By and large, the originality of our approach relies on the flexibility in the granularity of the representation: PACO is only a module of a more comprehensive framework that sees the detection as a socio-technical task, where packet-centric information can

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX IBLOd: <http://www.cra.psu.edu/IBLOd#>
 SELECT DISTINCT ?srcport ?dstport
  WHERE {{?event IBLOd:member IBLOd:NetworkTraffic-041215;
         IBLOd:has_src_port ?srcport;
         IBLOd:has_dst_port ?dstport;
         IBLOd:has_source_node ?s;
         IBLOd:has_destination_node ?d;
         IBLOd:has_order ?order.
          FILTER(?order >= "1"^^xsd:positiveInteger &&
                 ?order<= "4735"^^xsd:positiveInteger).}}
```

Fig. 3. A SPARQL query that returns all the distinct combinations of source and destination ports for a packets exchange sequence between two nodes.

be used by the decision maker at the cyber operation level. In principle, using CRATELO we can also model beliefs, goals and emotions of defenders and attackers, although it's beyond the scope of the current work to address these dimensions.

V. Using Hybrid Models in Cyber Defense

Next, we examine the interplay between knowledge and cognition in cyber defense by integrating the packet-centric ontology with cognitive agents who make decisions regarding the state of a network into a hybrid computational architecture. For the packet-centric knowledge-base we use PACO and the agents are computational models of the IBL theory.

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX IBLOd: <http://www.cra.psu.edu/IBLOd#>
 SELECT (COUNT(?order) AS ?numberOfACKResponses)
  WHERE  {?event IBLOd:member IBLOd:NetworkTraffic-041215;
         IBLOd:has_source_node ?sn;
         IBLOd:has_destination_node ?dn;
         IBLOd:has_tcp.flags.syn false;
         IBLOd:has_tcp.flags.ack true;
         IBLOd:has_tcp.flags.reset true;
         IBLOd:has_order ?order.
          FILTER (?order >= "1"^^xsd:positiveInteger &&
                ?order <= "4735"^^xsd:positiveInteger).}
```

Fig. 4. A SPARQL query that returns the number of times a source node sent packets to a closed port on the destination node.

A. Port Scanning Scenario

Port scanning is designed to probe network nodes for open ports. The existence of an open port can provide some indication on the availability of services. This type of information gathering can be part of a defensive or offensive operation. From the attacker's perspective, a port scan is useful for gathering relevant information for launching a successful attack and indeed most attacks are preceded by some form of scanning activity (reconnaissance), particularly vulnerability scanning [20]. Therefore, the defender will try to detect external scans while the attacker interest is to perform a scan without being detected [21].

In this work, we assume first that the attacker uses external resources to identify the attack IP address (i.e., the target). Following, the attacker identifies port ranges to scan on the specific target. These are the ports for services for which the attacker has sophisticated attacks available. We also assume, that the target is using standard ports and not randomized ports. Thus, knowing that a port is open provides an accurate indication that a service is running on the target.

B. Cognitive Models for Port Scanning Detection

To better understand the interplay between cognition and knowledge and how semantic information supports the ongoing work of the cyber defender, we developed two cognitive models for cyber defender agents. Both agents observe a situation, make decisions whether there is a scan or not, and learn from feedback and past experiences. However, the one agent operates without the knowledge based provided by PACO, while the other is querying PACO to acquire temporal information and situational awareness.

1) Experience Only Agent: To examine the interplay between information, cognition and knowledge, we initially constructed an agent using an IBL model which classifies network events based on their attributes and learns from experience only. The decision making process of this IBL agent depends on the low level network traffic information, and the agent could learn only from its own experiences without the ability to acquire knowledge by querying the ontology. The situation as observed by the agent in this condition is given by

$$S_i = \{p, sIP, dIP, SYN, ACK, RST\} \quad (1)$$

TABLE I
PAYOFF MATRIX WHICH DETERMINING THE FEEDBACK FOR AN AGENT MAKING A DECISION IN A GIVEN SITUATION

		Agent's Decision	
		Scan	No Scan
Packet Type	Scan	Hit: 10	Miss: -10
	No Scan	False alarm: -5	Correct Rejection: 5

Where p is the protocol type (e.g., TCP, HTTP) of the packet, sIP and dIP are the source and destination IP addresses of the packet. SYN, ACK and RST are 1-bit boolean flags that indicate on the state of a connection.

The agent observed a situation S_i and made a decision which corresponds to classifying a packet as being part of a scan or not. This decision process involves retrieving relevant instances (i.e., past experiences) from the agent's memory, computing retrieval probability for each of the instances and, choosing the decision option that yields the highest expected utility, based on the previous decisions recorded in the instances. The process of choosing the option with the highest expected utility is influenced by the recency and frequency of past experiences, memory decay (d) and a noise parameter for capturing the variability in memory activation (σ) [11].

After making a decision, the agent received a utility feedback, representing the outcome of the decision in a given situation. The experienced utility (i.e., payoff) is determined based on the payoff matrix illustrated in Table I. The payoff that an agent receives following a decision, is determined by the accuracy of the decision, based on the ground truth, detailed in section V-C. The payoffs in the matrix emphasize the positive and negative utilities from hits and misses over correct rejections and false alarms.

2) Semantic Information and Experience Agent: In contrast to the previous agent model, this agent can send SPARQL queries to the PACO ontology, that provides specific knowledge of the scenario, temporal information and augmented situational awareness. As such, this model observes the same situation as the *Experience Only* agent: however, instead of using this information to make a decision, the agent uses the information to generate queries (which, in turn, provides richer information). Using PACO, the agent can generalize from and reason about the characteristics of a sequence of packets transferred from one network node to the other. Therefore, the situation observed by the agent consist of the outputs from multiple queries regarding the *conversation* between two specific IP addresses, where one is the source and the other is the destination. The situation for any packet, transmitted between a source and a destination IP addresses, is given by

$$S_i = \{p, sPorts, dPorts, avgSYN, avgACK\text{-}RST\} \quad (2)$$

Where the attributes of the situation represent properties of a communication between source and destination IPs, using protocol p. The communication consists of a sequence of packets exchanged between the two network nodes up to the current packet. Thus, the agent can examine each packet within

the context of a sequence. Given the source IP of the current package, attribute $sPorts$ indicates on the average number of ports in the source node that sent packets to the destination node. Similarly, attribute $dPorts$ indicates how many ports in the destination node received packets from the source. The attribute $avgSYN$ describes the average ratio between SYN packets and normal traffic recived from the source of the packet. Attribute $avgACK\text{-}RST$ provides complementary information, the average ratio of between ACK-RST packets and normal traffic the destination sent back to the source. This type of answer indicates that the packet was sent to a closed port (i.e., a port that is not used by any service on the target node).

Based on the set of attributes described above, the *Semantic information and Experience* agent classified packets. The *Semantic information and Experience* agent received feedback for these decisions using the same payoff matrix as the *Experience Only* agent.

C. Simulation Experiment

We evaluated the differences between the two agent models through simulation experiment. In the experiment, agents classified the packets captured from the traffic in a small network with 16 nodes (i.e., unique IP addresses). The captured communication between the network nodes included 4735 packets. The nodes used several types of protocols to exchange packets, for example SMB and SSL. However, the majority of the traffic (99.56%) used the TCP protocol. Within this network, the adversary was located in a node with the IP address of 192.168.1.8. The adversary used a specific port to scan the 1000 common ports of the target node (192.168.1.3) using Nmap defaults [22]. This information was not provided to the agents and served as the ground truth for evaluating the detection performance of the agents and providing them with feedback. The captured network traffic was converted into an XML data structure that was used to populate `PACO` and the *Semantic Information and Experience* agent could then query using SPARQL. The output of the SPARQL queries served as the attributes of a situation as described in Eq. 2.

The values of the free parameters across the two agents were kept the same, with $d = 1.5$ for memory decay and $\sigma = .25$ for noise. These values are considered to be the ACT-R defaults and are commonly used for IBL models as well [23]. Each agent classified the 4735 packets and received feedback following each decision, and this was repeated for 20 iterations.

To compare the performance of the *Experience Only* agent with the *Semantic Information and Experience* agent we used the following metrics:

1) **Correct packet classification** indicates on the proportion of packets classified correctly as being a *Scan* or *No Scan* packet.
2) **Correct detection of scanning sequence** indicates on the proportion of conversations between two IPs that were correctly classified as scans.

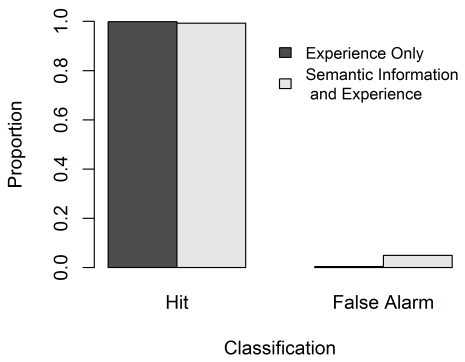


Fig. 5. Proportions of hits and false alarms for the two agents.

3) **Learned classification rule** indicates on the decision rule the agents constructed from the repeated experiences.

VI. Results

In this section, we show our experimental results and analyze the observed trends based on the performance comparison of the two modeling approaches.

Correct packet classification When analyzing the ability of the agents to classify correctly a scan packet, and as seen in Fig. 5, we find that the *Experience Only* agent (mean=.999, SD=0) and the *Semantic Information and Experience* agent (mean=.992, SD=.002) performed similarly with a minor advantage to the *Experience Only* agent, $t(38)=.387$, p=ns. However, the *Semantic Information and Experience* agent (mean=.050, SD=.077) generated a significantly higher number of false alerts compared to the *Experience Only* agent (mean=.004, SD=0), $t(38)=2.661$, $p=.011$.

Correct detection of scanning sequence utilizes the classification of a packet as belonging to a scan or to normal traffic between two network nodes. This high level decision aims to answer the question whether network node A is scanning network node B. With respect to this question, if the network traffic from node A to node B includes one or more packets that are classified as scan packets, then node A is scanning node B. When analyzing the ability of the two agents to answer the question whether node A is scanning node B, both agents detected that the adversary was scanning a specific network node (i.e., 192.168.1.8 $\overset{SYNscan}{\rightarrow}$ 192.168.1.3). However, the *Experience Only* agent detected on average additional 2.3 out of 22 sequences between network nodes as scans (i.e., 10% false scans), while the decisions of the *Semantic Information and Experience* agent yielded 0 false classification of packet sequences. Despite the higher false classification rate of individual packets the *Semantic Information and Experience* agent had, all these false classified packets belonged to the responses of the scanned node (ACK packets) to the adversary scan (i.e., 192.168.1.3 $\overset{ACKresponse}{\rightarrow}$ 192.168.1.8).

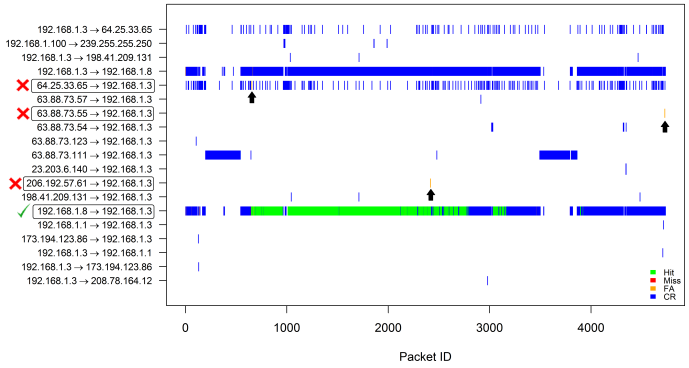


Fig. 6. Detection outcomes of the *Experience Only* agent during a single iteration with black arrows highlighting false classification of packets, red cross marks indicating on sequences of packets that were incorrectly classified as scans and green check mark for correct classification.

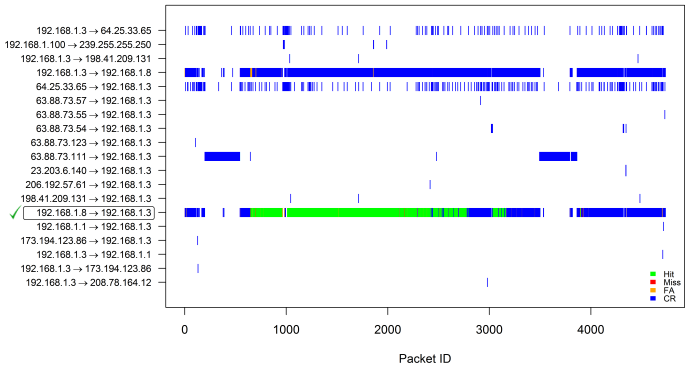


Fig. 7. Detection outcomes of the *Semantic Information and Experience* agent during a single iteration with green check mark indicating on the correct classification of a packet sequence.

Figures 6 and 7 illustrate the interplay between packet classification and sequence classification. In both figures, we present the same network sequences and how the packets were classified by each agent. As seen in Fig. 6, the *Experience Only* agent generated a very low number of false alerts (highlighted by arrows). However, these packets corresponding to these alarms were distributed across multiple sequences. As a result, the entire sequence was classified as a scan. On the other hand, and as seen in Fig. 7, the false alerts generated by the *Semantic Information and Experience* agent were all part of the communication between the scanned node to the source of the scan. Note that both agents were able to separate between legitimate traffic between 192.168.1.8 and 192.168.1.3 that was not part of the scan and used UDP and other protocols.

The **Learned classification rule** used by each agent can be formalized by examining the instances stored in the memory of each agent, and their activation. The combination of attributes and decision in highly activated instances represent beliefs regarding a relationship between a situation and the appropriate decision. The decision rule formulated by the *Experience only* agent was that any TCP packet with a SYN flag is part of an ongoing scan between the source of the packet and its destination. This rule yields high accuracy in detecting scan packets as all the scan packets had a SYN flag. However, packets with SYN flag are also part of legitimate handshake between network node and for that reason the *Experience Only* agent detected a higher proportion of packets sequences as scans. In contrast, the *Semantic Information and Experience* agent observed the temporal properties of a packet sequence. The decision rule formulated by this agent suggests that a scan packet uses TCP protocol and is part of a sequence of packets in which the source node is using a low number of ports to send packets to a high number of destination ports and the average number of SYN packets sent to a port is very close to 1. In addition, the rule constructed by the *Semantic Information and Experience* agent indicates the based on experience, the target node of the packet is very likely to respond to the current packet with a ACK-RST packet, indicating that the destination of the packets coming from the source node tends to be a closed port.

VII. Discussion and Conclusion

Analytical capabilities of the human decision maker are needed and are indispensable when ensuring the security of any cyber infrastructure. It is the human abilities to integrate information, to reason, to learn and to quickly adjust to changes that make such significant contribution to cyber security. The understanding of these processes relies on our integration of knowledge from human cognitive theories and knowledge-based technologies. In this study we propose an architecture to combine cognitive models and ontologies in the domain of cyber defense.

We developed a packet-centric ontology PACO which allows us to represent and capture the atomic elements of network communication, i.e., packets and sequences of packets. PACO serves as the basis for more holistic semantic representations of cyber operation, cyber assets, threats and risks, available through CRATELO. We also developed an IBL cognitive model capable of accessing the information in PACO and using it when detecting adversarial port scan. When making decisions, the ability of the IBL agent to access PACO and retrieve information improved its performance, compared to the same IBL agent that did not utilize PACO. We show that when answering the questions 'Is IP A scanning IP B?', an agent with access to a packet-centric ontology delivers a much lower false alerts rate and by that show superior performance. Overall, the access to semantic information allowed the agent to acquire better situation awareness by incorporating summarized information into the decision making process. PACO extended the agents ability to inspect temporal relationship between a packet sent from a specific source and previous replays of packet's destination to communication coming from that source. Such reasoning requires a representation of a source and a destination, as well as the ability to switch between these roles in order to observe the response patterns.

The agents explored rules in the form of *IF* a situation *THEN* a decision, and learned which rule maximizes their

payoff. While the attributes of the situation part are influenced by the availability of information, the cutoff values of the attributes were learned from experience. Furthermore, the decision rule that the agent with access to a packet-centric ontology learned from experience is valid and useful beyond the limited scope of the network scenario we used in the study.

However, the existence of knowledge is a precondition rather than a guarantee for improvement: correctly querying the information is the key for the major improvement. In the process of modeling, we used domain experts to construct the queries that aggregate and retrieve information. By using cognitive agent we were able to test different queries and combinations of attributes, to identify representations that facilitate the decision making process of a network defender.

While PACO has the potential of representing packet level information for complex and diverse network communication, the current cognitive model was developed to accommodate a simplistic network scenario. Port scanning can take many forms (vertical and horizontal scans), can use different protocols and can be highly distributed over time (i.e., low-and-slow scan). Therefore, although we used a high fidelity network traffic, future research should scale up the volume of traffic as well as the complexity of the network scan. Such additions will likely challenge the cognitive agent. However, providing the agent access to the middle and high levels of CRATELO might be the key component for the agent's success in more complex and challenging tasks. The benefit of pairing cognitive agents and ontologies goes beyond the ability to gauge into the decision making process of the human analyst. Such combination can serve as an initial step towards the development of cognitively inspired decision aid tool for automating some tasks that are currently performed by human analyst.

ACKNOWLEDGMENT

This research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-13-2-0045 (ARL Cyber Security CRA). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

REFERENCES

[1] C. Gonzalez, N. Ben-Asher, A. Oltramari, and C. Lebiere, "Cognition and technology," in *Cyber Defense and Situational Awareness*, ser. Advances in Information Security, A. Kott, C. Wang, and R. F. Erbacher, Eds. Springer International Publishing, 2014, vol. 62, pp. 93–117.

[2] A. DAmico and K. Whitley, "The real work of computer network defense analysts," in *VizSEC 2007*. Springer, 2008, pp. 19–37.

[3] J. E. Rowley, "The wisdom hierarchy: representations of the dikw hierarchy," *Journal of information science*, 2007.

[4] N. Ben-Asher and C. Gonzalez, "Effects of cyber security knowledge on attack detection," *Computers in Human Behavior*, vol. 48, pp. 51–61, 2015.

[5] C. B. Lee, C. Roedel, and E. Silenok, "Detection and characterization of port scan attacks," *Technical report, Univeristy of California, Department of Computer Science and Engineering*, 2003.

[6] A. Oltramari, N. Ben-Asher, L. Cranor, L. Bauer, and N. Christin, "General requirements of a hybrid-modeling framework for cyber security," in *Military Communications Conference (MILCOM)*. IEEE, 2014, pp. 129–135.

[7] A. Oltramari, L. F. Cranor, R. J. Walls, and P. McDaniel, "Building an ontology of cyber security," in *9th International Conference on Semantic Technologies for Defense, Intelligence and Security (STIDS)*, 2014, pp. 54–61.

[8] C. Masolo, S. Borgo, A. Gangemi, N. Guarino, A. Oltramari, R. Oltramari, L. Schneider, L. P. Istc-cnr, and I. Horrocks, "Wonderweb deliverable d17. the wonderweb library of foundational ontologies and the dolce ontology," 2002.

[9] L. Obrst, P. Chase, and R. Markeloff, "Developing an ontology of the cyber security domain." in *7th International Conference on Semantic Technologies for Defense, Intelligence and Security (STIDS)*, 2012, pp. 49–56.

[10] D. A. Mundie and D. M. McIntire, "The mal: A malware analysis lexicon," DTIC Document, Tech. Rep., 2013.

[11] C. Gonzalez, J. F. Lerch, and C. Lebiere, "Instance-based learning in dynamic decision making," *Cognitive Science*, vol. 27, no. 4, pp. 591–635, 2003.

[12] J. R. Anderson and C. Lebiere, *The atomic components of thought*. Lawrence Erlbaum Associates Publishers, 1998.

[13] J. E. Laird, A. Newell, and P. S. Rosenbloom, "Soar: An architecture for general intelligence," *Artificial intelligence*, vol. 33, no. 1, pp. 1–64, 1987.

[14] T. Lejarraga, V. Dutt, and C. Gonzalez, "Instance-based learning: A general model of repeated binary choice," *Journal of Behavioral Decision Making*, vol. 25, no. 2, pp. 143–153, 2012.

[15] N. Ben-Asher, V. Dutt, and C. Gonzalez, "Accounting for the integration of descriptive and experiential information in a repeated prisoner's dilemma using an instance-based learning model," in *22th Behavior Representation in Modeling & Simulation (BRiMS) Conference*, 2013, pp. 11–14.

[16] C. Gonzalez, N. Ben-Asher, J. M. Martin, and V. Dutt, "A cognitive model of dynamic cooperation with varied interdependency information," *Cognitive science*, 2014.

[17] C. Gonzalez and N. Ben-Asher, "Learning to cooperate in the prisoners dilemma: Robustness of predictions of an instance-based learning model," in *35th annual meeting of the Cognitive Science Society (CogSci 2014)*, 2014, pp. 2287–2292.

[18] J. F. Allen, "Maintaining knowledge about temporal intervals," *Communications of the ACM*, vol. 26, no. 11, pp. 832–843, 1983.

[19] M. De Vivo, E. Carrasco, G. Isern, and G. O. de Vivo, "A review of port scanning techniques," *ACM SIGCOMM Computer Communication Review*, vol. 29, no. 2, pp. 41–48, 1999.

[20] E. M. Hutchins, M. J. Cloppert, and R. M. Amin, "Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains," *Leading Issues in Information Warfare & Security Research*, vol. 1, p. 80, 2011.

[21] M. H. Bhuyan, D. Bhattacharyya, and J. K. Kalita, "Surveying port scans and their detection methodologies," *The Computer Journal*, vol. 10, pp. 1565–1581, 2011.

[22] Nmap network mapper. [Online]. Available: https://nmap.org/

[23] N. Ben-Asher, J.-H. Cho, and S. Adalı, "Cognitive leadership framework using instance-based learning," in *24th Conference on Behavior Representation in Modeling and Simulation (BRiMS)*, March 2015.

Enabling New Technologies for Cyber Security Defense with the ICAS Cyber Security Ontology

Malek Ben Salem
Accenture Technology Labs
Arlington, VA
malek.ben.salem@accenture.com

Chris Wacek
Invincea Labs
Arlington, VA
christopher.wacek@invincea.com

***Abstract*—Incident response teams that are charged with breach discovery and containment face several challenges, the most important of which is access to pertinent data. Our TAPIO (Targeted Attack Premonition using Integrated Operational data) tool is designed to solve this problem by automatically extracting data from across the enterprise into a fully linked semantic graph and making it accessible in real time. Automated data translation reduces the costs to deploy and extend the system, while presenting data as a linked graph gives analysts a powerful tool for rapidly exploring the causes and effects of a particular event. At the heart of this tool is a cyber security ontology that is specially constructed to enable the TAPIO tool to automatically ingest data from a wide range of data sources, and which provides semantic relationships across the landscape of an enterprise network. In this paper we present this ontology, describe some of the decisions made during its development, and outline how it enables automated mapping technologies of the TAPIO system.**

***Index Terms*—cyber security, ontology, cyber analysis, semantic technologies, ontology patterns, forensic analysis**

I. Introduction

Cyber security suffers from a critical talent shortage [10]. According to Cisco's 2014 Annual Security Report, over a million cyber security positions went unfilled worldwide [7], and companies are constantly facing challenges recruiting and retaining skilled security professionals. As targeted attacks on computer networks have become more prevalent, the need for tools that allow analysts to effectively hunt for malicious activity has exploded. The existing solutions in this space - security event information management (SIEM) systems - seek to ingest and aggregate all of the logs or events that might be of interest to a defender, generally storing each according to its source schema. Adding new data sources to these systems requires engineering new "connector" to be able to source and parse the data. Tracking the broader context is left to the skills and experience of the analyst, and correlation between events is frequently a manual and mental process [14].

Our prototype TAPIO system seeks to upend this paradigm by ingesting events and the context around them, and leveraging our cyber security ontology as a guide for transforming that information into a fully linked semantic graph. By producing a fully linked semantic graph, we change the task of correlating and connecting events and actions into a simple traversal of the data graph. The aim is to reduce the specialized security knowledge required for an analyst to be effective at understanding and evaluating the threat presented by a given alert, and to enable them to hunt for threats intuitively.

To enable this, we leverage semantic web technologies such as Resource Description Framework (RDF) and Web Ontology Language (OWL). TAPIO agents running locally on devices across the network 1) automatically interpret the data sources and translate the unknown schemata of each source to a common ontology, 2) link the data, and 3) store it in a local RDF store. The information is then accessible through a real-time query capability to security analysts. At the heart of the capabilities listed above lies the cyber security ontology, which we refer to as the Integrated Cyber Analysis System (ICAS) ontology.

Typically, the primary goal of any ontology is to enable knowledge sharing and re-use. An ontology can be considered as a conceptualizing of a certain specified domain at some level of abstraction. By creating a common lexicon around *what exists* in the domain, an ontology enables the creation of tools and components that are capable of storing and exchanging information about that domain. There are many existing abstractions of computer and network operations - one need only lightly review the veritable alphabet soup of ontologies and taxonomies out there to make that clear [15, 25, 23, 24, 22] .

However, in the case of TAPIO the ontology has a second purpose - to support automated data translation by the system, and in doing so lower the barriers to providing access to new data sources to security analysts. This dual purpose led us to construct the ICAS ontology presented in this paper, building upon the great body of work in knowledge representation that has been developed in previous ontologies and taxonomies. In this paper we:

- present the ICAS cyber security ontology that models security related elements and the semantic relationships between them;
- identify several necessary considerations when developing an ontology with the dual goal of providing a data schema and enabling automated translation to it; and
- provide an overview of how the TAPIO system uses the ontology to support its data mapping.

The remainder of this paper is organized as follows. In Section II, we review related work. We then present the main ontology requirements in Section III. We present the

ontology and the design considerations that were involved in constructing it in Section IV. In Section V, we outline how the ontology enables the cyber defense TAPIO tool. We focus in particular on how the ontology enables the extraction of data schemata from previously unseen data, thus allowing the interpretation of that data and its mapping into a graph database. We discuss some of the challenges and limitations of using and maintaining the ontology in practice as well as some directions of our future work in Section VI.

II. Related Work

In this section, we review prior work related to developing our cyber security ontology. We start by covering previous general cyber security ontologies and security standards relevant to our work. Then we review some event models that we leveraged in this work.

A. Security Standards

The US government launched the Information Security Automation Program (ISAP) to develop security standards in order to enable the automation of technical security operations. Several standards have been developed including the Common Vulnerabilities and Exposures (CVE), Common Configuration Enumeration (CCE), Common Weakness Enumeration (CWE), Common Vulnerability Scoring System (CVSS), the Extensible Configuration Checklist Description Format (XCCDF), the Open Vulnerability and Assessment Language (OVAL), and the Common Attack Pattern Enumeration and Configuration (CAPEC). These standards are for the most part taxonomies, captured in XML format. Therefore, they fail to capture the semantics and relationships connecting them, thus making them not very useful for automated vulnerability management. To overcome this shortcoming, Wang and Gui leveraged CVE, CWE, CVSS and CAPEC to build an ontology for vulnerability management (OVM)[27]. The ontology allows the user to compare similar products based on their vulnerability information.

B. General Security Ontologies

There have been a few early attempts to build a cyber security ontology.

A few general security ontologies have been proposed. Herzog's Ontology [9] had 463 security-related concepts, and Fenz's ontology [8] had about 635 security-related concepts. The latter focused on threats targeting hardware assets rather than information assets. The Navy Research Lab's security ontology [11] had only 75 security concepts. A more detailed comparison of these ontolgies is available in [20].

The main weakness of these ontologies is that they focus more on objects rather than on events. Capturing events is needed to reconstruct the timeline of a cyber attack. The existing general cyber security ontologies – while they do cover many of the cyber security-related concepts – fail to provide a framework for capturing cyber and security events and the relationships between them. However, exploring the connections between events is important to improve situational awareness during and after a cyber attack, and is therefore paramount to any digital forensic investigation or tactical cyber defense.

Obrst et al. presented an overall architecture for a cyber security ontology that focused on malware in their seminal "Developing an Ontology for the Cyber Security Domain" paper [16]. The architecture included the foundational ontologies, which they referred to as the upper ontologies. These constitute the domain-independent ontologies describing concepts such as *entities* and *collections*. Mid-level ontologies in their architecture are ontologies that make assertions that span multiple domains, such as OWL-Time. At the lowest level of the architecture are the domain ontologies which represent domain-specific concepts. The ontology described in this work follows the same tiered architecture.

Another cyber security ontology was developed by Oltramari et al. [17]. The ontology, which is still a work in progress, aims to cover operational cyber defense by addressing the human factor besides the technological spectrum in complex cyber defence **operations**. The ontology focuses on cyber defender operations, and is complementary to our ontology, which focuses on attack discovery and contextualization.

C. Event Modeling

In [21], Scherp et al. presented a formal representation of events that enables capturing and representing human experience when dealing with various situations outside the cyber context. They developed several ontology design patterns for event correlation, causation and documentation. The latter is used to capture additional documentation entered by humans as they intervene, experience and interpret while being involved in these events.

Recently, Chabot et al. proposed an approach for reconstructing the timeline of events for digital forensic investigations [5, 6]. They developed an event model designed specifically for digital forensic investigations by linking events to their digital footprints. For example, an authentication event is linked to the authentication log entry, which constitutes the footprint of the event. This event model allows not only for the temporal correlation of events, but also for event subject and object correlation, as well as rule-based correlations. The latter can capture cause-effect relationships between events that can be predefined by subject matter experts.

There is a trade-off between capturing more semantics about events to provide richer event contextualization and the size of the triple store where this information is stored. In this work, we opted for the more light-weight event model described in [21], applied to the cyber environment, in order to limit the size of the triple stores at the endpoint devices and therefore speed up running queries against these stores. This tradeoff becomes even more important for sensors and smaller devices with limited memory and computational power, as we deploy our integrated cyber analysis systems on IoT (Internet of Things) networks and push more cyber analysis to the edge of the network.

III. Ontology Requirements

In this section, we introduce the cyber security ontology and start by listing the key ontology requirements elicited during the ontology design process:

- Representation of any entity (physical/cyber) that needs to be deployed and managed in an enterprise's IT network: One of the main goals of our ontology is to map a complete picture of the IT environment for the security analyst and to provide general coverage of the cyber security and digital forensics domains. The ontology should encompass all such entities that would be involved in an incident response action or digital forensic investigation. There are well defined ontologies in various domains, which we re-use and integrate into the ICAS ontology. These include the Ontology for Vulnerability Management described above [27]. We have also developed an extensive set of domain ontologies, including the memory artifacts ontology and the registry ontology.
- Representation of lower level events: Beyond capturing the physical and cyber entities, the ontology should capture cyber security events and digital forensics/evidence such as login sessions, software installations, etc.
- Temporal and local contextualization of events: The significance of cyber events depends on their local and temporal context. The chronological order of (security) events may suggest underlying connections between them, such as two consecutive authentications from remote locations that are geographically widely separated occurring within a short period of time. So to correlate these events and draw connections between them, the ontology must support comparing their temporal dimensions.
- Abstraction of higher level events: Cyber events are interdependent and could be correlated/aggregated to higher-level events; For instance, several port scanning events could be abstracted into the reconnaissance step of an attack.
- Abstraction of cyber security attacks and attacker's TTPs (Techniques, Tactics and Procedures): Attack progression interpretation requires reconstructing the timeline of the events relevant to the attack, and reasoning on temporal events. It also capturing general concepts related to security events, such as attack patterns.
- Annotation of extracted facts (relations between entities) with confidence values that reflect the trust in the correctness of the statement: Uncertainty relates to events based on missing or partial sources, which can lead to dealing with unknown information or contradictory facts, such as the example of the two authentication events from two different places within a short window of time.
- Removal of wrong/invalidated statements in order to obtain a high quality knowledge base: Events and entities need to be annotated with validity times and confidence scores, so that (temporal) consistency constraints may be checked and inconsistencies eliminated from the knowledge base. A high-quality knowledge base is needed for the reasoning engine.

IV. ICAS Ontology Design and Development

Our ontology is based on the foundational ontology DOLCE + DnS Ultralite (DUL) [19]. DUL is a lightweight version of the DOLCE [1] + DnS [18] ontology. It provides the upper level concepts that form the basis of interoperability between lower-level ontologies. So we adopted DUL's design decision which distinguishes between *events* and *objects*, where *events* unfold over time and *objects* unfold over space.

One way of thinking about this is that *objects* are elements that exist through time - the same process can be observed at many different instants in time - while *events* relate those *objects* at a specific moment in time.

The ICAS ontology is organized as a collection of sub-ontologies that capture specific conceptual areas. At present there are 30 sub-ontologies ranging from the highly specific *Memory Artifacts* which models the types of elements that might be found in a memory dump, to the much more general *Process* ontology which describes general information about running processes. The sub-ontologies are generally an organizational construct - object properties create regular connections between them.

In keeping with the intention to model all security relevant elements of an enterprise network, the ICAS ontology models network concepts in addition to the host-level observables described above, although networking information is much more frequently expressed via *events* than via *objects*. It does not attempt to replicate network traffic information at the most granular level - a task better suited to formats like NetFlow.

There are many ways to represent concepts as *objects* within an ontology, and the representation used really depends on the intended use of the ontology. For instance, a file can be represented as a single entity with a path, size, and a hash value. Alternately, an ontology can mirror the file system model where file contents are represented distinctly from its location in the file system.

In developing the *object* representations in the ICAS ontology we followed the following guidelines:

- represent each concept as close to reality as possible, without requiring platform specific distinctions
- represent the state of the world, not the rules of the world

In the case of files and file paths the first guideline led us to represent them separately, since this makes it possible to record observations of file paths without requiring knowledge of the underlying file, and permits the contents of a file to be linked from multiple different file paths (as is the case in the real world).

Windows Registry keys exemplify the other side of the first guideline. Registry Keys contain configuration information for Windows, but other platforms don't use a registry in the same way. However, all platforms have some variation on a key-value configuration store accessible by paths, whether registry paths or file system paths. As such, the ICAS ontology defines a generic `ConfigurationFileName` object that is used to represent Windows Registry keys, OS X

`.plist` files, and Linux `/etc` files. Values keys within the registry or within the configuration files can be represented with `ConfigurationKey` objects attached to the `ConfigurationFileName` object.

The second guideline has two goals: increased flexibility and reduced complexity. Consider the representation of file paths in a file system. One could model the directory structure as a complete sequence of related objects (i.e. `C:\Windows\System32` is three `Directory` objects related via the `contains` relationship, and the entire file system could be walked in the same way that one can traverse a directory tree. Alternately, this could be represented as a single `Directory` object with the entire path stored as a Datatype Property.

We choose the latter method to represent file system objects (and other concepts with hierarchical relationships such as IP networks), for several reasons. The first is that it vastly reduces the complexity of object representations, which in turn reduces the complexity of data translation and data querying. When translating data into the ontology, the former method requires a specialized processor to recognize file paths and translate them into a sequence of hierarchically related objects; similarly when composing queries, literal file paths must be decomposed into a sequence of directory objects for the query engine. The translations on each side are specialized and platform specific, which introduces opportunities for error.

The other reason is that the ICAS ontology is intended to support a tool to represent data for analysts, and human analysts are very capable of applying their own understanding to the data presented to them. For instance, that shared prefixes in file paths indicate shared hierarchical structure is commonly known. A search for all files in a directory can easily be composed by applying the analysts own understanding using a substring filter. This means that the analyst can apply their intuition easily and is not forced into understanding a rigid structure that might differ from their interpretation.

Combined, these two guidelines provide a flexible, powerful framework for object representation in the ontology that enables describing entities observed in the world, without attempting to describe the specifics of all the protocols within that world.

To make the cyber security ontology enable attack discovery and security analysis, it was critical to use a robust methodology for representing *events* and the relationships between them. Exploring connections between events is important to any cyber investigation. This requires knowledge about what events preceded, co-occurred with or succeeded others. It may require information about the time span of each event and which events occurred in the same time window. We model events as instants with a corresponding duration in time according to the W3C OWL Time ontology, which enables considering them as a sequence within a `Timeline` according to the model proposed in [26].

V. Applying the Ontology

The ICAS ontology described to this point is designed to provide an abstraction of the operation of a computer network, straddling the traditional demarcation lines between the host and network-attached devices with a focus on how elements inter-relate.

When cyber security analysts get access to the semantic relationships between information, this transforms the investigative task by natively supporting the natural human inclination to connect information. Instead of having to mentally track potentially relevant pieces of information and identify relationships on the fly, their mental resources are freed up to focus on the actual task: determining whether or not a pattern actually looks representative of malicious behavior.

A. TAPIO Overview

Our prototype TAPIO system is designed to enable this transformation by performing *ontology-guided data translation*. We describe this as *ontology-guided* because none of the techniques employed by TAPIO are specific to the cyber security domain; instead it draws information used for mapping directly from what is encoded in the ontology itself. An ontology for a different domain would enable TAPIO to translate data sourced for that domain.

In the cyber security setting, TAPIO searches for information on target systems and translates that information into linked data represented according to the ontology, without the use of handwritten rules or parsers. In this it effectively operates similarly to a database "view", enabling analysts to query linked data from across the network without knowledge of the underlying source. TAPIO consists of several components: native data sources, a data translation layer, RDF base storage, and a SPARQL query layer. The data sources are capable of retrieving data from a variety of locations, including host-based commands and APIs, remote HTTP and SQL connections, and network observation. The data translation layer accepts records from the data sources and converts them into RDF triples which are stored by the storage and the query layers.

TAPIO uses the ICAS ontology as both data schema and as an information source for automated data translation. The system stores all data in RDF triples that represent objects and properties as described in the ontology. This enables us to use the powerful capabilities of the SPARQL query language natively, where the ontology doubles as the definition of the schema against which queries can be executed.

The TAPIO system also provides feedback to the query interface about what data has been *realized*. That is, while the ICAS ontology describes the set of information that can be searched, only a subset of those objects and properties are likely to have been observed. We describe an observed ontology property or object as being *realized*. This feature means that analysts do not lose time by searching for data that will never be translated.

B. The Schema Translation Problem

One of the biggest issues for computer systems designed to provide data querying capabilities is that any data must be normalized and translated to fit the schema in use. This is true whether one is talking about a relational database schema or a full RDFS/OWL ontology such as the one used by TAPIO, and the growth in popularity of data federation and data warehousing has resulted in a variety of research efforts in this area [13, 3, 4, 2]. Much of the research focuses on translating schemata from one representation to another, e.g., mapping an XML representation of a running process into a relational database schema for that running process.

While we leverage many of the intuitions and suggestions proposed in this body of research, we suggest that the problem TAPIO seeks to address is somewhat distinct in that we seek to find a mapping from multiple *unknown* schemas to our ICAS ontology. Essentially we attempt to concurrently perform schema detection and schema translation by taking data from a source (which has a schema although we do not know what it is) and mapping it into subgraphs that are valid according to the ontology.

1) Assumptions and Resources: We make a number of reasonable assumptions about the data that we seek to translate:

1. Input data consists of a stream of *records*, where each record contains an arbitrary number of key-value pairs.
2. Key-value pairs that occur in a record together are related.
3. Keys are textual and convey some meaning related to the semantics of the data (i.e. they are not totally random).

At first blush, the first assumption does not seem reasonable; data is rarely structured in key-value form. However, we argue that almost any input data source, whether structured or unstructured can be transformed into sets of key-value pairs through techniques such as structure detection or named-entity recognition labeling. Fig. 1 shows how this might work for a common authentication log file. The TAPIO system attempts a variety of pre-processing steps to attempt to determine this structure. In the case of an authentication log containing lines like the one shown in Fig. 1, these techniques include a two pass-approach that searches for different *formats* within the log file using an unsupervised n-gram based clustering approach, and then attempts to identify *fields* within each format by observing which segments appear to be constants and which have high variance within the cluster. The details of this and the other techniques are not included here for space reasons.

The second assumption is reasonable so long as the source record is well-formed; it would be an exceedingly strange data source in which data contained within a given record was unrelated. This assumption is key to our ability to perform multi-origin mapping; we use it as a constraint on the candidate mappings we select by arguing that if candidate mappings cannot be easily related, then they are not likely to be correct because the underlying data is not related. At the same time, the processes which extract structured data from each data source are themselves heuristic and could inadvertently identify key-value pairs that do not represent real data; for instance, the word "user" in Fig. 1 is not a data field. To accomodate this reality, the TAPIO system assumes that keys within a record are related, but does not require that all identified keys are *used* in a mapping.

The third assumption is not a strict requirement, so long the same data field uses the same key over a sequence of input records. However, several of the techniques discussed in Section V-B2 operate on natural language and thus are improved if the assumption holds.

Under these conditions we have a couple of pieces of information that our mapping process can leverage: individual keys and values whether separate or combined, the set of keys in a record, the set of values for a given key over a sequence of records, or a combination of any of the above. Some of these features, such as the keys, incorporate semantic meaning, while others contain hints about the type of data that is embedded within them. A timestamp, for instance, has a reasonably distinctive format. In addition to the data itself, each record is annotated with some information about the source from which it was obtained or observed.

2) Mapping Pipeline: To maximize the chance of an accurate translation from key-value records to a linked subgraph, we perform the translation in several stages designed to iteratively identify candidate data translations, refine those candidates, and construct graph mappings using those candidates.

TAPIO's mapping pipeline is *data-oriented*: we restrict the set of potential candidates to the *DatatypeProperties* contained in the ontology, requiring that literals in the incoming records be mapped to data in the resulting graph. Objects in the resulting graph are then realized by implication, e.g. we have observed a `process#PID` property and its domain is `Process` thus implying that an object of that type exists.

The process of identifying candidate data translations is called *name resolution*. At this stage TAPIO uses a variety of algorithms in parallel, drawing techniques from machine learning, information retrieval, and other domains. Each *name resolver* produces a set of translations from the keys contained in the input data record to the candidates in the ontology. These candidates are then refined in a *meta name resolver*, whose role is to learn the efficacy of each *name resolver* algorithm in the first and use that to weight the results.

Given a refined set of candidates, we attempt to identify the best method for constructing a linked semantic subgraph from some subset of them. This *linking* stage applies an algorithm based on the method employed by Knoblock, et. al. [12] to each possible combination of the data candidates. The ontology is the source of information about how the semantic subgraph can be constructed. Each candidate datatype property causes the creation of node with of the type of its `rdfs:domain`. Multiple properties with the same domain are attached to the same object, unless they are duplicative with another property; in that case a second object is created. The linking algorithm then attempts to connect the objects by traversing *ObjectProperties* identified in the ontology. The

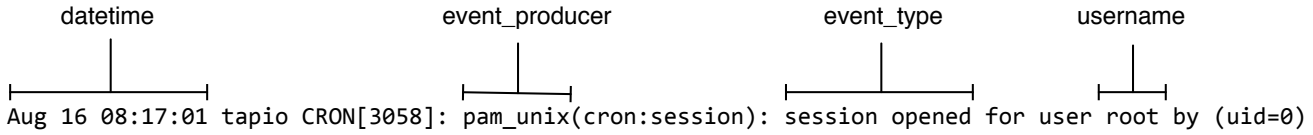


Fig. 1. An example of how unstructured data can be transformed to key-value pairs. Sample keys are shown above segments of the data

result is a connected subgraph generated based on the input combination of datatype properties.

The best resulting subgraph is selected using a scoring method that trades off between including additional candidates and the size of the resulting subgraph.

VI. Discussion

In the previous sections we have discussed our approach to constructing an ontology for representing all information that might be useful for a cyber defender, several elements that make building an ontology designed to be used by an automated mapping system difficult, and an overview of our prototype that attempts to perform that automated mapping. We now turn to a discussion of some of the practical challenges that our TAPIO system is likely to encounter in real-world operations.

A. Using Ontologies for Automated Mapping

RDFS/OWL ontologies provide a very flexible method for capturing and exchanging the semantics of data, but the very nature of linked data can present some challenges when attempting to automatically translate observed data into an ontological representation thereof. Semantic graphs encode data not only in *DatatypeProperty* edges (effectively data fields), but also in the *Class* of nodes and *ObjectProperties* connecting them. A correct translation from observed data to an ontology subgraph requires not only that the data fields are correctly aligned to datatype properties, but that the classes and semantic relationships between them are chosen correctly. This has several ramifications when designing an ontology to be used for automated mapping:

1) Data Domain Specificity: The first is that in any ontology used for automated mapping, domain and range labels become very important for both data and object properties. Our mapping algorithms attempt to map observed data to *DatatypeProperties*, i.e. to literal data labels. According to the definitions of RDFS, the existence of a property with a domain D allows us to infer that an object which is a member of the class D exists. We use this logic to instantiate an object of class D based on the existence of the datatype property. If we are not specific about domain values, whether through non-specification or overly generic specification, the system cannot infer the existence of specific objects. Consider the example ontology shown in Listing 1, which represents two types of objects - Users and Processes - with a single generic `name` data property. We call this property *generic* because the domain is the generic `owl:Thing` class instead of either `Process` or `User`. If the mapping algorithms correctly identify that an element of observed data maps to the `name` property, they still have little ability to guess which class it should be associated with.

```
:Process a owl:Class ;
    rdfs:label "Process"@en ;
    rdfs:subClassOf owl:Thing .

:User a owl:Class ;
    rdfs:label "User account"@en ;
    rdfs:subClassOf owl:Thing .

:name
    a owl:DatatypeProperty ,
      owl:FunctionalProperty ;
    rdfs:domain Owl:Thing ;
    rdfs:range xsd:string .
```

Listing 1. Example Ontology for User and Process with generic properties

In developing the ontology, we aimed to be as specific as possible with respect to datatype property domains. This means that the ontology often contains similar properties - `hasName` exists in both the `user` and `process` sub-ontologies - that at first glance seem like they should be consolidated into one generic property. However, each has a different dICAS omain so that we can assert the existence of a `UserAccount` or `Process` object respectively.

2) Ambiguity in Object Properties: As discussed, the automated mapper constructs a set of objects by inferring their existence after mapping literal data elements to datatype properties. At this point, each object is independent and the system still needs to identify the correct *ObjectProperties* to provide the semantic relationships between them.

It is quite natural for a human to consider multiple nuanced semantic relationships between objects and discern the correct one in any given situation. Consider the existence of a `Process` object and an `IPCSocket` object. There are several different possible relationships between these two objects - the process either `connectsTo` the socket, or it `binds` the socket. To a human looking at the input data, it is relatively easy to discern which of these two situations they are observing; but machines have far less perspective that they can bring to bear.

An automated mapping system has little option but to choose one (or both) of the properties at random. Leveraging OWL constructs such as `disjointWith` can avoid the latter case, but does little to alleviate the larger issue.

In the ICAS ontology, we take the approach of requiring that all semantically-related but specific *Object Properties* share a base property that the mapper can *fall back* to in the absence of more specific information. In the example outlined above, this means that we create a new property `usesSocket` and designate both `connectsTo` and `binds` as subproperties of this object property. We then permit the mapping algorithms to choose either the base property or at most one of its subproperties, giving them flexibility to be specific if there is supporting evidence, but allowing a generic fallback that does not imply semantics that are incorrect. At present, we permit only one tier of subproperties for simplicity.

B. Operating in a Dynamic World

Another challenge arises due to our use of the ontology as a *commitment* to a data representation in a distributed data storage system. As data is observed and collected, it is stored according to the semantics of the ontology at that point in time. In the event that the ontology needs to be updated to account for new information or new understanding, any existing data needs to be adjusted to avoid semantic *skew* between the incoming data and query layer and the existing data.

The naive solution to this problem is to simply discard historical data when the ontology is updated, on the assumption that changes to the ontology as a semantic model are rare events. This is an unsatisfying solution. We envision several more robust solutions to this problem. The first is to identify the conflicts between the existing and new ontologies and request that the user provide translation rules for how to update the existing data. The second solution is to tag all data with the version of the ontology under which it was mapped, and constrain searches to a particular semantic data slice. This latter method is easy to implement, but has the effect of partitioning the database and significantly reducing the efficacy of the tool. The former method of using translation rules to *migrate* the old data to the new schema maintains the utility of the system, but is much more expensive from the perspective of both knowledge engineering and system performance. We intend to explore both methods as TAPIO matures.

In our future work, we will also continue focusing on optimizing the ontology in order to improve query performance. We will explore the use of additional semantics to process complex events and achieve a better understanding of the relationships between them.

VII. Conclusion

In this paper we have suggested that semantic web ontologies can play an active role in developing new paradigms for computer security. In support of that argument, we describe our prototype TAPIO tool that uses the newly developed ICAS cyber security ontology to enable both description of semantically-related security data and machine-assisted translation of arbitrary sources to the ontology. We discuss some of the requirements and challenges presented by using semantic web technology in this dual role. Our goal is that TAPIO will help change the way we think about computer network defense by enabling analysts to seamlessly and intuitively hunt for targeted attacks, and that in doing so show the value of semantic ontologies in enabling new technologies for cyber security.

VIII. Acknowledgments

This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA) Integrated Cyber Analysis System (ICAS) program. The views, opinions, and/or findings contained in this article are those of the author(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

References

[1] *A Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE)*. URL: http://www.loa.istc.cnr.it/old/Papers/DOLCE2.1-FOL.pdf (visited on 06/08/2015).

[2] Paolo Atzeni, Paolo Cappellari, and Giorgio Gianforme. "MIDST: model independent schema and data translation". In: *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*. ACM. 2007, pp. 1134–1136.

[3] Paolo Atzeni, Paolo Cappellari, Philip Bernstein, et al. "Modelgen: Model independent schema translation". In: *Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on*. IEEE. 2005, pp. 1111–1112.

[4] Philip A Bernstein, Sergey Melnik, and Peter Mork. "Interactive schema translation with instance-level mappings". In: *Proceedings of the 31st international conference on Very large data bases*. VLDB Endowment. 2005, pp. 1283–1286.

[5] Yoan Chabot et al. "A Complete Formalized Knowledge Representation Model for Advanced Digital Forensics Timeline Analysis". In: *Digit. Investig.* 11 (Aug. 2014), S95–S105. ISSN: 1742-2876.

[6] Y. Chabot et al. "Automatic Timeline Construction and Analysis for Computer Forensics Purposes". In: *Intelligence and Security Informatics Conference (JISIC), 2014 IEEE Joint*. Sept. 2014, pp. 276–279. DOI: 10.1109/JISIC.2014.54.

[7] *CISCO's 2014 Annual Security Report*. URL: https://www.cisco.com/web/offers/lp/2014-annual-security-report/preview.html (visited on 07/21/2015).

[8] Stefan Fenz, Thomas Pruckner, and Arman Manutscheri. "Ontological Mapping of Information Security Best-Practice Guidelines". In: *Business Information Systems*. Ed. by Witold Abramowicz. Vol. 21. Lecture Notes in Business Information Processing. Springer Berlin Heidelberg, 2009, pp. 49–60. ISBN: 978-3-642-01189-4.

[9] Almut Herzog, Nahid Shahmehri, and Claudiu Dumar. "An Ontology of Information Security". In: *International Journal of Information Security and Privacy (IJISP)* 1 (4 2007), pp. 1–23.

[10] John P. Mello Jr. "Cybersecurity Suffers from Talent Shortage". In: (Apr. 2015). URL: http://www.monster.com/technology/a/Cybersecurity-Suffers-from-Talent-Shortage (visited on 04/27/2015).

[11] Anya Kim, Jim Luo, and Myong Kang. "Security Ontology for Annotating Resources". In: *Proceedings of the 2005 OTM Confederated International Conference on On the Move to Meaningful Internet Systems: CoopIS, COA, and ODBASE - Volume Part II*. OTM'05. Agia Napa, Cyprus: Springer-Verlag, 2005, pp. 1483–1499. URL: http://www.nrl.navy.mil/itd/chacs/sites/www.nrl.navy.mil.itd.chacs/files/pdfs/Kim%20etal2005.pdf.

[12] Craig A. Knoblock et al. "Semi-automatically mapping structured sources into the semantic web". In: *The Semantic Web: Research and Applications*. Springer, 2012, pp. 375–390. URL: http://link.springer.com/chapter/10.1007/978-3-642-30284-8_32 (visited on 12/01/2014).

[13] Dongwon Lee et al. "Nesting-Based Relational-to-XML Schema Translation." In: *WebDB*. 2001, pp. 61–66.

[14] Lindsley G. Boiney. *The Human Side of Agile Cyber Defense: Leveraging Cyber Analysts' Expertise*. 2014.

[15] Alan W McMorran. "An introduction to iec 61970-301 & 61968-11: The common information model". In: *University of Strathclyde* 93 (2007), p. 124.

[16] Leo Obrst, Penny Chase, and Richard Markeloff. "Developing an Ontology of the Cyber Security Domain". In: *Proceedings of the Seventh International Conference on Semantic Technologies for Intelligence, Defense, and Security*. STIDS '12. Fairfax, Virginia, USA, 2012, pp. 49–56. URL: http://ceur-ws.org/Vol-966/STIDS2012_T06_ObrstEtAl_CyberOntology.pdf.

[17] Alessandro Oltramari et al. "Building an Ontology of Cyber Security". In: *Proceedings of the Ninth International Conference on Semantic Technologies for Intelligence, Defense, and Security*. STIDS '14. Fairfax, Virginia, USA, 2014, pp. 54–61. URL: http://stids.c4i.gmu.edu/papers/STIDSPapers/STIDS2014_T8_OltramariEtAl.pdf.

[18] *Ontology:DnS*. URL: http://ontologydesignpatterns.org/wiki/Ontology:DnS (visited on 06/08/2015).

[19] *Ontology:DOLCE+DnS Ultralite*. URL: http://www.ontologydesignpatterns.org/wiki/Ontology:DOLCE+DnS_Ultralite (visited on 06/08/2015).

[20] Simona Ramanauskaite et al. "Security Ontology for Adaptive Mapping of Security Standards". In: *International Journal of Computers Communications and Control* 8.6 (2013), pp. 878–890. URL: http://univagora.ro/jour/index.php/ijccc/article/view/764.

[21] Ansgar Scherp et al. "F–a Model of Events Based on the Foundational Ontology Dolce+DnS Ultralight". In: *Proceedings of the Fifth International Conference on Knowledge Capture*. K-CAP '09. Redondo Beach, California, USA: ACM, 2009, pp. 137–144. ISBN: 978-1-60558-658-8. DOI: 10.1145/1597735.1597760. URL: http://doi.acm.org/10.1145/1597735.1597760.

[22] *The CybOX Project*. GitHub. URL: https://github.com/CybOXProject (visited on 05/11/2015).

[23] *The MAEC Project*. GitHub. URL: https://github.com/MAECProject (visited on 05/11/2015).

[24] *The STIX Project*. GitHub. URL: https://github.com/STIXProject (visited on 05/11/2015).

[25] *The TAXII Project*. GitHub. URL: https://github.com/TAXIIProject (visited on 05/11/2015).

[26] *The Timeline Ontology*. URL: http://motools.sf.net/timeline/teimline.html (visited on 08/17/2015).

[27] Ju An Wang and Minzhe Guo. "OVM: An Ontology for Vulnerability Management". In: *Proceedings of the 5th Annual Workshop on Cyber Security and Information Intelligence Research: Cyber Security and Information Intelligence Challenges and Strategies*. CSIIRW '09. Oak Ridge, Tennessee, USA: ACM, 2009, 34:1–34:4. ISBN: 978-1-60558-518-5.

Similarity in Semantic Graphs: Combining Structural, Literal, and Ontology-based Measures

Lindsey Vanderlyn , Carl Andersen, Plamen Petrov

Raytheon BBN Technologies, Arlington, VA, USA
{lvanderl, canderse, ppetrov}@bbn.com

***Abstract*—Semantic graphs provide a valuable way to represent data while preserving real world meaning. As these graphs become more popular for storing large quantities of data, it is important to have methods of determining similarity between nodes in the graph. This paper extends previous structural similarity algorithms by taking advantage of meaning contained in a graph's literals and the graph's ontology and allowing users to control how much each type of similarity effects overall scores. Preliminary tests indicate that including these sources of similarity increases scores in way that is better aligned with human intuition.**

Keywords: Semantic Graphs, Graph Similarity, Semantic Similarity, Structural Similarity, Ontological Similarity, Literal Similarity, Intelligence Problem Decomposition, Random Walks

I. Introduction

One of the challenges faced with increasingly large data sets is analyzing the information. In particular, finding similar pieces of data from within a larger data set can prove difficult. This problem is especially present when analyzing semantic graphs, which preserve the meaning and context of data by representing objects in terms of their relationships. Raytheon BBN Technologies (BBN) has been expanding previous research to fully utilize information encoded in a semantic graph.

The present paper proposes SSDM+, an algorithm which extends existing techniques for computing the *structural similarity* between two graph nodes. Structural similarity techniques measure node pair similarity by examining the similarity of the pair's respective subgraphs. In the following sections, we will give a brief introduction to semantic graphs, detail the motivation behind this research, describe the prior work we build on, explain the new algorithm we have developed, and demonstrate the application of our algorithm to the decomposition of a notional intelligence problem of "Illegal Fishing".

II. Semantic Graphs

Resource Description Framework (RDF) graphs are comprised of two types of elements: resources and literals. Resources are things which can be described, such as objects and relationships. To remove ambiguity about which object a statement is being made, resources are given globally unique resource identifiers. Literals, in comparison, are used to annotate resources with data values such as strings and integers. There is no expectation that literals will be unique, so with RDF statements in the form of subject-predicate-object triples, literals are never allowed to be the subject or the predicate.

Another key feature of RDF graphs is the inclusion of ontology data in the graph itself. Ontologies are often compared to relational database schemas. Ontologies define classes, relationships, and their inheritances. Through these definitions, all of the statements defined for a class are added to the graph (inferred) every time a user creates a new instance of that class. For example, when given an ontology, a reasoning engine will infer that instances of a class are also instances of all of the class's ancestors.

III. Motivation for this work

This work is primarily motivated as part of an effort to develop a tool for creating intelligence problem decompositions represented as semantic graphs. In this context, finding similarity between nodes in a semantic graph could allow analysts to collaborate and reuse portions of existing problem decompositions when working on a new problem.

A. Semantic problem decomposition

In order for analysts to determine what information needs to be collected for a particular intelligence problem, they must first break the problem down in a logical and systematic way - a process we refer to as problem decomposition. To illustrate this, we will use

a fictitious intelligence problem of "Illegal Fishing in Hawaii" shown on Fig. 1. The first step is to determine the **states** in which entities related to the problem (e.g., fishing boats) can exist. For example, a fishing boat might be either "in port" or "in a legal fishing zone". Next, the analyst must specify how to tell if an entity is in a particular state by creating **indicators**. Indicators must be boolean statements such as "boat moving" or "outriggers deployed". The analyst then specifies the observable phenomena, or **stimuli**, that needs to be collected in order to determine if an indicator is present. Next the analyst defines sensor system specific collection parameters, called **observation needs** such as resolution, location, and time to collect the stimuli. Finally, an analyst can specify **algorithms** which combine input from one or more observation needs to determine whether an indicator is present.

Semantic graphs are a good way to represent problem decompositions because they provide an intuitive way for analysts to capture the relationships from a mental model of the problem into a format, which is both machine-readable and human-understandable. In addition, storing the decomposition as a graph preserves the semantics of the relationships between decomposition components. Finally, these graphs can also be reasoned over to determine completeness, decomposition alternatives, or efficiency of collection.

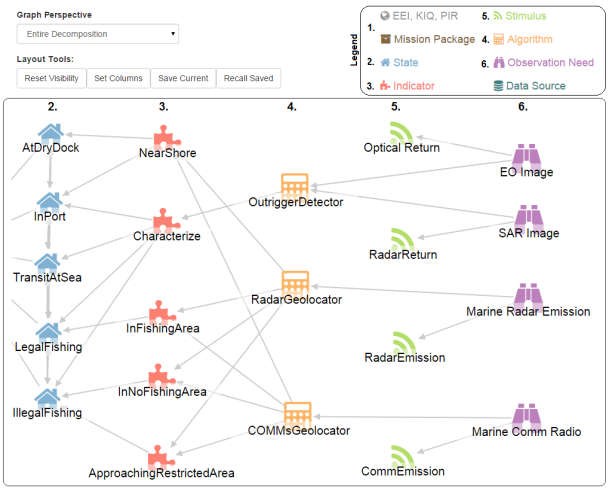


Fig. 1: *Example of a simple semantic problem decomposition graph for the intelligence problem "Illegal Fishing in Hawaii". Nodes in column 1 represent intelligence problems (not shown), column 2 - States, column 3 - Indicators, column 4 - Algorithms, column 5 - Stimuli, and nodes in column 6 represent Observation Needs. Edges represent logical relationships such as "Supports" or "State Transition".*

IV. Previous Algorithms

Our work on SSDM+ derives from three earlier works. These are SimRank [2], the work of Fogaras and Racz in their paper: Scaling Link-Based Similarity Search [5], and the Semantic Similarity Distance Metric (SSDM) [7]. These algorithms share the following desirable qualities:

- They are domain-independent, meaning that they can be applied to any data representing relationships between entities.
- They can be computed efficiently over very large datasets, in contrast to algorithms which scale very poorly, such as ones which rely on Singular Value Decomposition [4].
- They have the ability not only to determine the similarity of any two given nodes, but also to generate a list of entities which are most similar to one specified by a user.
- The computations are easily understood, meaning that similarity scores generated can be explained based on the actual computation performed - something which can be difficult with more abstract calculations.
- The algorithms look beyond a node's immediate neighbor to create a broader knowledge of the entity's structural context.

We provide a more in depth explanation of each of the previous algorithms below followed by an overview of the changes we have made with SSDM+.

A. SimRank

The key insight of SimRank is that two nodes are similar if they are connected to similar nodes [2]. This simple idea can be translated mathematically as: the similarity score between two entities is the average pairwise similarity of their neighbors, scaled by a decay factor.

A concrete example of this can be seen with movie data. Consider the example in Fig. 2.

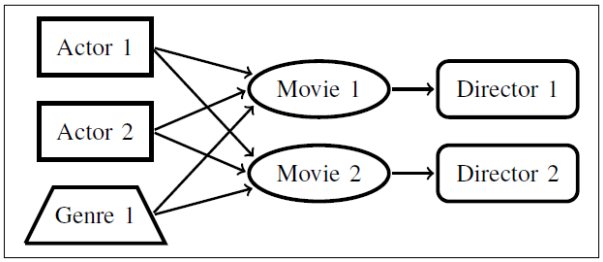


Fig. 2: *Example of a semantic graph [7] with information on movies. Director 1 and Director 2 are similar because they directed similar movies, although the directors themselves are not connected*

Movie 1 and Movie 2 appear to be similar to each other because they are in the same genre and they both have two of the same actors. Furthermore, although

Director 1 and Director 2 do not share any nodes, they can also be considered similar because they directed similar movies. SimRank provides a formalization of this type of idea.

Each pair's similarity is dependent on many other pairs because of the recursive definition of SimRank. On small datasets, the system can be solved with an iterative algorithm. On large datasets, it can be solved using an efficient approximate method outlined by Fogaras and Racz in [5]. The SSDM calculation and our extensions are based on this efficient approximate method.

B. Fogaras and Racz's extensions

The algorithm outlined by Fogaras and Racz relies on the mathematical notion of a *random walk* through a graph, in which an abstract walker steps from node to node through the graph by following random edges [5]. In the original SimRank paper, Jeh and Widom [2] observed that the SimRank score of two nodes can be approximated from the expected meeting time of two random walkers starting at those two nodes; a longer expected meeting time corresponds with a lower SimRank score. Fogaras and Racz used this observation to develop an efficient and scalable algorithm for calculating similarity scores. In their algorithm, one walker is initialized per node and at every time step, each walker steps across one edge. To reduce the number of computations required, walkers converge (are treated as a single walker) when they meet at the same node. Fogaras and Racz found that this type of convergence does not reduce the correctness of the approximate calculation.

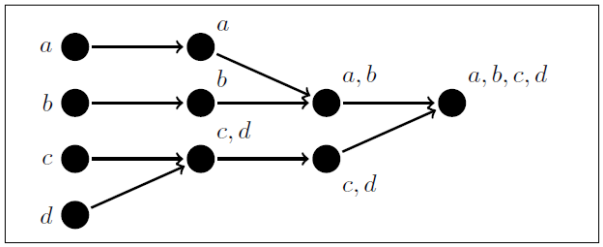


Fig. 3: *Diagram of walker convergence [7]. Walkers a, b, c, and d begin as independent. As the progress through the graph (left to right) they meet each other and converge.*

Additionally, Fogaras and Racz's algorithm incentivises walkers to converge if they are near each other. Convergence is encouraged as a result of how a walker chooses its next step. In this model, all of the edges in the graph are randomly assigned an index and walkers will always choose the edge with the lowest index to step to. This shuffling of indexes is universal. This means that if walkers share multiple of the same possible next steps, and the walkers select a shared node as the next step, they must choose the same shared node. Restated: the set of shared possible next steps for the two nodes only contains one edge with the lowest index. Therefore, if each of the walkers chooses the edge with the lowest index from this set, they will choose the same edge and meet.

The output of one run of Fogaras and Racz's scalable SimRank algorithm is a collection of *fingerprint graphs* which store the convergence data for each of the random walkers. Each graph stores the node a walker began on, the node it merged into, and how many steps it took for that convergence to occur. These fingerprint graphs are precomputed and are collected to be stored in database which can be queried at runtime. Separating the fingerprint graph generation and querying allows real time similarity queries to be quite rapid.

C. SSDM

SSDM [7], also developed by BBN, was closely related to SimRank with two principal differences. The first difference is that SSDM seeks to be independent of ontological choice, meaning that the directionality of each edge (predicate) is ignored.

The second involves the semantics of RDF graphs. Whereas SimRank is a measure over unlabeled directed graphs, SSDM incorporates the semantic labels given to edges. This makes it well-suited to semantic graph data, which is organized in subject-predicate-object triples. Subjects and objects are equivalent to nodes in the graph and predicates are equivalent to edges. In RDF graphs, these predicates, or labeled edges, describe how the nodes they connect are related to each other. In order to utilize the meaning contained in these edge labels, the SSDM algorithm only considers two walkers to have met if they arrived at the same node having traveled through an identical list of edges to get there. This adds a stricter definition of similarity; it is no longer good enough that two nodes were connected to the same third one, they now had to be connected in the same (semantic) way.

V. SSDM+: Incorporating literal and ontology-based similarity

Our new algorithm builds off of previous work with SSDM by allowing literals to influence similarity scores, relaxing the need for walkers to meet at the exact same resource in order to converge, and creating a stricter approach to how the ontology influences similarity. This is motivated by the desire to create a cohesive model for including literal and ontological similarity in a larger structural similarity framework. Additionally, we removed the weighting of similarity scores based on the time it takes walkers to meet (decay), because the identical paths restriction already significantly decreases the probability of any walkers meeting after very few

steps and we found that further decay was not necessary. Otherwise we retained the same functionality and methods as the original SSDM algorithm.

In SSDM, similarity was calculated only taking into account the resources in the graph; literals in the graph were completely ignored. Additionally because of the strict definition of convergence, nodes which humans might consider to be very nearly the same (such as multiple instances of the same class) could not contribute directly to the similarity scores. This approach does not fully utilize the ontological information stored in a semantic graph. String literals often serve as meaningful labels for nearby nodes; numerical literals store quantities that tend to be of similar magnitude for nearby nodes. Therefore ignoring literals causes some nodes to appear much more similar than a human might consider them and the requirement for walkers to walk the exact same predicate path, prevented inferred inheritance from sufficiently representing the ontological similarity of two nodes. In particular, because of the number of inferred classes a resource might have, it was highly probably that two nodes, which shared an ancestor, might walk a path to that ancestor, but each take a different number of steps to get there. Additionally, many of the inferred classes are very high level concepts (e.g. owl:Thing) and the number of instances of these classes is high enough in a graph that even if two walkers did converge at one of these parent classes, a human would not consider that convergence to have added any similarity.

With SSDM+, we combine both the literal and ontological similarity into a larger structural approach. There may be other existing works which make use of literal and ontological similarity, but we are unaware of any others which do so as part of a cohesive structural similarity measurement.

A. Literal Similarity

In order to incorporate literal similarity, we had to loosen the definitions of convergence present in the original paper [7]. Rather than only converging when two walkers had reached the same resource (referred to here as "physcal" convergence), walkers could converge if they reached two literals that were sufficiently similar. In this case, we defined similarity for numeric literals as the ratio of the smaller over the larger - or percent. The similarity for all other literals - which for simplicity were converted to strings - was calculated as a Levenshtein distance between the two, normalized for their sizes. Levenshtein distance is a measure of structural similarity between two strings by measuring the number of substitutions, insertions, and deletions required to turn one string into the other. These metrics work well with the data sets we tested, but for graphs which contain long strings or vastly different types of data, alternative to Levenshtein distance (e.g., [6]) can be utilized.

B. Ontological Similarity

In an analogous way, we added ontology-based similarity by allowing walkers to converge if they stepped onto resources which we considered to be ontologically similar. In this case, we define "ontologically similar" as: sharing a most-distant-salient ancestor, with salience defined as:

$$Salience = 1 - (instances/total) \quad (1)$$

where *instances* is the number of instances of a class and *total* represents the total number of nodes in the graph. This number represents how unique each class is, or how much of the graph is **not** made up of instances of that class.

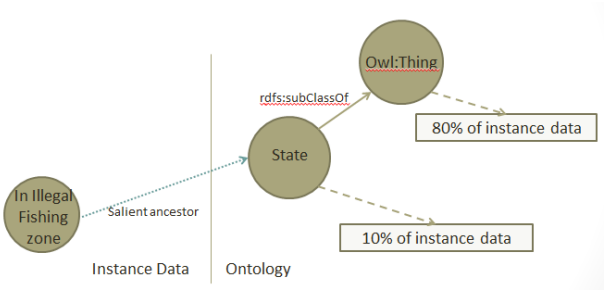


Fig. 4: *Example of finding the most-distantly-salient ancestor. The node "In Illegal Fishing zone" is an instance of a State (fairly unique). State is also a subclass of owl:Thing, but because that is too common to be considered salient, "In Illegal Fishing zone's" most-distant-salient node is State.*

To find the most-distant-salient ancestor, we precomputed the salience of each class and traced its ancestry until we could find the most distantly related ancestor which remained more salient than a user defined cutoff (see example in Fig. 4). Defining similarity this way means that all instances of a class that is considered to be salient (or instances of that class's children) will have the same most distantly related ancestor, thus reducing the number of comparisons needed to find similar nodes. Additionally, when we implemented the ontology-based matching, we removed nodes representing inferred types from the graph, which allows the user to fully control the effect of ontology-based matches on the final similarity score and removes the possibility for nodes to match based on a shared, unsalient class.

C. Five Types of Convergence

To make this tool as flexible as possible, we compute an overall similarity score composed of five different similarity scoring methods. The user assigns weights

for each type, allowing them to emphasize the method they believe will deliver the most accurate results in the current context.

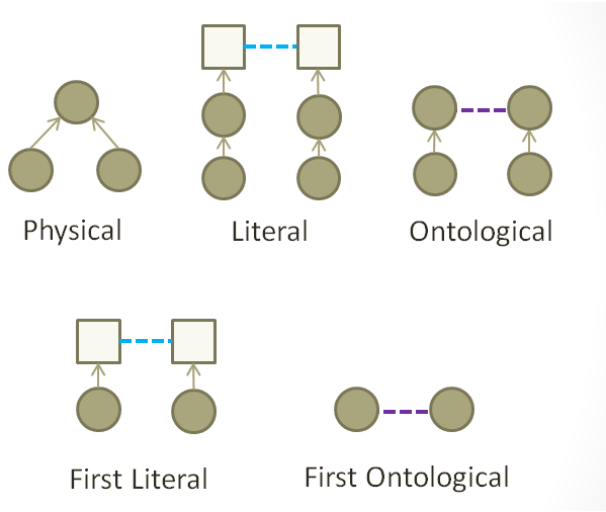


Fig. 5: *Visualization of different convergence types. Arrows represent edges in the graph connecting nodes. Light (blue) dotted lines indicate literal similarity of unconnected nodes. Dark (purple) dotted lines indicate ontological similarity of two unconnected nodes.*

Walkers are considered to have converged when:

1) Physical convergence: two or more walkers meet at the same resource

2) Ontology-based convergence: two or more walkers step onto nodes whose classes share a most-distant-salient ancestor

3) Literal based convergence: two or more walkers step onto literals that are more similar than a user defined cutoff

4) First ontology-based convergence: the nodes they start on are considered to be ontologically the same - because the graph sizes can become quite large, this is only evaluated after the first step

5) First literal based convergence: walkers step onto similar literals after the first step - this is considered significant because it is the only step where the literals are guaranteed to be connected to the node a walker started on

The last two types of convergence are separate from the others because they are the only two types of convergence that happen based on the original nodes a walker started on. If a user were to query the similarity between two nodes, while they are important to finding the overall structural similarity, those two are the only types of similarity that are uniquely about the queried nodes.

D. Integration of new types of convergence

As with the original SSDM, finding similarity is broken into two stages: a preprocessing stage where fingerprint graphs (trees) are generated from the convergences that occur during random walkers and a runtime stage where these trees are queried. The changes we have made affect the former. SSDM+, our new algorithm, maintains the random walk paradigm of the original SSDM algorithm and continues to use convergences to generate trees. The primary difference is the way walkers can converge. By extension, trees now also contain an additional piece of information: the type of convergence which has occurred. Tracking the convergence type allows a user to control how much each type of convergence affects the similarity scores they receive.

By preserving the random walks paradigm, we gain a consistent framework for assessing all five types of similarity. From a more practical view, however, the requirement that walkers must travel the same predicate path greatly reduces the number of comparisons which must be completed to find literal and ontological similarity.

In addition, the order which we check convergences was also influenced by the need to reduce the number of comparisons, a need which becomes increasingly important as datasets grow. We first check the ”physical” convergences because they are the most common type of convergence. We then check for literal convergences and then for ontological convergences. This order matters less because we separate the literals and resources so the two types of convergence are independent of one another. In order to reduce the number of comparisons we must perform, we changed the way “physical” convergence is calculated. In the new method, we sort all walkers by the path they have traveled, then split within those groups, by the current location of the walker. With the nodes that did not converge, but did walk the same path, we split the walkers based on whether they are located at a resource or a literal and look to see if they can converge via ontological similarity or literal similarity. This is a more efficient method (as compared to the original SSDM), when literal and ontological similarity must also be determined. The main drawback to this order of comparisons is that literal and ontological similarity are not calculated until after the first step has been taken. This greatly increases the speed of the calculation, but does ignore ontological and literal similarity between two original nodes if their walkers did not take the same first step. Given the high number of random walks we perform for each graph, we consider this acceptable.

VI. Testing and Tuning

In the course of the development of the SSDM+ algorithm, we executed multiple test cases comparing

the results of the algorithm against human intuition on which nodes should be considered to be similar. Based on these empirical results, we developed a procedure for tuning the similarity paramters, as outlined below.

A. Determining literal and ontological cutoff similarities

To determine some baseline values for similarity thresholds, we experimented with both the literal and ontological cutoffs. Cutoffs represented the minimum similarity needed for a user to consider two literals or two resources to be the same for the purposes of allowing convergence. We recorded the time it took to generate trees and the similarity scores between three sets of nodes. These experiments allowed us to optimize the shortest time needed to generate the trees for the highest increase in similarity from the added literal and ontological matching. For all of the tests, the weighting on each type of convergence was kept equal so that only the cutoffs were being varied. From our experiments, we found that literals needed to be at least 60% similar and classes could be considered salient if less than 60% of the instance data was an instance of that class. With higher cutoffs, the time needed to generate the trees began to increase dramatically. With lower cutoffs, similarity scores did not match very well with human predications. It should be noted that these baseline values were generated using mock intelligence problem decompositions and may need to be adjusted for other types of datasets.

B. Determining weights for physical, literal, and ontological convergences

In order to see the effect of allowing literal and ontological matching, we created a series of decompositions. We then tested them to see how changing the weighting of each type of similarity affected which nodes were returned as similar when queried and how similar the algorithm considered them. As a first test, we constructed ten identical graphs whose only difference was in the literal labels associated with each node. The goal of this test was to show that when a node was selected, the most similar resources would be the corresponding nodes in the other graphs.

Once we verified that this was true, we started to test on graphs which were generated from the same template, but were not identical. Templates in the Semantic Problem Decomposition (SPD) tool are fragments of a decomposition that can be reused. When they are copied into a new graph, they retain all of their properties, except that all of the nodes in the graph are given new randomly generate Universal Resource Identifiers (URIs), which are globally unique. These graphs were the primary data we used to test the weighting. In these tests, we used a smaller dataset - four graphs, rather than ten - and aimed to find parameters which returned results that we, as humans, understood to be similar. In order to get more universal results, we made sure to test several different types of nodes and observe the results. This proved to be interesting as some of the nodes were more influenced by literals - in both positive and negative manners - than others. From these tests we were able to find the weights which had the greatest improvement over the results generated from only considering physical convergence.

Results of this type of tuning may be dependent to the datasets we used. We believe that, in order to use this for other datasets, it would be important to run a similar test to choose appropriate weights.

VII. Application: Illegal Fishing Problem Decomposition

A tool for finding similarity between nodes in a semantic graph can be useful in a variety of settings. In this case, however, we are primarily interested in how it can aid in rapid development of intelligence problem decompositions via the retrieval and reuse of existing decomposition structures. While representing problem decompositions in a semantic way has many benefits, it can also be a time consuming process to manually input all of the relevant nodes. We envision using SSDM+ to suggest nodes from existing problem decompositions which are similar to a node selected by a user. In this way, the user would be able to look at the graphs associated with each suggestion and, if desired, copy the suggested node and its children into the new decomposition.

This interaction reduces the amount of time it takes an analyst to perform their job, but also gives them more context for generating a new decomposition. In addition to saving the analyst time, a tool like this also provides analysts the opportunity to see how their peers solved similar problems. This type of knowledge sharing could allow analysts to envision new ways of completing their decomposition tasks. This latter feature has the potential to be especially significant, given the limited sensor resources for collecting stimuli.

In this section we use the mock decomposition problem of illegal fishing which we described earlier in the paper and demonstrate some of the preliminary results we have been generating.

A. Results

The results shown in this section were determined by comparative analysis of similarity suggestions. We created four decompositions from the same starting template and modified them by adding and deleting

Top Three Suggestions of Most Similar Nodes and Their Similarity Scores for the Node: "In Illegal Fishing Zone 4"

Types of convergence	First suggestion (score)	Second suggestion (score)	Third suggestion (score)
Physical (SSDM)	In Illegal Fishing Zone (0.57)	In Illegal Fishing Zone2 (0.46)	In transit3 (0.33)
Physical and Literal	In Illegal Fishing Zone (0.57)	In Illegal Fishing Zone2 (0.46)	In transit3 (0.33)
Physical and Ontological	In Illegal Fishing Zone (0.85)	In Illegal Fishing Zone2 (0.75)	In Illegal Fishing Zone3 (0.73)
Physical, Literal, and Ontological	In Illegal Fishing Zone (0.85)	In Illegal Fishing Zone2 (0.75)	In Illegal Fishing Zone3 (0.73)

TABLE I: *Top three results from similar decompositions when querying for most similar nodes to the "In Illegal Fishing Zone" node. The far left column describes the types of convergence which contributed to the scores and the remaining columns represent the labels of the highest scoring nodes and their similarity scores. All of the suggestions are the same class (States), but SSDM+'s inclusion of Literal and Ontological convergence produces scores which align better with human understanding, and does find the "In Illegal Fishing Zone" node from each graph to be the top three most similar.*

Top Three Suggestions of Most Similar Nodes and Their Similarity Scores for the Node: "Outriggers not deployed 4"

Types of convergence	First suggestion (score)	Second suggestion (score)	Third suggestion (score)
Physical (SSDM)	Comms2 (0.40)	SAR (0.38)	SAR4 (0.37)
Physical and Literal	Comms2 (0.40)	SAR (0.38)	SAR4 (0.37)
Physical and Ontological	Outriggers not deployed (0.69)	Outriggers deployed2 (0.51)	Transponder On3 (0.42)
Physical, Literal, and Ontological	Outriggers not deployed (0.70)	Outriggers deployed2 (0.51)	Boat Moving2 (0.42)

TABLE II: *Top three results from similar decompositions when querying for most similar nodes to the "Outriggers not deployed" node. The far left column describes the types of convergence which contributed to the scores and the remaining columns represent the labels of the highest scoring nodes and their similarity scores. SSDM+'s use of Literal and Ontological convergence performs in a more human understandable way here: suggesting nodes which are all of the same class (Indicator) as the node for which the query was generated, in contrast to the original SSDM algorithm which suggests only Stimuli nodes.*

nodes and connections. This gave us four similar, but not identical decompositions. We then generated suggestions for several nodes varying which types of convergences contributed to the similarity scores. A sample of the results can be seen in Table I and Table II where we show the top three suggested nodes and their similarity scores for two of our test nodes. For simplicity, if a type of convergence was included the weight was set to 1 and if it was not included, the weight was set to 0.

In the first example, both algorithms perform similarly. The main difference is the magnitude of the scores. It should be noted, however, that while both algorithms produced results which were all of the same class (State) as the node from which the query was generated, the modified algorithm - when ontological similarity was included - was able to provide a "In Illegal Fishing Zone" node for each of the top three results. The second example, though shows a more extreme distance. The results of the modified algorithm, are from the same class (Indicators) as the node from which the query was generated, while the results from the original algorithm are Stimuli - a class which connects to Indicators.

From our experimentation, we found that are two main improvements offered by SSDM+. The first improvement is that the top suggestions are more frequently closer matches for the node which has been queried. The second improvement stems from the addition of terms for new types of similarity which were previously ignored, resulting in higher scores that align more closely with how a human might perceive the similarity. For example in Table I, the top suggestions are all for the corresponding nodes from other similar decomposition models. These top-suggestion nodes have nearly the same labels as the queried node and, in this test, have identical connections. The similarity scores increase dramatically with ontological similarity turned on, demonstrating the value-added of ontological matching; however, even when considering non-ontological matching the the top three suggestions of SSDM+ are still as good or better than that of SSDM.

While more extensive testing would provide a more definitive picture, our preliminary results indicate that supplementing physical convergence with literal and ontological convergence does increase the accuracy of similarity calculations and the relevance of the results generated. Additionally, we believe that the incentivized Physical convergence described in Section V may unequally weight the Physical component of the score. We plan to investigate and address this issue in future work.

VIII. Future Work

There are several possible directions for future work. The first, and most pressing, is a more rigorous comparison between SSDM+ and the original SSDM algorithm. A major difficulty in performing such a comparison is the lack of concrete ground truth similarity values to compare our algorithm results to. To that end, there is a need to develop a repeatable system for quantifying which nodes humans consider to be similar. However, due to the subjective nature of such measurements,

constructing ground truth data based on user judgments presents many challenges. An alternative approach could be to compare the similarity results from our SSDM+ algorithm to similarity derived from entity attribute comparison - comparisons of the presence/absence of certain attributes. If we choose to pursue this path, there are several well-established algorithms [8] that we could use. However, since the data we have been using so far consists of only a handful of main classes, such comparisons may not be as useful.

The second area for improvement is in tree generation. One of the goals prompting this research was to develop a way for intelligence analysts to reuse portions of existing decompositions when developing new mission packages. To aid in that process, we hoped that analysts could click on a node they had added to their graph and query for all similar nodes among all of the mission packages that had already been developed. This is not yet possible because of the latency involved in the tree (re-)generation stage. In order to re-run the entire graph analysis, it takes more than a minute on small graphs with size of approximately three hundred nodes. Such a long delay is unlikely to be tolerated by an analyst in interactive mode. We are currently investigating ways to reduce latency, including further optimization of the algorithm and applying the algorithm incrementally as nodes are added to a graph.

We also plan to explore more accurate techniques of determining literal similarity. In particular, at present, we analyze only the structure of textual literals rather than looking at the meaning of that text. Finally, although we have relaxed the strictness in matching nodes, we still require that walkers walk an identical predicate path. It may be, that in a similar way to the additional insight gained by allowing ontological matching, we could also gain a better measure of similarity by relaxing the predicate restrictions so that walkers would also be allowed to travel similar rather than identical predicate paths. The challenge with this would be to find a computationally efficient way of relaxing the restrictions because performing something similar to our ontological check for convergence would be too time intensive for a dataset of any reasonable size.

Additionally, the role of literals is currently undermined by the fact that we encourage convergence in the same way as the original SSDM algorithm which biases physical convergences to occur. In doing this, literal matches occur much less frequently and thus have less of an effect on the total results. Due to the low repeatability, however, we found it was not feasible to stop encouraging convergence. To fully utilize similarity from literals in the graph, an alternative to the way convergence is encouraged would need to be developed.

IX. Conclusion

Due to the wide spectrum of application of semantic graphs, finding similarity within and between them also has significant utility. In this paper, we focused on extending previous structural semantic similarity algorithms with functionality aimed at aiding analysts in the development of new intelligence problem decompositions. The resulting extended SSDM (or SSDM+) algorithm will be incorporated in a tool that finds fragments of existing graphs, which could be relevant to the problem being decomposed. This not only saves analyst time by allowing them to reuse relevant pieces from existing decompositions, but also allows the exploration of options for completing the decompositions in ways that might not have been previously considered. The modifications we made to the SSDM algorithm allow it to take better advantage of the data stored in semantic graphs rather than merely using structural (physical) convergences. By providing users control (via weights) over the various types of convergences - physical, literal, and ontological - SSDM+ becomes easily customizable to varying datasets. The extended algorithm also produces results that better match human intuitions on similarity. In the future, we plan to develop a more rigorous test to quantify these improvements and to speed up the off-line (tree) generation of similarity traces.

References

[1] T. Berners-Lee, M. Handler, & O. Lassila, "The semantic web", *Scientific American*, May 2008. [Online]. Avaiable: http://www.ds3web.it/miscellanea/the_semantic_web.pdf

[2] G. Jeh & J. Widom, "SimRank: a measure of structural-context similarity", in *Proceedings of the eigth ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD '02 New York, Ny: ACM, 2002, pp. 538-543. [Online]. Available: http://doi.acm.org/10.1145/77504/775126

[3] A. Islam & D. Inkpen, "Semantic text similarity using corpus-based word similarity and string similarity", *ACM Transactions on Knowledge Discovery from Data*, vol. 2 no. 2 pp. 1-25, July 2008. [Online]. Available: http://doi.acm.org/10.1145/1376815.1376819

[4] R. Hartley & A. Zisserman, *Multiple view Geometry in Computer Vision*. Cambridge University Press, 2003. [Online]. Available: http://books.google.com/books?id=si3R3Pfa98QC

[5] D. Fogaras & B. Racz, "Scaling link-based similarity search", in *Proceedings of the 14th international conference on World Wide Web*, ser. WWW '05. New York, NY: ACM, 2005, pp. 641-650. [Online]. Available: http://doi.acm.org/10.1145/1060745.1060839

[6] W. H. Gomaa, "A Survey of Text Similarity Approaches", *INternational Journal of Computer Applications*, vol 68, no. 13, 2013. doi:10.1.1.403.5446

[7] C. Olsson, P. Petrov, J. Sherman, A. Perez-Lopez. "Finding and Explaining Similarities in Linked Data", *Semantic Technology for Intelligence, Defense, and Security (STIDS 2011)*, Fairfax, Virginia, November 2011.

[8] B. Gallagher "Matching Structure and Semantics: A Survey on Graph-Based Pattern Matching", *American Association for Artificial Intelligence*, 2006, [Online]. Available: http://www.aaai.org/Papers/Symposia/Fall/2006/FS-06-02/FS06-02-007.pdf

Towards an Operational Semantic Theory of Cyber Defense Against Advanced Persistent Threats

Steven Meckl, Gheorghe Tecuci, Mihai Boicu, Dorin Marcu
Learning Agents Center, Volgenau School of Engineering, George Mason University, Fairfax, VA 22030, USA
smeckl@masonlive.gmu.edu, teuci@gmu.edu, mboicu@gmu.edu, dmarcu@gmu.edu

Abstract— **This paper presents current work on developing an operational semantic theory of cyber defense against advanced persistent threats (APTs), which is grounded in cyber threat analytics, science of evidence, knowledge engineering, and machine learning. After introducing advanced persistent threats, it overviews a systematic APT detection framework and the corresponding APT detection models, the formal representation and learning of these models in the knowledge base of a cognitive agent, and the development and integration of such agents into a specific cyber security operation center.**

advanced persistent threat, cyber threat analytics, cognitive assistant, evidence-based reasoning, knowledge-based learning, ontology, argumentation models, symbolic probabilities.

I. INTRODUCTION

An *Advanced Persistent Threat* (APT) is an adversary that leverages superior resources, knowledge, and tactics to achieve its goals through computer network exploitation (CNE). APTs are characterized by their persistence in gaining and maintaining access to targeted networks and their ability to adapt to efforts of network defenders to identify and remediate their activity [1].

Security research companies have been tracking APT groups for years, independently giving them unique names as specific tools, techniques, and procedures (TTPs) are attributed to a group. FireEye/Mandiant has published reports on 30 APT groups since 2013, naming them simply APT1 through APT30 [2].

APT1 is the name given by Mandiant to a group of APT actors, attributed to China's People's Liberation Army unit 61398, who have lead a campaign of cyber espionage since at least 2004. APT1 is known for a regimented approach to computer intrusion activity. An APT1 intrusion typically consists of the following phases: (i) gain access to a network by sending fraudulent, malicious email messages to specific users (spearphishing); (ii) use multiple types of backdoor programs to maintain presence and provide remote connectivity to the target network; (iii) use a collection of command-and-control (C2) servers to obfuscate the source of their attacks; (iv) escalate privileges and acquire legitimate login credentials to access network resources; (v) move laterally within the target network using legitimate credentials to gain redundant points of presence and identify information of interest; and (vi) exfiltrate targeted information through their C2 infrastructure [1].

The technical appendices of Mandiant's report [1] include detailed information on their known backdoor, C2, and exfiltration tools, and a comprehensive set of indicators of compromise (IOCs). Following their report, other security researchers, including the Contagio blog [3] have released supplementary analysis on APT1 malware, techniques, and infrastructure.

APT1, among other attacker groups, practices *evolutionary development* to adapt to changes in network defense technology or simply to increase efficiency. Further analysis of APT1 by Mila at contagiodump.blogspot.com [3] shows a timeline of the attacker group's tool usage from 2004 to 2012, including information on dozens of samples of malware. The group evolved their tool set slowly over the course of at least eight years. These changes in the way malware presents itself on the network and on disk have made it difficult for signature-based intrusion detection tools to detect attacks because the attacks can change static information in their malware faster than defenders can adapt. However, the patterns of behavior change more slowly and with less variance.

Analysis of the indicators of compromise (IOCs) published in [1] shows that APT1 malware demonstrates clusters of behavior. Subsets of the programs share sets of techniques for communicating on the network, persisting through a reboot, or storing data on disk. One example of this is the cluster of malware comprised of BANGAT, SEASALT, KURTON, and AURIGA. While the specific strings used to register the malware as a service or device driver and the names of files and Registry keys differ, the only substantial difference in IOCs between those four tools is the persistence mechanism used to survive a reboot.

Clusters of malware are often called *malware families* in published research. Each member of the family incrementally builds on previous versions as they get detected and become less effective. The Sobig virus is an example of a malware family. It was used in 2003 in a widespread email phishing attack. Joe Stewart describes how the Sobig virus evolved over five different revisions [4, 5]. Over the course of those revisions, the author changed how the malware set its expiration timer, where the command-and-control servers were located, and how encryption was used in an effort to improve the effectiveness of the malware.

Modern cyber defense against APTs is currently done in a cybersecurity operations center (CSOC), which employs teams of network defense experts, analysts, system administrators, and forensics experts. CSOCs leverage a rich tool set including host-based and network-based intrusion detection systems (IDSs), data collections, analysis tools, and visualization tools. CSOCs receive incident information from high-value sources – law enforcement, user reporting, or threat intelligence from

other CSOCs – and unconfirmed alerts from security infrastructure such as antivirus software, IDSs, heuristic alerts, or machine learning algorithms. The analyst's responsibility is to monitor alerts and log information from all of these information sources, each having differing levels of credibility, and use them to make a determination about the presence or absence of intrusion activity [6]. However, because a single event alone does not provide sufficient evidence that an intrusion event has occurred, and modern detection technologies are error-prone, each event must be carefully examined and investigated by a human analyst [6]. In a large enterprise, tens of thousands of alerts per day can be reported. Therefore, even sensors with a false positive rate of less than one percent can generate enough false positives to be unmanageable by even large CSOCs.

This paper presents current research on developing an operational semantic theory of cyber defense against advanced persistent threats. Grounded in cyber threat analytics, science of evidence, knowledge engineering, and machine learning, this theory provides a systematic approach to cyber defense, as well as analytical knowledge and models that can be formally represented in the knowledge bases of cognitive agents, enabling them to perform the functions of security analysts, both automatically and in collaboration with the analysts. These cognitive agents will be directly trained by security analysts to detect APTs, through specific analysis examples and explanations, acquiring and generalizing their expertise. As a result, their analyses will be very explicit, easy to understand, and easily updated by the analysts.

Such agents have the potential to radically change established practice by automating the APT analysis process, significantly increasing the CSOC's efficiency, reducing operation costs, increasing detection rates, and decreasing false positive rates. Most importantly, these agents are designed to continuously learn from security analysts and from the agents' own experience, to keep up with and even anticipate new threats. They will also facilitate sharing of evolving APT analytic expertise and training of the cyber analysts.

The next section presents the systematic APT detection framework and the corresponding APT detection models. Section III overviews the formal representation of these models in an APT ontology and APT detection patterns with ontology-based applicability conditions. Section IV discusses the learning of these models. Section V overviews the architecture of a learning agent shell, a general agent building tool that contains reasoning and learning modules for APT detection, as well as a general knowledge base. It also discusses its use in the generation of customized agents for a specific CSOC.

II. Discovery-based APT Detection Framework

The APT detection framework is represented in Fig.1. Evidence of suspicious activity triggers alternative explanatory hypotheses, which are used to guide the collection of relevant evidence which, in turn, is used to assess the hypotheses.

As will be discussed in this paper, this is both an adaptation and an extension of the general discovery-based framework we have previously developed for intelligence analysis, and implemented in the TIACRITIS [7], Disciple-CD [8, 9] and Cogent [10] analytical tools. A major difference, however, is the significantly higher degree of automation required by the cyber defense process. While the human analysts will still be involved in this process, it is assumed that all the operations can be automatically performed by the cognitive agents.

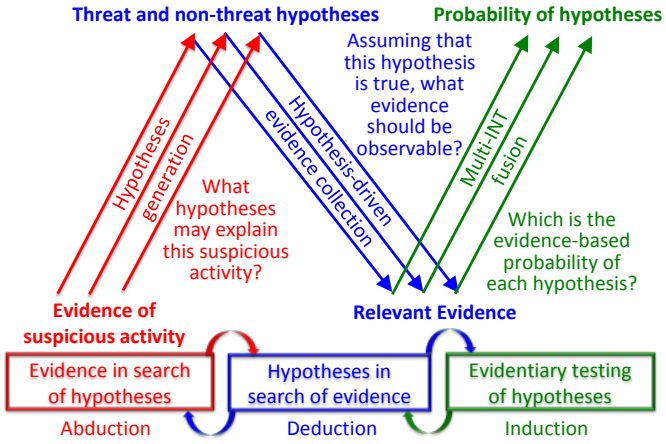


Fig. 1. Automatic APT detection framework.

APT detection is modelled as a continuous collaboration of three automated processes: *Evidence in search of hypotheses*, *Hypotheses in search of evidence*, and *Evidentiary testing of hypotheses*, each described in the following sections.

A. Evidence in Search of Hypotheses

As shown in the left hand side of Fig.1, evidence of suspicious activity was detected (e.g., by monitoring agents, by Bro [11] or Snort [12] IDSs, etc.) and the question is: *What hypotheses may explain it?* Through *abductive* (imaginative) reasoning, which shows that something is *possibly* true, the agent generates a set of alternative hypotheses, some corresponding to actual APT activity, while others corresponding to non-threat activities. Fig.2 is an illustration of this process.

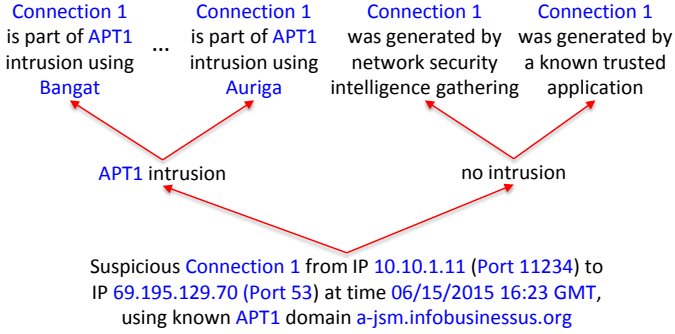


Fig. 2. Evidence in search of hypotheses.

A suspicious connection was signaled by a Snort alert [12] from the internal IP 10.10.1.11 to 69.195.129.70, which is mapped to "a-jsm.infobusinessus.org," a known APT1 domain. It is therefore possible that there is an APT1 intrusion using one of the APT1 malware implants (Bangat, Seasalt, Kurton, Auriga, etc. [1]). But it is also possible that there is no intrusion, and the above connection is the result of network security intelligence gathering, or it was generated by a known trusted application for some legitimate purpose. Each of these

hypotheses may explain the suspicious connection. The agent would need to automatically analyze each of these hypotheses to determine which of them is actually true. For this, it needs additional evidence which is obtained through the next process.

B. *Hypotheses in Search of Evidence*

As shown in the middle part of Fig.1, the agent puts each of the generated alternative hypotheses to work guiding the collection of relevant evidence. The question is: *Assuming that this hypothesis is true, what evidence should be observable?*

Fig.3 is an illustration of this process for the hypothesis "Connection 1 is part of APT1 intrusion using Bangat." This hypothesis is successively decomposed into simpler and simpler hypotheses, down to the level of elementary hypotheses for which it is clear what evidence to look for, and collection agents can be automatically invoked with specific search requests. In particular, if there is an APT1 intrusion using Bangat, then there should be activity attributable to APT1 on the Alpha network containing the computer with the IP 10.10.1.11, and the Bangat malware should be present on this host computer. Each of these sub-hypotheses is reduced, in turn, to specific indicators. The indicators for the first one are the possible detection of patterns of DNS resolution consistent with the TTPs of APT1, and the usage of other APT1 domains on the Alpha network. These, in turn, lead to the generation of specific search requests to be carried out by special collection agents, as indicated at the bottom of Fig.3.

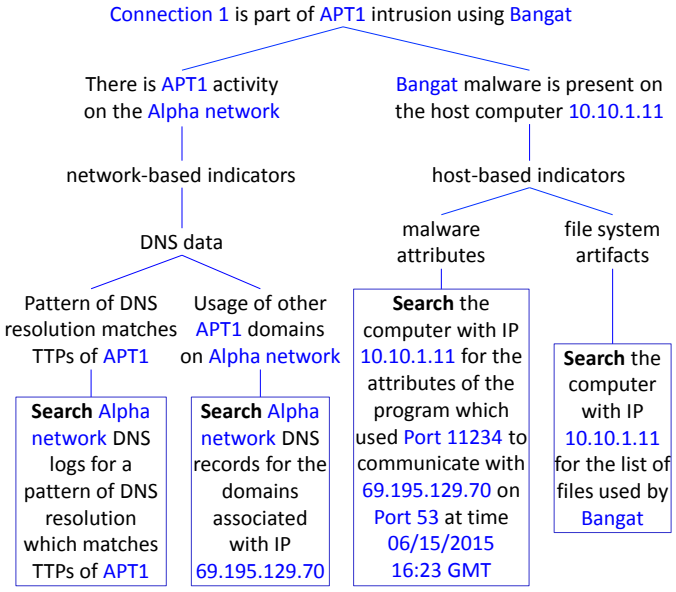


Fig. 3. Hypothesis in search of evidence.

1) *Context-dependent Reasoning*

Notice in the tree from Fig.3 that some words, such as APT1 and Bangat, appear in blue. This is because they are part of the agent's knowledge base, and are recognized by it.

Notice also that only some hypotheses are completely specified, while their sub-hypotheses are abstracted and understood in the context of their upper-level hypotheses. For example, "host-based indicators" is understood as "There are host-based indicators that the Bangat malware is present on the host computer 10.10.1.11." Similarly, "malware attributes" is understood as "The malware attributes are a host-based indicator that the Bangat malware is present on the host computer 10.10.1.11."

Context-dependent analysis enables a very succinct representation of the reasoning structures of the agent, helping the analyst to visualize a larger portion of it in the agent's whiteboard [10]. At the same time, the agent is aware of the complete representation of each hypothesis which is necessary for the learning and reuse of analytic expertise, as will be discussed in a follow-on section.

2) *Collection Agents*

The collection agents will return the found evidence as formal statements, each with its own *credibility* which represents the probability that the statement is correct [9]. For example, "Search the computer with IP 10.10.1.11 for the attributes of the program which used Port 11234 to communicate with 69.195.129.70 on Port 53 at time 06/15/2015 16:23 GMT" may return, among others, the following items of evidence:

[E4] ati.exe 10.10.1.11 made connection Connection 1 (credibility: certain)

[E5] ati.exe 10.10.1.11 is registered as a Windows Service with name iprip (credibility: certain)

[E6] ati.exe 10.10.1.11 has as unique Bangat string superhard corp. (credibility: certain)

In some cases, the collection agents may also return the *relevance* [9] of an item of evidence to a corresponding elementary hypothesis. For example, the result of "Search Alpha network DNS logs for a pattern of DNS resolution which matches TTPs of APT1" may return the following item of evidence:

[E1] pattern of DNS resolution partially matches TTPs of APT1 (credibility: certain, relevance: very likely)

In this case the collection agent is certain that there is a partial match, and the relevance of very likely expresses the degree of match.

This evidence is represented in the agent's knowledge base, and is used to estimate the probability of the top-level hypothesis from Fig.3, which is done through the next process.

C. *Evidentiary Testing of Hypotheses*

As shown in the right hand side of Fig.1, the agent uses the discovered evidence to test each hypothesis. Hypothesis testing is probabilistic because the evidence is always incomplete, usually inconclusive, frequently ambiguous, commonly dissonant, and with various degrees of credibility [9, 13].

As in Cogent [10], it is possible to use different assessment scales, but all are based on the same system of Baconian probabilities [14, 15] with Fuzzy qualifiers [16], where the values are on an ordered positive scale, as in the following "Probability" scale which is used in this paper:

lack of support < likely < very likely < almost certain < certain

In this case, there may be a lack of support from the available evidence to the considered hypothesis, or the evidence may indicate some level of support (e.g., likely).

Examples of other assessment scales are:

lack of belief < weak < moderate < strong < total belief

no strength < very low < low < medium < high < < very high < full strength

As evidence is returned by the collection agents, the corresponding parts of the tree in Fig.3 are regenerated, either to assess sub-hypotheses or to generate additional search requests, ultimately resulting in an evidence-based argumentation for assessing the top-level hypothesis, as illustrated in Fig.4.

Notice that each leaf hypothesis is directly assessed based on evidence, and these assessments are automatically combined, from bottom-up, based on the structure of the argumentation, to obtain the assessment of the top hypothesis. Let us consider the item of evidence "[E5] ati.exe 10.10.1.11 is registered as a Windows service with name iprip" from the bottom of Fig.4. Notice that the statement asserted by this item of evidence is precisely the elementary hypothesis to which it is attached ("ati.exe 10.10.1.11 is registered as a Windows service with name iprip"). Therefore, its *relevance* is certain. When the collection agent returns this item of evidence, it also returns its *credibility*, which, in this case, is also certain. The credibility and the relevance are combined (through the minimum function) to obtain the inferential force of the item of evidence on the elementary hypothesis (certain, in this case), as discussed in [9, 10].

Notice also "[E7] ati.exe 10.10.1.11 does not have as MD5 hash Bangat MD5 hash." This is disfavoring evidence for the hypothesis "ati.exe 10.10.1.11 has as MD5 hash Bangat MD5 hash," and therefore there is a lack of support for this hypothesis.

As indicated, the probabilities of the elementary hypotheses are combined, from bottom-up, based on the structure of the argumentation. For example, there are two favoring arguments for the hypothesis "ati.exe 10.10.1.11 has Bangat attributes":

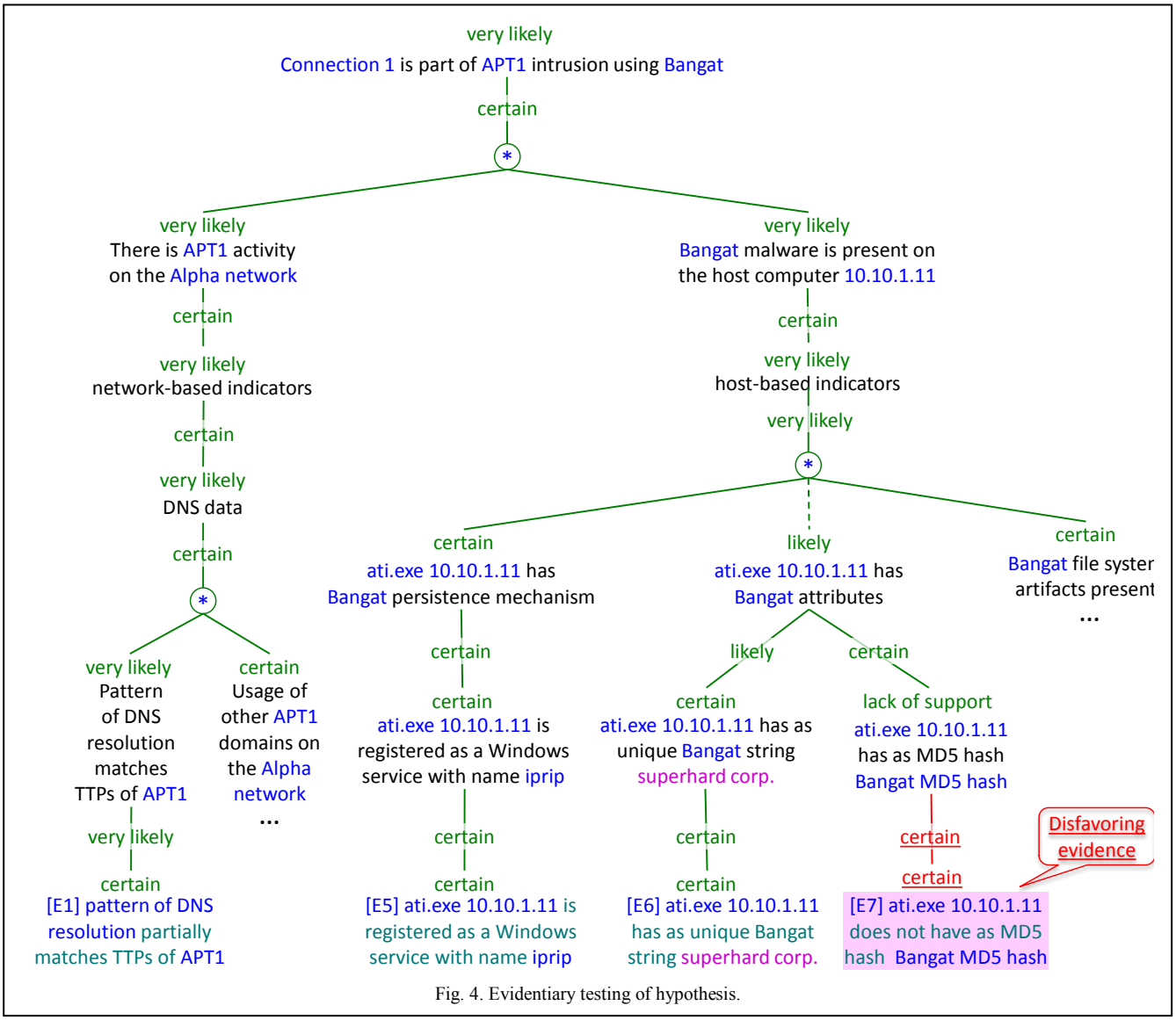


Fig. 4. Evidentiary testing of hypothesis.

IF "ati.exe 10.10.1.11 has as unique Bangat string superhard corp" THEN "ati.exe 10.10.1.11 has Bangat attributes" is likely

IF "ati.exe 10.10.1.11 has as MD5 hash Bangat MD5 hash" THEN "ati.exe 10.10.1.11 has Bangat attributes" is certain

The relevance of each argument (i.e., likely and certain, respectively) is combined with the probability of the corresponding sub-hypothesis (i.e., certain and lack of support, respectively), and the results are again combined, to produce an assessment of the probability of "ati.exe 10.10.1.11 has Bangat attributes": max(min(certain, likely), min(lack of support, certain)) = likely.

Further up in the argumentation are three possible host-based indicators of the presence of the Bangat malware on the host computer 10.10.1.11: persistence mechanism (P), malware attributes (M), and file system artifacts (F). Each indicator may be present with a certain probability, or it may not be present. The more indicators are present, the more relevant they are, collectively, to the presence of the Bangat malware. Fig.5 shows the relevance of each subset of these indicators.

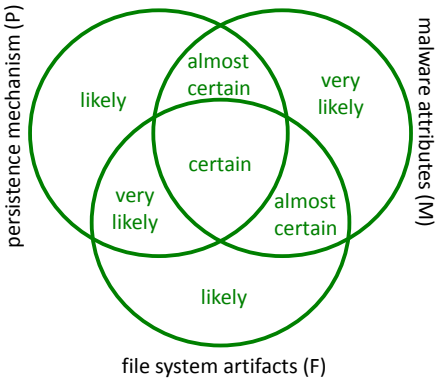


Fig. 5. Relevance of different subsets of indicators.

If all of them are present (i.e., the intersection of the three discs), then their relevance is certain. However, if only the persistence mechanism (P) is present, then its relevance is only likely. The actual probability of the "host-based indicator" hypothesis based on the three indicators (i.e., P, M, F) can be assessed by computing the inferential force for each possible combination of present indicators, and then selecting the maximum value. In this particular example, since all the three indicators are present, their relevance is certain, but the minimum of their probabilities is likely, which produces an overall assessment of likely. However, one obtains a higher assessment by ignoring the "malware attributes" indicator which the evidence indicates to be only likely. In such a case, the relevance of the other two indicators is lower (very likely), but they are both certain, giving an assessment of very likely to the "host-based indicators" hypothesis. In Fig.4, this "combined indicators" operator is marked with "*". Fig.4 also shows that the "Bangat attributes" indicator is ignored in this assessment by using an interrupted relevance link.

There are additional argument structures that are not illustrated by the argumentation from Fig.4. For example, a hypothesis may have an argument that consists of a conjunction of sub-hypotheses, in which case its probability is obtained as the minimum between the relevance of the argument and the probabilities of the sub-hypotheses.

For a hypothesis there may be both favoring and disfavoring arguments, and a hypothesis may have both favoring and disfavoring evidence. In such cases, the probability of the hypothesis is obtained by using an on-balance function which is initialized to a default set of values and automatically updated based on the values provided by the analyst when analyzing hypotheses.

III. Representation of the APT Detection Models

The previous section presented a systematic process of APT detection through evidence in search of hypotheses, hypotheses in search of evidence, and evidentiary testing of hypotheses (see Fig.1). To enable an agent to automatically perform this kind of reasoning, the knowledge of the APT detection models has to be formally represented.

We employ a learnable hybrid knowledge representation consisting of an APT ontology and reasoning tree patterns with ontology-based applicability conditions. The ontology language is an extension of RDFS [17, 18] with additional features to facilitate learning and evidence representation [19, 20, 21]. A fragment of the APT ontology, corresponding to the reasoning discussed in the previous section, is illustrated in Fig.6.

The middle left side shows the (partial) representation of Connection 1, the suspicious connection that triggered the APT1 detection process. The upper-right side represents some malware instances and concepts (e.g., APT1, Bangat, Seasalt, APT group). Under this fragment there is a partial representation of the network structure, showing the Alpha network and Corp_wkst_1, the computer used as host by the Bangat malware. The upper-left side shows some general concepts related to cybersecurity. The bottom right shows two features together with their domains and ranges. The bottom left of Fig.6 shows some of the evidence items returned by the collection agents as a result of executing the searches from the bottom part of Fig.3. Each evidence about a fact (e.g., "ati.exe 10.10.1.11 has as unique Bangat string superhard corp.") is labelled with its evidence name (i.e., [E6]), and has a certain credibility (not shown in Fig.6). Such facts are used to generate argumentation structures such as that from Fig.4, as will be discussed next.

Another component of the hybrid knowledge representation are the general tree patterns with ontology-based applicability conditions, such as that shown in Fig.7. The condition represents the semantics of the tree pattern and the context in which it can be applied to automatically generate specific tree structures. In particular, the tree pattern from Fig.7 will generate the search tree from Fig.3 if its condition is satisfied in the current situation, where V1 is instantiated to Connection 1, V2 is instantiated to APT1, and V3 to Bangat. The current situation is represented by the facts in the ontology from Fig.6.

IV. Mixed-Initiative Learning of APT Detection Models

Tree patterns, such as that in Fig.7, can be learned from specific examples of trees defined by cyber security experts, by employing a multi-strategy learning approach that integrates

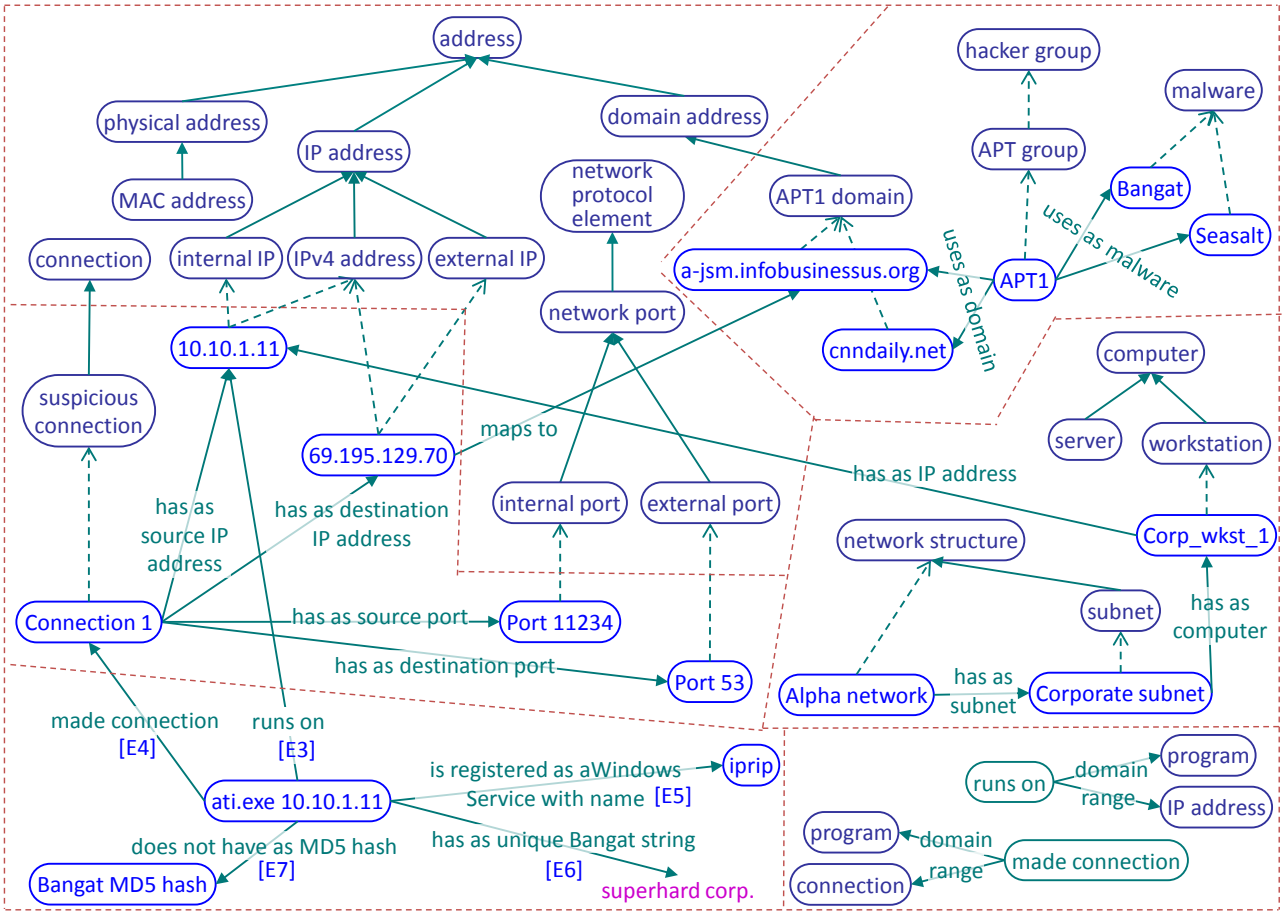


Fig.6. Ontology fragment.

several learning strategies including *learning from examples*, *learning from explanations*, and *learning by analogy and experimentation*, in a mixed-initiative interaction with the analyst, as presented in [20 - 24].

In essence, the expert will direct the agent to learn a tree pattern from a fragment of an argumentation, such as that shown at the bottom left of Fig.8. Then the agent interacts with the expert to determine the important features of the instances from the argument fragment. They include those that link the instances appearing in the top-level hypothesis (i.e., Bangat and 10.10.1.11) with the instances that appear only in the sub-hypotheses (i.e., ati.exe 10.10.1.11). Such a feature is "[E3] ati.exe 10.10.1.11 runs on 10.10.1.11." These and other potentially relevant features, such as "[E4] ati.exe 10.10.1.11 made connection Connection 1," are proposed by the agent and the relevant ones are selected by the expert. The upper left side of Fig.8 shows the identified relevant features together with the superconcepts of the instances in the APT ontology.

Next the agent automatically generates the tree pattern from the bottom right side of Fig.8. The tree pattern is obtained by simply replacing each instance (e.g., Bangat) with a variable (i.e., V1), and by removing the probabilities of the hypotheses. Direct relevance links, such as certain of "host-based indicator", are preserved in the pattern. The relevance of the combined indicators, such as those under the "host-based indicator" hypothesis, is generalized to a function.

The agent also automatically generates the applicability condition of the learned pattern, shown in the upper right side of Fig.8. Notice however that, instead of a single applicability condition (such as the one in Fig.7), there is an upper bound condition and a lower bound condition. They are obtained as maximal and, respectively, minimal generalizations of the instances and their important relationships, in the context of the APT ontology which is used as a generalization hierarchy.

As the agent learns new tree patterns from the cybersecurity expert, their interaction evolves from a teacher-student interaction, toward an interaction where they both collaborate on APT detection. In this case the agent automatically generates argumentation structures by applying the partially learned patterns and the expert critiques the reasoning, guiding the agent in refining its patterns. Correct argument structures generated by the agent lead to automatic generalization of the lower bound conditions of the used patterns. Any mistake identified by the expert leads either to the specialization of the upper bound condition of the responsible pattern, or to the addition of an except-when condition (with both an upper bound and a lower bound). The except-when conditions should not be satisfied in order for the pattern to be applicable. In time, the lower and the upper bound conditions of a pattern converge toward one another and to an exact applicability condition. The goal is to improve the applicability condition of the pattern so that it only generates correct argumentation fragments.

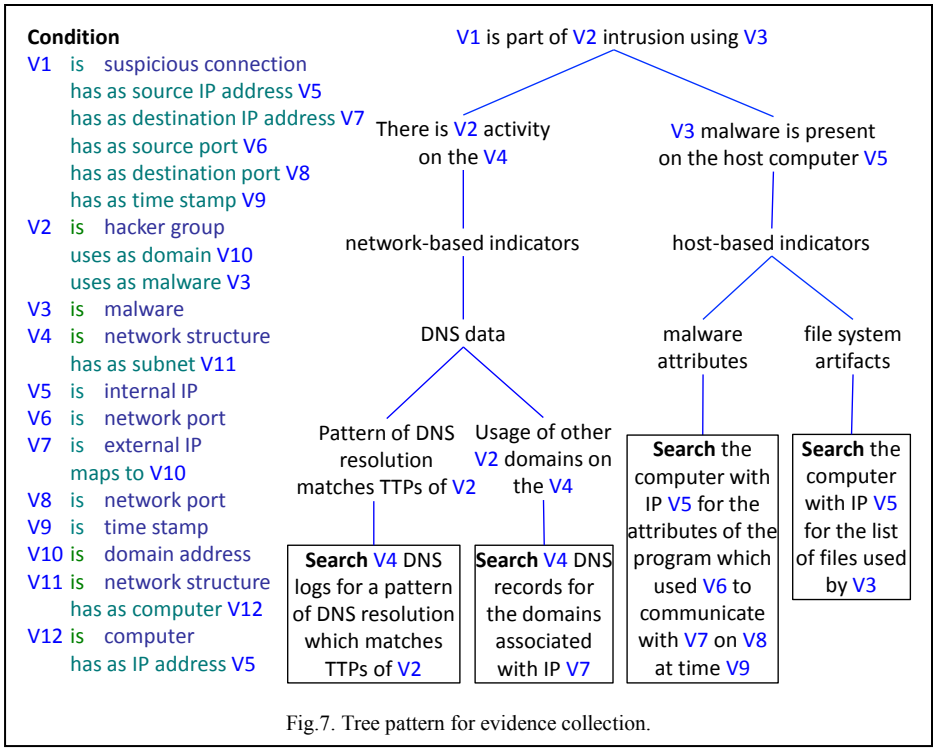


Fig.7. Tree pattern for evidence collection.

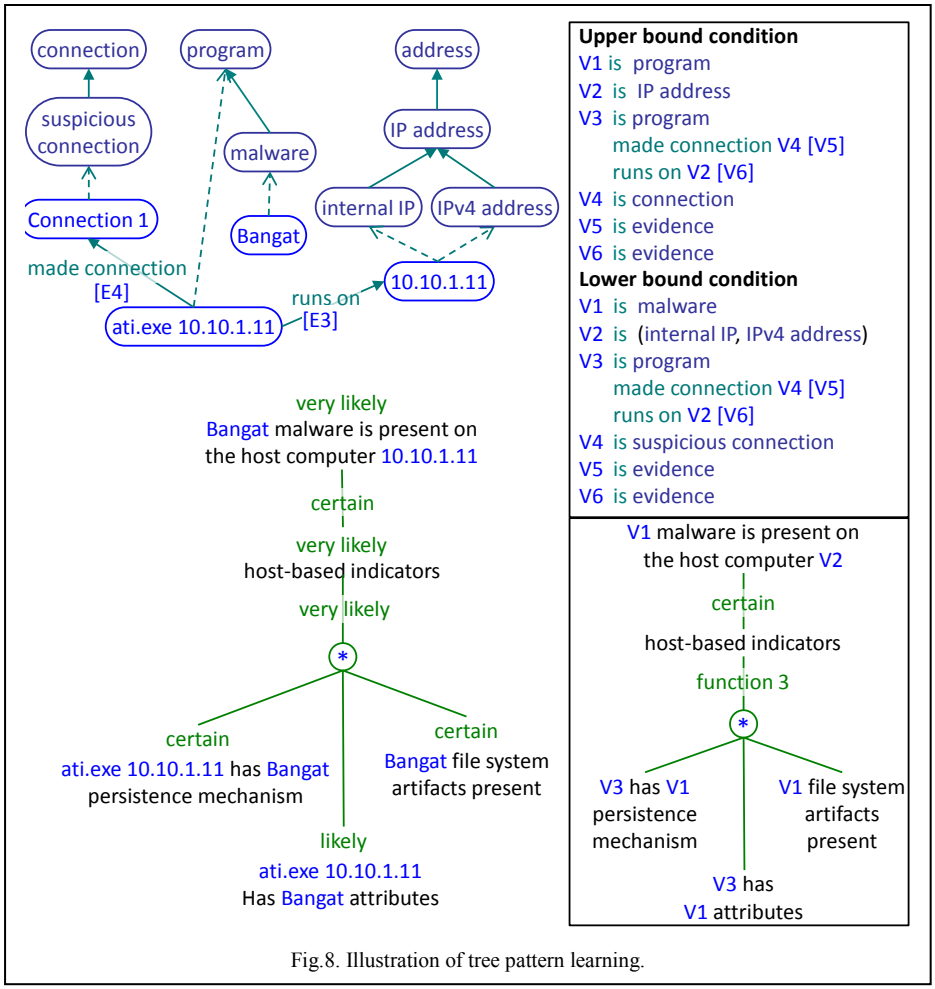


Fig.8. Illustration of tree pattern learning.

V. Learning Agent Shell

The developed APT detection theory and associated methods for knowledge representation, reasoning, and learning are being used to develop a prototype *learning agent shell* whose overall architecture is presented in Fig.9. This is a general agent development tool that contains general reasoning modules for the three APT detection processes from Fig.1, modules for development and refinement of the agent's ontology, modules for learning and refining the detection patterns, as well as modules for management of evidence, knowledge bases in the knowledge repository, and knowledge in knowledge bases. The learning agent shell will also incorporate a significant amount of general cybersecurity expertise for automatic APT detection into its general knowledge base, which is applicable in any CSOC. This includes an ontology, as well as tree patterns with ontology-based applicability conditions, for detection of a wide range of APT activities. The learning agent shell is used to rapidly generate a set of agents and their knowledge bases, all customized to a specific CSOC. They include a Hypotheses Generation Agent, several Automatic Analysis Agents, and several Mixed-Initiative Analysis Assistants.

The main customization of the knowledge base consists of populating it with a representation of CSOC's network, including its layout (see the middle right hand side of Fig.6), available sensors, operating systems used, and network security tools used.

The Hypotheses Generation Agent is a customization of the learning agent shell centered around the Hypotheses Generation module. It will generate hypotheses, such as those from the top part of Fig.2, from a variety of logs, network capture sensors, and intrusion detection devices and systems such as Bro [11] and Snort [12].

The Automatic Analysis Agents are all instantiations of the learning agent shell centered around the Automatic Analysis module. For each new generated hypothesis (e.g., "Connection 1 is part of APT1 intrusion using Bangat"), an instantiation of such an agent is automatically created to generate analysis trees such as those in Fig.3 and Fig.4.

Search requests generated by the Automatic Analysis Agents (see the bottom of Fig.3) are sent to a Collection Manager which manages and forwards them to corresponding Local Collection Agents. It also returns evidence (and its credibility) found by the collection agents to the corresponding automatic analysis agents.

Each Mixed-Initiative Analysis Assistant is an instantiation of the learning agent shell centered around the Mixed-Initiative Analysis module. It enables a specific CSOC analyst or operator to access the current state of the detection process in the form of a list of hypotheses of interest and their partial analyses. It also collaborates with the analyst who may update any of the analyses, provide additional evidence, or make several assumptions when no evidence is found. In the case of a new type of attack, the analyst and the mixed-initiative assistant can develop together an analysis tree from which new detection patterns are learned.

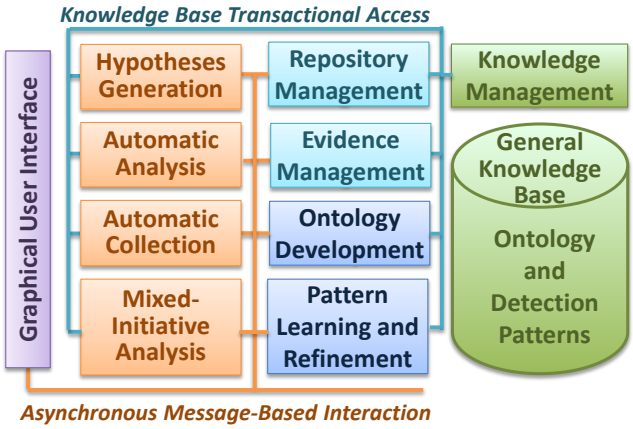


Fig.9. Learning agent shell for APT detection.

VI. Conclusions

We have presented current work on developing a semantic theory of cyber defense against advanced persistent threats, and using it to develop collaborative cognitive agents that generate defensible and persuasive analyses which show very clearly the argumentation logic, what evidence was used, and how. Such agents are integrated in cybersecurity operations centers to improve both the speed and the quality of the performed analyses, and to significantly increase the probability of accurately detecting intrusion activity while drastically reducing the workload of the CSOC operators. Moreover, this approach is developed to enable continuous learning from cybersecurity experts, based on evolving threat intelligence, to cope with new threats, thus providing high defense agility to the CSOC.

References

[1] Mandiant Intelligence. 2013. APT1: Exposing one of China's cyber espionage units, *Mandiant.com*.

[2] FireEye. 2015. APT30 and the Mechanics of a Long-running Cyber Espionage Operation, *FireEye,* April.

[3] Mila. 2013. Mandiant APT1 samples categorized by malware families, *contagio*, 03 March.

[4] Stewart J. 2003a. Sobig.a and the Spam You Received Today, *joestewart.org*, 08 Jul. http://www.joestewart.org/sobig.html, Accessed: 15 May 2015.

[5] Stewart J. 2003b. Sobig.e - Evolution of the Worm, *joestewart.org*, 08-Jul-2003. http://www.joestewart.org/sobig-e.html, Accessed: 15 May 2015.

[6] Zimmerman C. 2014. *Ten Strategies of a World-Class Cybersecurity Operations Center*. MITRE Corporation.

[7] Tecuci G., Marcu D., Boicu M., Schum D.A., Russell K. 2011. Computational Theory and Cognitive Assistant for Intelligence Analysis, in *Proceedings of the Sixth International Conference on Semantic Technologies for Intelligence, Defense, and Security – STIDS*, pp. 68-75, Fairfax, VA, 16-18 November.

[8] Tecuci G., Schum D.A., Marcu D., Boicu M. 2014. Computational Approach and Cognitive Assistant for Evidence-Based Reasoning in Intelligence Analysis, *International Journal of Intelligent Defence Support Systems*, 5(2):146–172.

[9] Tecuci G., Schum D.A., Marcu D., Boicu M. 2015. *Intelligence Analysis as Discovery of Evidence, Hypotheses, and Arguments: Connecting the Dots*, Cambridge University Press, to appear.

[10] Tecuci G., Marcu D., Boicu M., Schum D. 2015. COGENT: Cognitive Agent for Cogent Analysis. In the *Proceedings of the 2015 AAAI Fall Symposium "Cognitive Assistance in Government and Public Sector Applications"*, Arlington, VA, November.

[11] Bro home page http://www.bro-ids.org .

[12] Snort homepage https://www.snort.org/

[13] Schum D. A. 2001. *The Evidential Foundations of Probabilistic Reasoning,* Northwestern University Press.

[14] Cohen L. J. 1977. *The Probable and the Provable,* Clarendon Press, Oxford.

[15] Cohen L. J. 1989. *An Introduction to the Philosophy of Induction and Probability*, Clarendon Press, Oxford.

[16] Zadeh L. 1983. The Role of Fuzzy Logic in the Management of Uncertainty in Expert Systems. *Fuzzy Sets and Systems*, 11:199-227.

[17] W3C. 2004. http://www.w3.org/TR/rdf-schema/

[18] Allemang D. and Hendler J. 2011. *Semantic Web for the Working Ontologist: Effective Modeling in RDFS and Owl*, Morgan Kaufmann Publishers.

[19] Tecuci G., Boicu M., Ayers C., Cammons D. 2005. Personal Cognitive Assistants for Military Intelligence Analysis: Mixed-Initiative Learning, Tutoring, and Problem Solving. In *Proceedings of the 1st International Conference on Intelligence Analysis*, McLean, VA, 2-6 May.

[20] Tecuci G., Boicu M., Marcu D., Stanescu B., Boicu C., Comello J., Lopez A., Donlon J., Cleckner W. 2002. Development and Deployment of a Disciple Agent for Center of Gravity Analysis. In *Proceedings of the Eighteenth National Conference of Artificial Intelligence and the Fourteenth Conference on Innovative Applications of Artificial Intelligence*, AAAI-02/IAAI-02, pp. 853 - 860, Edmonton, Alberta, Canada, AAAI Press/The MIT Press.

[21] Tecuci G. Boicu M. Boicu C. Marcu D. Stanescu B. Barbulescu M. 2005. The Disciple-RKF Learning and Reasoning Agent, *Computational Intelligence,* 21(4):462-479.

[22] Tecuci G., Boicu M., Marcu D., Boicu C., Barbulescu M., Ayers C., Cammons D. 2007. Cognitive Assistants for Analysts, *Journal of Intelligence Community Research and Development*. Also in Auger J. and Wimbish W. eds. *Proteus Futures Digest*, pp.303-329, Joint publication of National Intelligence University, Office of the Director of National Intelligence, and US Army War College Center for Strategic Leadership.

[23] Tecuci, G., Boicu, M., Cox, M. T. 2007. Seven Aspects of Mixed-Initiative Reasoning: An Introduction to the Special Issue on Mixed-Initiative Assistants, *AI Magazine*, 28(2):11-18, Summer.

[24] Tecuci G., Marcu D., Boicu M., Schum D.A. 2015. *Knowledge Engineering: Building Personal Learning Assistants for Evidence-based Reasoning*, Cambridge Univ Press (to appear).

Genetic Counseling Using Workflow-based EMRs

Bo Yu*, Duminda Wijesekera*, Paulo Costa †, Sharath Hiremagalore*
*Department of Computer Science, George Mason University, Fairfax, VA, USA
{byu3,dwijesek,shiremag}@gmu.edu
†Department of Systems Engineering and Operations Research, George Mason University, Fairfax, VA 22030
pcosta@gmu.edu

***Abstract*—Widespread use of genetic tests for medical treatment and clinical genetic counseling–as a cost-effective treatment for an increasing number of hereditary disorders–has led to study of privacy and disclosure issues, and has compelled governments to limit disclosure of test results. To the best of our knowledge, no clinical workflows for genetic counseling apply applicable information disclosure laws have been documented and enforced in Electronic Medical Records (EMRs). To fill this void, herein we model a representative genetic counseling workflow and show how to simultaneously enforce privacy and informed consents in an open-source EMR. Our prototype provides workflow-guided counseling as well as consent management that enforces state and federal law-compliant genetic information sharing.**

***Index Terms*—Genetic Privacy, Privacy Laws, Electronic Medical Records (EMRs), Workflow Management Systems, Ontology**

I. Introduction

As genetics research advances, the list of predictable diseases is growing. For example, having been identified genetic mutations which associated with diseases include breast cancer, ovarian cancer, sickle cell anemia, etc. Studies have shown that preventive care costs significantly less than treatment upon diagnosis of a disease [18], [26]. Therefore, genetic tests, along with family history, are becoming a common practice in identifying risks of many hereditary conditions. clinical genetic services are the complex processes, usually involve genetic tests for finding gene mutations to make eventual disease onset predictable and Genetic counseling for explaining genetic test outcomes and suggest possible courses of action [12] to genetic tests requesters. Genetic test results are not only being used as indication basis for providing preventive and preemptive treatment for hereditary diseases, but also being broadly utilized for research purposes to discover more and more new findings. To compare with other medical researchers, researchers in genetic medicine need to use both genetic test results and their owners identifiable information, so more open accesses are required, e.g. using an opt-out consent that is much less rigorous in format for sharing data.

Genetic tests usually involve finding known changes - referred to as *mutations* - in a gene of a person that causes diseases. Researches have identified genetic mutations associated with various diseases such as breast cancer, ovarian cancer, sickle cell disease, β-thalassemia, left ventricular non-compaction cardiomyopathy (LVNC), and Alzheimer disease and many others. As new research on known gene mutations become available, and medically acceptable as indications, more diseases are added to the list for genetic tests that are available.

Several companies such as 23andMe [1], Gene by Gene [4], Color Genomics [3], and others, offer genetic tests and risk assessment services in the direct consumer market. Additionally, larger laboratories such as Myriad Genetics cater directly to health-care providers [8]. Information from these results are analyzed by professionals in genetic science who then provide counseling services to patients. With increased competition and lower costs of genetic tests [7], genetic counseling is going to play an important role in preventive care. This places genetic counselors in a critical path to explain the outcomes of genetic tests, and suggest possible courses of action [19].

Conducting genetic tests involves addressing ethical and privacy issues. Samples of human blood/tissue, and derived genetic information are able to precisely identify an individual and a group of related people that may be susceptible the same diseases as the original sample donor. Consequently, when genetic sequence information is shared without consent, lost, stolen, or used for a purpose other that which consent was obtained, the identity of a person is compromised. This information can be used by a third party to discriminate or, worse, harm the donor or a group of people. Prince et al. describe three practical genetic counseling cases that illustrate genetic discrimination [20]. Individuals may face discrimination in life, disability, and long-term care insurances. In other cases, when genetic information privacy is compromised, an individual may experience the stigma of having to carry a genetic marker for a disorder or disease.

Although, genetic tests have existed for a while, using genetic information for diagnosis and treatment is a part of a larger process that is being broadly termed as genetic counseling. Consequently, the precise processes (workflows) used by medical practitioners for genetic counseling is not very well defined. A good counter example is the workflow for hemodialysis where the a standard workflow for treatment is used by medical practitioners [24]. Although there is an increase in the bio-medical master degrees awarded by medical schools [14], less than 35 universities offer degrees with specialization in genetic counseling [2]. This results in difference in the process followed during genetic counseling. As a means of articulating different workflows currently emerging in genetic counseling and their larger usage in patient diagnosis and treatment, we have developed a prototype for genetic counseling with a flexible way of specifying and using

these workflows. In our prototype, we implement a workflow based on [17]. However, our tool can be easily modified to accommodate the changes in the workflow at a later stage.

The two US federal laws that regulate sharing of genetic information are Health Insurance Portability and accountability act of 1996 (HIPAA) [10], and Genetic Information Non-discrimination act of 2008 (GINA) [5]. HIPAA considers genetic information to be confidential medical information and regulates health-care providers. GINA regulates employers and health insurance companies but not health-care providers in using genetic information and protects individuals form discrimination based on genetic conditions. However, GINA does not apply to federal government employees or employers with fewer than 15 employees. These complex laws, in addition to the fragmented laws in each state, form the basis for information sharing and consent management workflows of our prototype system. In creating our prototype system, we also noticed significant regulatory gaps that create additional burdens in providing automated workflow-based guidance in genetic counseling.

Challenges. The following are the challenges for implementing a workflow-based EMR for genetic Counseling:

- Genetic Counseling is a new and an emerging field where the workflow has not been standardized, although providing a basis to do so would facilitate this emerging area and the mission of training genetic counselors.
- Genetic information collected for tests and their sharing have to conform to HIPAA [10] and GINA [5] regulations.
- State laws to protect Genetic information vary and add complexity to the system. HIPAA specifies that stricter sharing laws mandated by state regulation can override HIPAA policies.
- Although commercial systems may include Genetic Counseling in their packaged EMRs, it is difficult to verify their workflow as they are closed source.

Contributions. In order to address the above limitations of existing EMR systems, we present an end-to-end managed EMR prototype for genetic counseling that can accommodate the emerging workflows and diversity of state regulations (or lack thereof) in a re-programmable way. To the best of our knowledge, the proposed system is the first of its kind. Our working prototype has the following features:

- Automatically suggests Genetic Counseling for known disease codes
- Enforces a standardized work-flow for Genetic Counseling
- Automates paperless information sharing and medical treatment consent in accordance with local laws

The rest of the paper is organized as follows: Section II describes the closely related work in this area. In section III we describe the implementation of our system. Finally, we present our conclusions in Section IV.

II. Related Work

Electronic Medical records for Genetic Counseling

Electronic Medical Records (EMRs) plays a vital role of book keeping in the health-care industry. However, EMRs for genetic counseling present a unique set of challenges [13] as identified by Belmont et al. A major issue identified by the Belmont study the required uniformity in representing collected genetic data. Additionally, this study highlights the privacy, ethical and legal issues of handling genetic data in EMRs. Ours is a flexible freeware based platform to study these issues.

Scheuner et al. conduct a case study to verify if the current EMR systems meets genetic information needs [21]. This study involved results and conclusions gathered from discussion about 56 patient's electronic medical records with 10 EMR specialists, 16 medical geneticists, and 12 genetic counselors. An overall lack of support for functionality, structure, and tools for clinical decision making was an important finding.

A more recent study of the state of EMRs supporting genomics for personalized medicine again identifies structuring of data as a challenge [22]. The authors also identify clinical workflow management as a priority area that needs further research, development, and testing. Functionality, structure, and support for genetic information specific data is easily added to an EMR system. However, current EMR systems for genetic counseling still lack support for workflow enforcement and the ability to collect specialized genetic information sharing consents and enforce them on EMRs that contain the data.

Genetic Counseling ontology

The Gene Ontology Consortium has developed an ontology to store structured gene information in databases [16]. The ontology provides structured terms and vocabulary to store information regarding gene, gene products and sequences. The structure and terms developed by this team lacks support for capturing the terms used for informed consent requirements laid out by law. In a more closely related work, authors in [26] describe an ontology for treatment consent. The ontology presented is, however, insufficient for capturing the terms and vocabulary used in information disclosure consent. In this work, we develop the ontology containing the structure and terms required for information disclosure consent.

Genetic Counseling informed consent

Obtaining informed consent for diagnosis (including testing) and treatments is a very well studied area and a mandatory requirement on care providers. In particular, care providers are required to obtain informed consent for genetic counseling as much as any other treatment. A preliminary study convened jointly by the National Institute of Health (NIH) and the Center for Disease Control (CDC) [15] presents the risks and ethical issues involved in collecting and storing tissue samples for genetic tests in a research setting. Ethical and legal issues are similarly present in a clinical genetic counseling process.

Authors in [25] present an automated and paperless informed consent management system for medical treatments.

This work provides a generic framework to enforce informed consent for minors. The work presented in [25] only enforces general treatment consent. This work does not specifically address the additional issues related to genetic counseling and sharing of genetic information. Genetic counseling requires enforcement of information disclosure, research and sample and genetic information retention consents in addition to treatment consent. The objective of this work is to create a workflow that enforces all types of mandated informed consent requirements in the genetic counseling process to comply with local, state and federal regulations.

III. System

This section presents details of our prototype genetic counseling EMR system within a sample enforced workflow. We first describe genetic counseling workflow and the challenges involved in modeling the process. Then we present a model of this workflow using a work-flow engine. Next, we describe the different types of informed consent for genetic counseling. We then collate the federal and state laws that regulate disclosure of genetic information. Then we describe how we enforce the disclosure consent for genetic counseling to comply with various state and federal laws. Lastly, we describe how we integrate the workflow enforcement with an open-source Medical Record System (OpenMRS).

A. Genetic Counseling Workflow

The Genetic Counseling Definition Task Force defines *Genetic counseling* as *The process of helping people understand and adapt to medical, psychological, and familial implications of genetic contributions to disease*. A team of health care workers involving, but not limited to, Medical Practitioners, Bio-curators, Genetic Counselors, Molecular Pathologists, Medical Geneticists play an important role in Genetic Counseling. Similar to other medical procedures, genetic Counseling follows a well defined protocol. This protocol involves several tasks performed by the caregivers in a particular sequence called the workflow.

Although genetic counseling has been around for a while, we were unable to find a documented and modeled workflow for this process. Therefore, we present a detailed documentation for clinical genetic counseling. We model our workflow based on a presentation about clinical genetic counseling from Stanford's Clinical Genomics program [17]. In addition to the previous presentation, we refer to another presentation by O'Daniel et al. on Genomic Medicine [18] for additional details. Here we provide a detailed documentation and overview of the Genetic Counseling workflow. Figure 1 shows the workflow for genetic counseling modeled using a workflow editor, created using an open-source workflow system YAWL [23].

Our modeling of the workflow is tailored to genetic counseling for Hereditary Cancer Syndromes. However, this can be modified to accommodate any other genetic disorder. The following list walks through the details of the individual tasks involved in the process of providing genetic counseling to a patient. The following list focuses on describing individual tasks involved in the genetic counseling process. We present the details of consent management later in section III-B.

1) **Collection of Medical Records, Family History, and Social History:**
   - *Require Genetic Counseling:* A genetic counseling case originates with a physician's referral for a genetic counseling. Alternatively, in other cases a patient self requests genetic counseling after learning of a manifestation of a genetic disorder in a family member.
   - *Patient Walk-in:* In the case where a patient self requests genetic counseling, upon arriving at the genetic counselors office, the patient is asked to provide social and family history. Additionally, if copies of medical records are provided, they are recorded in the system. In our workflow model, the patient is presented with a detailed family and social history questionnaire to assess the risk of a suspected Hereditary Cancer Syndrome. We model this questionnaire based on forms developed by the Virginia Women's Center [11].
   - *Physician's Referral:* When a physician refers a patient for genetic counseling, the patient is asked to sign release forms for medical records from the referring physician (if not already provided). Else, if the patient brings in copies of medical records, they are recorded in the system.
   - *Requirements Review:* Information collected from the patient is reviewed for completeness. If any, missing information is collected from the patient, patient's primary care physician or public sources of ancestry information. This is shown in the 'Collect More Information' process in the modeled workflow (Fig. 1). Once all of the information (to the best of patients knowledge) is collected, one or more of following people review the collected information: Genetic Counselor, Molecular Pathologist, Medical Geneticist. Next, they verify if a genetic test is available to answer the questions posed by the patient or the treating physician.
2) **Pre-test counseling:** Once an insurance or payment authorization is obtained, the patient meets with a genetic counselor for pre-test counseling. The genetic counselor performs the following sub-tasks for pre-test counseling:
   - An informed consent is obtained for counseling, i.e., Genetic Counseling Treatment Consent. As a part of this informed treatment consent, the counselor provides the patient with information about the risks and benefits of this process.
   - The family, social, and medical history provided by the patient is reviewed.
   - The counselor discusses the expected range of results and their impact on the patient and their relatives. Additionally, the counselor discusses other potential incidental findings that may be in the

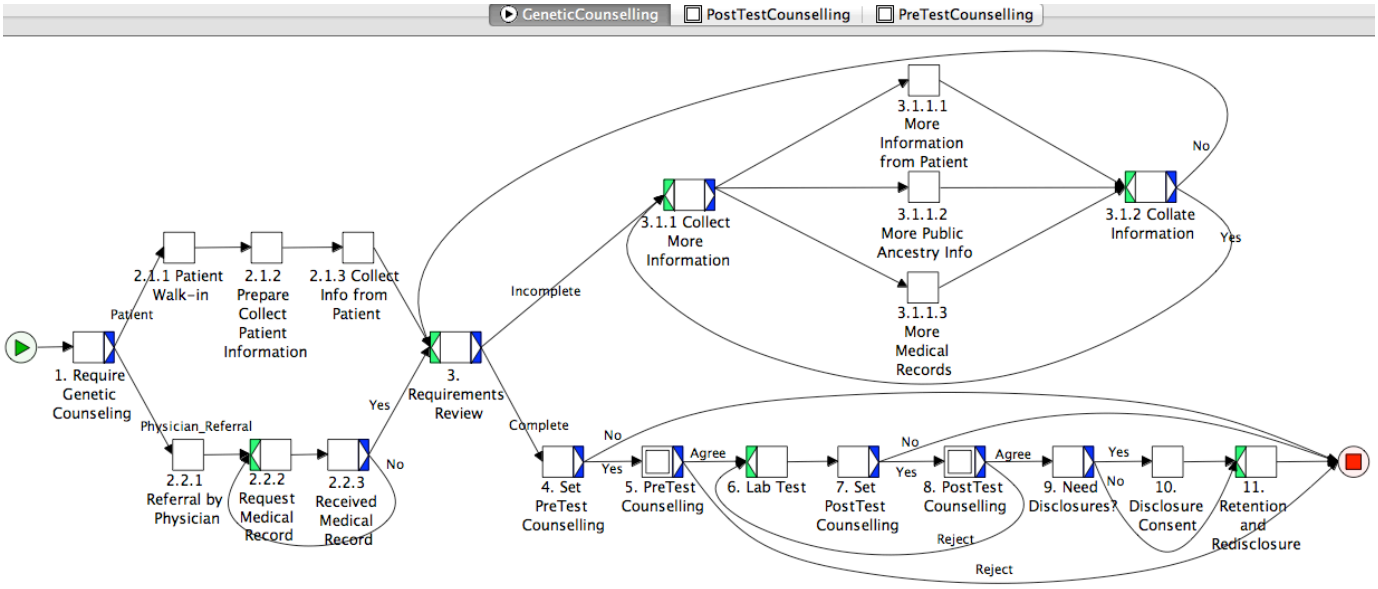


Fig. 1: Genetic Counseling Workflow

report. Incidental findings are those findings in the test results that are not associated with the primary reason for conducting the test, but indicative of other genetic disorders. The counselor also discusses ethical issues that could arise from conducting or disclosing these test results. Before ordering the tests, any questions or clarifications from the patient are answered.

- If applicable, consent for sharing information for research purposes is collected.

3) **Lab work:** Blood or tissue samples are collected from the patient for genome sequencing. Generally, 1 tube of blood is collected for sequencing. An additional tube may be collected for confirmatory studies. Another tube of blood may be collected from the patient with appropriate consent for research purposes. The blood and tissue samples are then sent to a laboratory for sequencing and the sequenced genetic information is returned to the genetic counseling team. We do not model the workflow for genome sequencing in the laboratory in this paper.
4) **Post-test counseling:** Once the sequencing results are reported back to the genetic counseling team, a draft report is prepared. This draft report is reviewed by a review team consisting of Genetic Counselors and/or Medical Geneticists, Physicians, and/or content experts. Once reviewed, the final report is uploaded into the EMR and is ready for the patient. When the patient meets the counselor to discuss the test results, the following subtasks are performed:
   - The test results are reviewed by the counselor with the patient. Once the review is complete, the counselor assess the comprehension and coping skills of the patient.
   - If the additional evaluation or referral for treatment (e.g. prophylactic mastectomy) is warranted, referrals or orders are issued by the genetic counselors. Alternatively, the patient may decline to accept the recommendations of the genetic counselor. If the recommendations are declined, it is recorded in the EMR.
   - The counselor finally discusses implications of the findings on related family members. At this point, the genetic counselor may recommend the patient to personally inform related family members about the test results and treatment options by providing templates. Alternatively, they may obtain consent to disclose test results to affected family members.
   - Lastly, if required, the counselor may schedule follow up appointments .
5) **Disclosure and other Consent:** At the end of the genetic counseling process, the care provider may be required to obtain consent to retain samples and/or genetic information (sequencing results) and genetic test reports based on state and federal regulation. Additionally, the care provider may require additional disclosure consents to share information with other individuals or entities as required by law. If consent to retain genetic samples and test information is denied, they must be destroyed and/or purged from the EMR records. Once these housekeeping tasks are complete, the genetic counseling workflow comes to an end.

B. *Informed Consent in Genetic Counseling*

As discussed earlier, if a proper workflow is not enforced for genetic counseling in the EMRs, it may leave health-care providers and/or health-care organizations open to lawsuits. Existing laws limiting genetic information disclosures do not

generally consider ethical issues involved during the disclosure process. Ethical issues present an abstract concept that need to be carefully considered by genetic counselors on a case-by-case basis. Additionally, an organization's ethical guidelines my regulate disclosure of genetic information, which upon disclosure may tangibly affect a third party. Lack of clear federal and state legislation covering ethical issues and individual Organizational ethics boards having different requirements makes it difficult to model them as workflows as of this writing. Therefore, we leave documenting the workflow for ethical disclosures as future work.

One stark difference between other medical treatments such as eye surgery, or hemo-dialysis is that the patient can stop genetic counseling at any stage of the workflow. Patients are permitted to revoke a consent at in point during the counseling process. Once a patient revokes a consent or declines to continue treatment, the care provider is required to document this in the EMR. Workflow in Figure 1 allows patients to revoke or decline treatment at specific check-points in the workflow.

In this part we focus on enforcing informed consent requirements stipulated by state and federal laws. Managing and documenting informed consent during the genetic counseling process is different from other general treatments. A patch work of state and federal laws and regulations to be followed by the care provider in each state adds additional complexity to the workflow of genetic counseling. The following types of informed consent is required for Genetic Counseling:

- **Treatment Consent:** Treatment consent for genetic counseling helps the patient understand the risks, benefits, and limitations of genetic counseling and are modeled in our workflow. Treatment consent regulation varies greatly among states, the details of which are captured in [25]. This consent is different from generic treatment consent as the test results may have the following outcomes: Positive, Negative, and Uncertain. For example, if a patient is being tested for Hereditary Breast Cancer markers BRCA1 or BRCA2, the results may be positive for mutations that increase the risk of breast cancer. The results may be negative for known mutations or it may contain unknown mutations that have not been studied. These test results are classified as medically actionable (e.g. prophylactic mastectomy) or in-actionable (no treatment is available). A separate consent is recommended for finding incidental mutations that may be actionable or in-actionable. Consent is also required for collecting blood and tissue samples from the patient prior to conducting a test. State laws provide for exemptions to requiring treatment consent in certain cases such as Paternity tests.
- **Information Disclosure Consent:** Information disclosure consent is regulated at both state and federal level. HIPAA, GINA, and individual state laws such Delaware Code §16.2.120 - §16.2.1227, protect patients from unauthorized disclosure of genetic information and test results. Laws usually require that the patient is provided with information about the type of information being disclosed and the name of the entity to which it is disclosed. In certain cases, the reason for disclosure may also be provided. HIPAA, and state laws provide for exemptions from this type of consent (e.g., Identification of bodies). Certain states also require that a consent be obtained for each instance of disclosure called re-disclosure consent. We model these exemptions and re-disclosure consent in our workflow.
- **Research Consent:** In most cases an informed consent is required from the patient to conduct a research. Research consents explains how the genetic information and samples collected would be used in the study and the potential outcomes of the study. They may also specify if the patient will be informed about any medically actionable findings from the research study and the extent of information that will be shared with the participants. Research consents are mostly governed by Internal Review Boards of individual organizations. Most organizations require a research consent for genetic research as a best practice. However, certain states exempt research organizations from research consent were only de-identified information is collected and/or disclosed as a part of the research.
- **Retention Consent:** Retention consent pertains to retaining blood or tissue samples and genetic information or test results once the test is complete and results are shared with the patient. Some states require blood and tissue samples be destroyed at the end of the test. Since blood and tissue samples are collected and sent to the laboratory for processing we do not capture retention consent in this particular case in our workflow. We capture in our workflow, cases where states require a retention consent for storing genetic information or test results. If retention consent is denied, information and test results are purged at the end of the genetic counseling process.

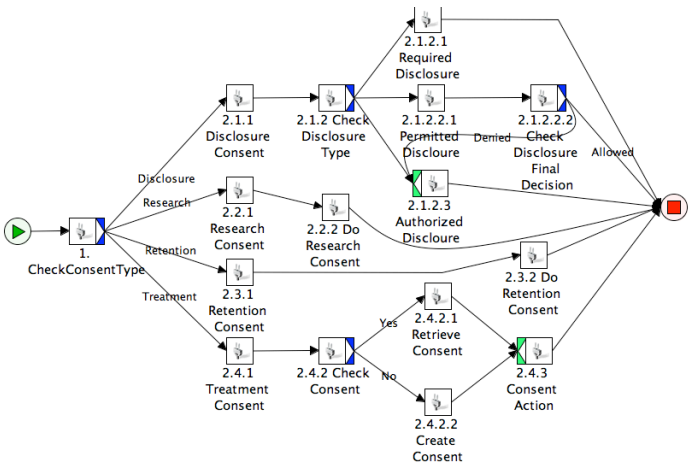


Fig. 2: Consent Workflow

1) Consent Management in the Workflow: In this section, we describe how we enforce the required consents in the genetic counseling workflow. Although we focus on the enforcement of disclosure consent, other types of consent may

be enforced in the workflow very similarly to the disclosure consent.

Figure 2 shows the workflow specification for the different types of consent modeled in YAWL. The consent workflow runs as a separate service in the workflow engine. When a task requires consent in the genetic counseling workflow, the consent workflow is invoked to manage the consent requirements. In the case of information disclosure consent, the workfow checks the purpose for the information request. HIPAA classifies these purposes into different disclosure categories described later in this section. If the disclosure type requires a consent, the consent workfow enforces this requirement. For simplicity, our modeled workflow enforces retention and research consent for all states as it is a best practice. For a detailed working of the treatment consent tasks, refer to [25].

In order for the workflow to enforce all of the above consents for genetic counseling, we use a rule base to codify laws. We do so by first modeling the term structure we use to specify rules and then use OWL Description Logic (DL) to specify these rules using the terms created in our ontology. We use the open source ontology editor Protégé and the Pellet reasoner to create the rules and specify the rules respectively. During the counseling process, the workflow engine YAWL invokes the rule bases's run-time to determine the required consents and generates the required consent forms that are displayed to the patient. Once the patient signs the forms, the workflow proceeds to the next step, thereby enforcing federal and applicable state laws during the counseling process.

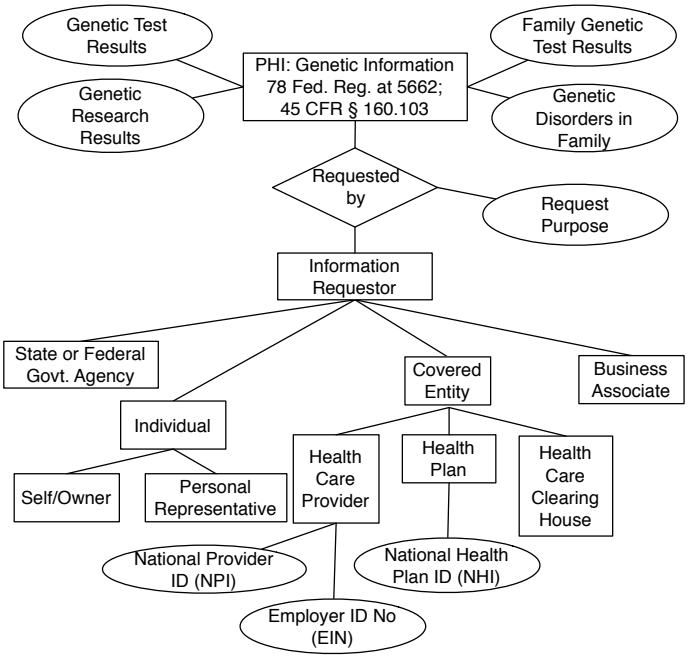


Fig. 3: Entity-Relationship Diagram for Genetic Information

In order to model the terms structure for genetic information disclosure, we start by identifying the relationship between the information requester and Protected Health Information (PHI) as defined by HIPAA. Figure 3 shows the ER diagram for information disclosure consent in genetic counseling. The figure shows the *requests* relationship between an *information requester* and *Genetic Information*. Genetic Information is considered as Protected Health Information (PHI) according to HIPAA. Definitions pertaining to genetic information are included in detail in this legislation. An individual's genetic information is defined to include the following:

- Genetic test results of the patient
- Genetic research results of the patient
- Genetic test results of a related family member
- Information about manifestation of a genetic disorder in a family member

HIPAA permits or denies information disclosure of PHI and states the informed consent requirements based on the type of information requester and the purpose for which the information is being requested. HIPAA further classifies an information requester as State/Federal Agencies, Individuals, Covered Entities, and Business Associates. These sub types are shown in Figure 3. Based on type of the requester requesting access to genetic information, HIPAA classifies request purpose into the following categories:

- **Required Disclosures:** Required disclosures do not require any consent from the patient and are required to be disclosed. E.g. HHS Compliance Review
- **Permitted Disclosures:** These disclosures are permitted by law without consent. However, an organization may choose to implement policies and procedures requiring additional consent requirements. Examples of permitted disclosures include Treatment, Health care Operations, Public Interest and Beneficial Activities, etc.
- **Authorized Disclosures:** Authorized disclosures require informed consent from the patient, such as for General Research, Marketing, Psychotherapy Notes, etc.

The complete set of Privacy Rules were obtained from HIPAA located at 45 C.F.R. Part 160. These rules are then written into the Pellet reasoner for enforcing information disclosure consent requirements. In addition to the federal information disclosure requirements described in HIPAA. State laws regulate genetic information disclosure. They may have additional or fewer requirements when compared to HIPAA. For example, Delaware has a comprehensive code on regulating disclosure of genetic information and test results. On the other hand, states such as Alabama have no specific legislation.

In 2008, only 27 of the 50 states had specific laws requiring consent to disclose genetic information [6]. As of 2014, this number had risen to 35 [9]. Other states are acting swiftly to protect address the issue of genetic information privacy of its citizens. We gather all pertinent state laws governing genetic information passed until 2014 from [9]. Table I shows genetic information disclosure laws for a sample of selected states. These laws were then transformed into a simple algorithm to permit or deny disclosures of genetic information. Finally, these algorithms were translated into rules to enforce a state laws in the reasoner as described above.

State	Informed Disclosure Consent Law	Required/Permitted Disclosures
Alabama	N/A	N/A
	...	
Delaware	- Unlimited access by subject to their own genetic information - Disclosures are to be authorized by obtaining informed consent of the tested individual describing the information to be disclosed and to whom	- Disclosure is necessary for the purposes of a criminal or death investigation or a criminal or juvenile proceeding or to protect the interests of an issuer in the detection or prevention of fraud, material misrepresentation or material nondisclosure - Disclosure is necessary to determine paternity - Disclosure is authorized by order of a court of competent jurisdiction - Disclosure is made pursuant to the DNA analysis and data bank requirements of §4713 of Title 29 - Disclosure is for the purpose of furnishing genetic information relating to a decedent for medical diagnosis of blood relatives of the decedent - Disclosure is for the purpose of identifying bodies - Disclosure is pursuant to newborn screening requirements established by state or federal law - Disclosure is authorized by federal law for the identification of persons - Disclosure is by an insurer to an insurance regulatory authority - Disclosure is authorized in accordance with §1201(4)d. of this title - Disclosure is otherwise permitted by law
	...	
Florida	Informed consent is required to disclose genetic test results	Public entities are exempt from disclosure restrictions. Pursuant to Florida Statutes §119.07(1) and Statutes 42(a), Article 1 of the Florida Constitution
	...	
Wyoming	N/A	N/A

TABLE I: State Laws limiting Genetic Information Disclosure

Flowchart 4 shows how genetic information disclosures are allowed or denied in a based on organizational policies, State, and Federal laws. When genetic information or test results are requested, information about the requester, Request purpose, and genetic information are collected in the EMR system. These are then passed to the modeled workflow, which uses the reasoner to assess whether a consent is required. The workflow then triggers the EMR system to display the appropriate consent forms. When both state and federal laws exist, HIPAA resolves eventual contentions by providing precedence to the law that does not require a consent for information disclosure. This precedence rule is implemented in the reasoner. The procedures modeled in flowchart 4 allow for individual organizational policies in the information disclosure process. However, our implementation does not model any organizational information disclosure policies.

2) Workflow enforced EMR: In the last step, we modify the source code of OpenMRS, an open source Electronic Medical Record System, to provide specialized interface for genetic counseling. We add JSP scripts to OpenMRS to generate treatment, information disclosure, and research consent. The EMR communicates with the YAWL run-time which in-turn communicates with the reasoner to complete our prototype system. Our system is able to enforce the genetic counseling workfow and consent requirements entailed in these workflows in accordance with state and federal laws.

IV. Conclusions

In order to model the genetic counseling workflow, we present and document the details of the sequence of tasks performed by the care providers during genetic counseling process. We also study and extract rules from federal and state laws that limit the disclosure of genetic information. We use these rules to implement a workflow-enforced Electronic Medical Record System tailored for working with Genetic Counseling.

To the best of our knowledge, our work presents a first working open-source prototype EMR for genetic counseling. Our genetic counseling EMR supports automatic paperless enforcement of treatment consent, information disclosure consent, research consent, and retention consent. This workflow enforced genetic counseling EMR would enable genetic counseling services provided by care providers and health care organizations to comply with state and federal laws concerning genetic information privacy. As a result, our EMR saves care providers and health care organizations from unnecessary litigation that would arise when proper procedures are not followed. Additionally, the electronic audit trail left by our EMR would help care providers and health care organizations during eventual litigations.

References

[1] 23andme inc. https://getcolor.com/#/. Accessed: 2015-04-27.

[2] Accreditation council for genetic counseling - accredited universities. http://gceducation.org/pages/accredited-programs.aspx. Accessed: 2015-05-04.

[3] Color genomics inc. https://www.23andme.com/. Accessed: 2015-04-27.

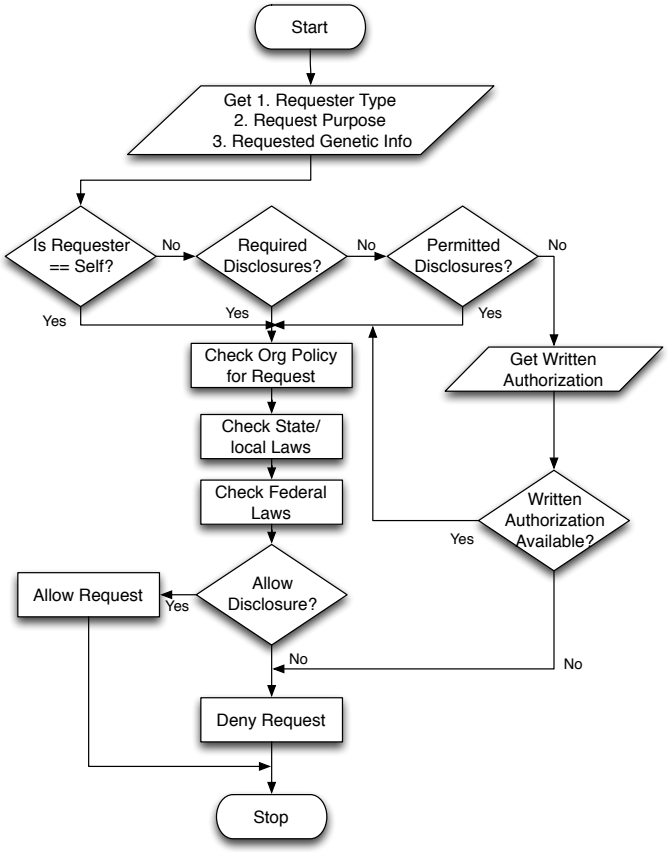


Fig. 4: Flowchart for allowing or denying Genetic Information Disclosures

[4] Gene by gene, ltd. https://www.genebygene.com/. Accessed: 2015-04-27.

[5] 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.

[6] Genetic privacy laws - national conference of state legislatures. http://www.ncsl.org/research/health/genetic-privacy-laws.aspx. Accessed: 2015-05-25.

[7] Genetic test for breast cancer becomes more affordable. http://www.theverge.com/2015/4/21/8458553/color-breast-cancer-gene-testing-brca-myriad. Accessed: 2015-04-23.

[8] Myriad genetics inc. https://www.myriad.com/. Accessed: 2015-04-27.

[9] Privacy issues in the sharing of genetic information. http://www.personalizedmedicinebulletin.com/wp-content/uploads/sites/205/2014/09/PrivacyIssuesintheSharingofGeneticInformation.pdf. Accessed: 2015-05-25.

[10] The Health Insurance Portability and Accountability Act of 1996 (HIPAA). *Pub. L. 104-191*, 110 Stat. 1936, codified as amended at 42 U.S.C §300gg and 29 U.S.C §1181 et seq. and 42 U.S.C §1320d et seq.

[11] Virginia women's center - cancer family history questionnaire. https://www.virginiawomenscenter.com/assets/HEREDITARY_CANCER_RISK_ASSESSMENT_FORM.pdf. Accessed: 2015-05-22.

[12] What is genetic counseling and do i need a genetic counsellor? http://www.nhs.uk/chq/pages/2370.aspx. Accessed: 2015-04-23.

[13] J. Belmont and A. L. McGuire. The futility of genomic counseling: essential role of electronic health records. *Genome Med*, 1(5):48, 2009.

[14] G. Blanck. The rise of the biomedical sciences master's program at us medical colleges. *Teaching and learning in medicine*, 26(4):409–411, 2014.

[15] E. Clayton, K. Steinberg, M. Khoury, and et al. Informed consent for genetic research on stored tissue samples. *JAMA*, 274(22):1786–1792, 1995.

[16] G. O. Consortium et al. The gene ontology (go) database and informatics resource. *Nucleic acids research*, 32(suppl 1):D258–D261, 2004.

[17] M. J. Establishing a clinical genomics program at an academic medical center. https://www.utoledo.edu/med/depts/path/pdfs/Merker%20APC_meeting_1014.pptx. Accessed: 2015-05-25.

[18] J. O'Daniel, J. Scott, E. Varga, V. Speights, and E. Gordon. Lecture iv: Genomic medicine: Communicating with the patient. http://www.pathologytraining.org/pdfs/uscap_trig/Lecture%20IV%201-12aHOrh.pptx. Cached on Google.

[19] J. M. ODaniel. The prospect of genome-guided preventive medicine: a need and opportunity for genetic counselors. *Journal of genetic counseling*, 19(4):315–327, 2010.

[20] A. E. Prince and M. I. Roche. Genetic information, non-discrimination, and privacy protections in genetic counseling practice. *Journal of genetic counseling*, 23(6):891–902, 2014.

[21] 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):510–517, 2009.

[22] M. H. Ullman-Cullere and J. P. Mathew. Emerging landscape of genomics in the electronic health record for personalized medicine. *Human mutation*, 32(5):512–516, 2011.

[23] W. M. P. van der Aalst and A. H. M. ter Hofstede. Yawl: Yet another workflow language. *Inf. Syst.*, 30(4):245–275, June 2005.

[24] B. Yu and D. Wijesekera. Building dialysis workflows into emrs. *Procedia Technology*, 9:985–995, 2013.

[25] B. Yu, D. Wijesekera, and P. Costa. Consent-based workflow control in {EMRs}. *Procedia Technology*, 16(0):1434 – 1445, 2014. {CENTERIS} 2014 - Conference on {ENTERprise} Information Systems / ProjMAN 2014 - International Conference on Project {MANagement} / {HCIST} 2014 - International Conference on Health and Social Care Information Systems and Technologies.

[26] B. Yu, D. Wijesekera, and P. Costa. An ontology for medical treatment consent. *Proceedings of the Ninth Conference on Semantic Technologies for Intelligence, Defense, and Security (STIDS 2014), Fairfax VA, USA*, 1304, 2014.

Controlled and Uncontrolled English for Ontology Editing

Brian Donohue
CUBRC
Buffalo, NY

Robert Ganger
CUBRC
Buffalo, NY

Tien Pham
Army Research Lab
Adelphi, MD

Amardeep Bhattal
IBM
Southampton, UK

Barry Smith
University at Buffalo
Buffalo, NY

Douglas Kutach
CUBRC
Buffalo, NY

Ron Rudnicki
CUBRC
Buffalo, NY

Geeth de Mel
IBM
White Plains, NY

Dave Braines
IBM
Portsmouth, UK

***Abstract*—Ontologies formally represent reality in a way that limits ambiguity and facilitates automated reasoning and data fusion, but is often daunting to the non-technical user. Thus, many researchers have endeavored to hide the formal syntax and semantics of ontologies behind the constructs of Controlled Natural Languages (CNLs), which retain the formal properties of ontologies while simultaneously presenting that information in a comprehensible natural language format. In this paper, we build upon previous work in this field by evaluating prospects of implementing International Technology Alliance Controlled English (ITA-CE) as a middleware for ontology editing. We also discuss at length a prototype of a natural language conversational interface application designed to facilitate ontology editing *via* the formulation of CNL constructs.**

Keywords—Ontology; Controlled English; Intelligence Collection

I. INTRODUCTION

Ontologies formally represent reality in a way that limits ambiguity and facilitates automated reasoning and data fusion. Many technologies are available for building, sharing, and using ontologies, including Web Ontology Language (OWL) and controlled natural languages (CNLs). On the one hand, OWL provides effective representation constructs and enables efficient reasoning procedures but is daunting to the non-technical user. On the other hand, CNLs, which are restricted versions of natural languages, provide a human-friendly representation format that is easier for non-technical users but there is no established standard for how statements of CNLs should map onto assertions defining an ontology.

Motivated by the accessibility of CNL, we explored how to create software infrastructure that would enable users to interact with an OWL knowledge base through CNL constructs. We see the value of such an infrastructure for the intelligence, defense and security communities as being realized in the use of ontology-driven information collection applications. Such applications typically have one of two opposing shortcomings. First, users can be prevented from entering information about an entity if that type of entity is not represented in the underlying ontology, but this can be perceived by users as unfriendly. Second, applications can let users enter such information, but this allows the ontology to be modified in ways that do not follow best practices.

In this report, we specifically examine the interplay between OWL 2 DL [1] and the International Technology Alliance's Controlled English, ITA-CE [2]-[6] to determine the feasibility of using ITA-CE as a medium through which ontologies can be correctly modified by non-technical users.

There are two main conclusions we have drawn from this investigation. First, a conversational interface application can assist users in ontology editing tasks. We developed a prototype software application that can be used either as a command-line application or as part of the conversational panel found in IBM's ITA-CE processing environment called 'CE Store'. Our application allows people to converse with a computer in everyday English so that the user's intentions regarding ontology can be rendered into a CNL equivalent. The CNL command can then be passed on to additional machine agents, which modify the ontology and store the result in an OWL file. The application can easily be extended to ingest other data formats such as relational databases and ontology formats other than OWL. At the same time, we discovered that such software does not strictly require the use of CE statements in general or of ITA-CE statements in particular. However, within the current implementation of the conversational interface application, ITA-CE is presented to human users as unambiguous confirmation prompts to ensure that the user's natural language commands were interpreted correctly.

The second conclusion was that ITA-CE might serve as a convenient communication medium for analysts and developers handling information sources in a variety of formats, and by enabling machine agents to exploit additional information sources when attempting to interpret requests made by users.

The rest of the document is structured as follows. Section II discusses existing approaches to utilizing CNL-based approaches to modifying ontologies. Section III provides an

overview of ITA-CE. In Section IV, we provide a detailed discussion of our implementation to support conversational OWL ontology editing. In Section V, we introduce a few illustrative scenarios to show the applicability of our work. We conclude in Section VI by sketching future directions of our work.

II. Controlled Natural Language Ontology Editors

The formal underpinnings of semantic technologies are substantial obstacles for a casual end-user. This usability problem has been widely noted already within the Semantic Web. For example, Rector [7] documented numerous errors commonly made by non-expert ontology users. These include (1) the failure to make all information explicit, (2) ignorance of the effects of range and domain restrictions, (3) mixing up defined and primitive classes, (4) misunderstanding common logical constructs ('and', 'or', 'some not', 'not some'), (5) presuming that classes are disjoint by default, and (6) being insensitive to open world reasoning. Thus, non-expert ontology users face an immense hurdle developing and utilizing ontology-based information sources.

In response to the usability problem, previous research [8]-[14] has sought to hide the formal syntax and semantics of ontologies behind CNL constructs. Several projects in particular have sought to exploit existing CNLs, or develop new CNLs, in order to simplify the tasks of creating, managing, and navigating ontologies.[1] Several software applications now allow users to edit ontologies by writing English sentences that are restricted in admissible vocabulary and grammatical constructions, yet relatively easily comprehended. This sets them apart from traditional tree-structured and graph-structured ontology editors such as Protégé and TopBraid Composer™.

To a great extent, these CNL ontology editors have helped to bridge the gap between casual users and ontological formalisms. However, their success is limited by at least one of four recalcitrant problems. First, most of the editors are not fully compliant with OWL 2 DL, the most widely used member of the OWL family of languages. ACE View does not currently support sentences that express data properties and their corresponding datatypes (e.g., having a date-time value associated with an event). GINO Editor and CLOnE Editor restrict users to only very basic OWL constructs (e.g., there appears to be no support for inserting class axioms or cardinality restrictions). The ROO Editor employs a more expressive language than ACE View, GINO Editor, and CLOnE Editor, but it shies away from OWL 2 DL expressivity for simplicity [12]. Only the Fluent Editor™ 2014 connects users with OWL2 DL's full expressive potential.

Second, some of the editors employ insufficient resources for explaining and correcting user input error. CLOnE Editor users, for example, complained of receiving little guidance for inputting CNL expressions and no feedback to explain syntactical errors [11]. Similarly, users of ROO Editor complained of receiving no feedback on semantic errors [13].

Third, the documented experiments conducted with CNL ontology editors suggest that all but the simplest editing tasks lie beyond the ordinary capabilities of the CNL ontology editor user. Of the editors surveyed above, GINO, CLOnE, and ROO ran experiments with non-expert ontology users. In each case, however, users were successful only in performing basic tasks (e.g., most users could create a class, property, or instance), and for the most part unsuccessful in executing any more sophisticated task (e.g., correctly adding an axiom). If the usability problem is to be solved altogether, then end-users will need an interface that relieves much of the burden of expressing the wide array of OWL constructs.

Fourth, all of the editors require end-users to master the stringent lexical and syntactic rules governing the implemented CNL. For instance, users must be sensitive to quantifiers ('every', 'some'), disallowed terms ('or', 'not'), and peculiar lexical conventions (e.g., in ACE, dashes between multiple elements of a term, e.g., 'stretch-of-river'). Although the GINO and Fluent editors assist users in entering CNL through predictive mechanisms, and although CLOnE and ROO employ less stringent lexical and syntactic restrictions, the fact remains that users are expected to write impeccable CNL sentences. Thus, even if CNL-based editors are more human friendly than traditional tree-structured or graph-structured ontology editors, they nevertheless run the risk of alienating the non-expert user, as put forward by Smart [15].

Related projects include formulations of alternative OWL syntaxes, aimed at simplifying OWL for non-expert users, and resulting in an ontology language that resembles, in many of its properties, a CNL. In particular, we note Manchester OWL Syntax [16], which is recognized by W3C, and Sydney OWL Syntax [17].

The software discussed in this paper aims to overcome the usability problem in a different way. If it were fully developed, it would minimize people's direct interaction with CNLs and eliminate altogether the requirement for them to write syntactically correct sentences of a CNL. Instead, a non-technical user could engage in ontology creation, editing, and management entirely by means of a natural language human-machine conversation. At most a user would need to read a sentence of CNL in order to confirm that the conversational agent has correctly interpreted the natural language input. Behind the scenes, the conversational agent would translate the user's

[1] There has been little comparative study of the vast array of CNLs actively in use. Kuhn (2014) surveys 100 recent CNLs, and Schwitter (et al., 2008) offers a detailed comparison of three of the more prominent CNLs in use today: Attempto Controlled English (ACE), Ordnance Survey Rabbit, and Sydney OWL Syntax (SOS).

commands into a CNL, which would then be passed along to other agents that have access to particular knowledge representation documents, e.g., an OWL/XML file. In our current preliminary implementation of this idea, all of the interaction can take place within the CNL processing environment designed by the International Technology Alliance (ITA) and within the context of previous ITA research on CNL-based tools and their military applications.

III. Overview of ITA Controlled English

In 2010, the International Technology Alliance (ITA) began developing a CNL known as ITA Controlled English (ITA-CE) for the purpose of supporting tasks within the Data-to-Decisions (D2D) framework, "specifically to assist coalition decision makers in distributed information environments through automated or semi-automated fusion processes" [4]. Previous ITA-CE research has addressed the problem of the miscommunication between US and UK military personnel rooted in lexical and cultural discrepancies and the problem of enhancing shared understanding and communication for military decision-making, especially through the exploitation of sensor resources. An example of this is MOIRA (Mobile Intelligence Reporting App), which aims to expedite data requisition within ISR missions.

At base, the syntax and semantics of ITA-CE are adopted from Sowa's Common Logic Controlled English [18]-[19], which in turn aligns itself closely to first-order logic. As with most CNLs, the resultant expressions of the language are readily comprehensible to ordinary English speakers. For example:

```
there is a person named Steve.
the person Steve is married to the person Jane.
the person Steve has the person Jane as spouse.
```

By writing ITA-CE sentences, users can gradually construct a model, in which all pertinent entities within that model – including types, properties, relations, and individuals – are specified. Thus, to construct a model, a user would write sentences defining the objects, properties, and relationships within that domain.

New terms are introduced to the model by means of "conceptualise" statements. For example:

```
conceptualise a ~ Chihuahua ~ C that is a dog.
conceptualise the Chihuahua C ~ barks at ~ the
person P1.
conceptualise the Chihuahua C has the person P2 as
~ owner ~.
```

The term being added or modified is set off by tildes (~), and if the new term is a noun, it is followed by an uppercase variable name.

Additionally, users can write rules of inference to the model in the form of "if-then" statements. For example:

```
if
  ( the person P1 is married to the person P2 )
then
  ( the person P2 is married to the person P1 ) .
```

This rule states that the "is married to" relation is symmetric. If a user includes this rule in a model and also includes the sentence "the person Steve is married to the person Jane," then software can easily and correctly infer "the person Jane is married to the person Steve."

Models can be defined and extended using the ITA-CE processing environment, called 'CE Store' [5].[2] ITA-CE's CE Store software allows users to define and execute custom "CE agents," including conversational agents with which users can converse in ordinary English. Below, we present a pair of CE agents, one of which assists the user in formalizing natural language expressions and the other of which retrieves and emends information within OWL files.

IV. Conversational Interface Application

The conversational interface application that we coded cooperates with the CE Store software, which takes the form of a web application that accepts ITA-CE sentences as input and responds by remembering the ITA-CE sentence and possibly triggering other behavior based on the content of the ITA-CE sentence. Our software is an add-on to the CE Store in the sense that it is a WAR file that can be placed in the same folder as the CE Store's WAR files and is configured by storing a few CE statements in the CE Store.

Given the suitability of the CE Store for incorporating ontology information, one possible strategy for editing ontologies by means of ITA-CE is first to translate the English into ITA-CE and then ITA-CE into OWL. In practice, however, such a strategy faces some obstacles.

To see why, first observe that in order for the software to guide the user in editing an ontology, it must somehow access all the relevant information about the ontology. Because the CE Store software stores its data in the form of ITA-CE statements, the ontology would first need to be loaded into CE Store. This would incur some data redundancy and would require the addition of some rules to draw inferences not explicitly asserted by the ontology. Neither of these is a sizable obstacle. However, if we were to adopt this strategy, and then a user made changes to the ontology, then the previous statements might become false. For example, an ontology might define a hierarchy in which Z is a direct subtype of X. If a user chooses to insert a new type Y so that X is a parent of Y and Y is a parent of Z, the previous statement will need to be deleted. Deletion can pose a problem for the CE Store because new statements are sometimes automatically inferred according to the rules already present in the store.

Although limited deletion of sentences is no obstacle, some kinds of changes to an ontology (that ought to be allowed) could potentially require identifying and deleting sentences en masse, at least if there are already statements about instances of the classes in the ontology. For example, we

[2] An alpha version is publically available for download at http://ibm.co/RDIa53.

might have a relation "receives information from" between artifacts, with some instances like "the machine KRF343 receives information from the machine EELR." and "the machine KRF343 receives information from the machine KRF343." Now suppose that someone alters the ontology by marking the "receives information from" relation as reflexive, which triggers (for every artifact X) the addition of a statement, "X receives information from X." Then, if the reflexivity of this relation is removed, the proper behavior would not be to have all such statements removed, but to remove only those that statements whose existence were generated by the rule. In that case, we might need to keep "the machine KRF343 receives information from the machine KRF343," because, say, it is a special radio that broadcasts messages and also records what it broadcasts in addition to what it receives from other broadcasters. The CE Store can keep track of which rules its sentences are derived from (as "rationale graphs"); even then, however, there remains a problem of managing which information *should* be deleted.

A second related worry is that the ontology editing software does not have exclusive control over the CE Store, nor can it verify ahead of time whether all the rules CE Store will be compatible with changes that a user makes. If a user accidentally adds a statement to the ontology which renders it inconsistent, it is possible that rules that trigger on the statements in the ontology will generate conflicting assertions, which can trigger further undesired effects that might be hard to predict and difficult to undo.

A third deficiency of the CE Store for maintaining the ontology information that users are editing is that it does not provide a means for keeping different users' ontology changes separate. In its current state, the CE Store does not allow any name spaces, which could differentiate between conflicting definitions of an entity.

Consequently, we found it advisable to adopt two design policies. First, it should not be required for the CE Store to ingest the ontology information, which is already in the OWL file. Second, we should not require the content of the OWL file match what ontology information (if any) kept in the CE Store. At the same time, we determined that we could still use the CE Store by inserting ITA-CE statements about the user's intention to make an ontology change or about what information about the ontology the user would like to obtain.

The structure of the resultant application is such that a user types ordinary English sentences into either a terminal window or the conversational interface of the CE Store, which triggers a response from a specially designed Java class, OntologyAgent. This Java class parses English input in attempt to infer the user's intentions regarding the loaded ontology. In the special case where the user's input is in controlled English, the OntologyAgent can act on it without needing to ask the user for further clarification, but when the user states a command in ordinary English, the OntologyAgent will do its best to interpret the input, sometimes responding with advice or questions. The output to the user is often in ordinary English, but when the OntologyAgent is prompting the user for confirmation, it provides a controlled English statement to coach the user so that on future occasions, the user can use controlled English for quicker unambiguous communication.

When the OntologyAgent needs information about the current state of the ontology, it sends out requests for information by placing appropriately structured ITA-CE sentences in the CE Store, which in turn trigger a response by another specially designed Java class, OwlAgent, which loads OWL ontology files and operates automated reasoners to answer questions about the inferred ontology, sending answers back to the OntologyAgent through ITA-CE statements placed in the CE Store. Information is shuttled back and forth between the OwlAgent and OntologyAgent until the OntologyAgent feels confident about the meaning of the user's initial request. Once the user has confirmed the accuracy of the OntologyAgent's interpretation, it passes an ITA-CE string to an OwlAgent *via* the CE Store conversational interface, the OwlAgent analyzes the requested modification, and, so long as it protects the integrity of the OWL file, updates the ontology accordingly. Given this structure, the program can (and does) log changes to all ontologies, keeps different users' ontologies separate, and saves changes to the ontologies incrementally to allow users to undo changes.

In order to enhance its interpretive capabilities, the OntologyAgent attempts to leverage information already present within the OWL file. It does so by posing queries to the OwlAgent (again, *via* the CE Store), such as, 'Does the class C exist in the ontology?' and 'Does the user's request violate any domain or range restrictions on object properties?' The OwlAgent, who manages changes to OWL files, answers the OntologyAgent's queries, for example, 'There is no such class in the OWL file' or 'That requested change would violate a domain restriction'. As a result, the OntologyAgent can make suggestions based on the information already contained within the OWL file and thus provide guidance to users wishing to modify an ontology. Currently, the software allows users to add classes anywhere in the class hierarchy and to add any desired existential restrictions. With the basic framework having been coded, it is straightforward to expand the software to allow other types of ontology changes.

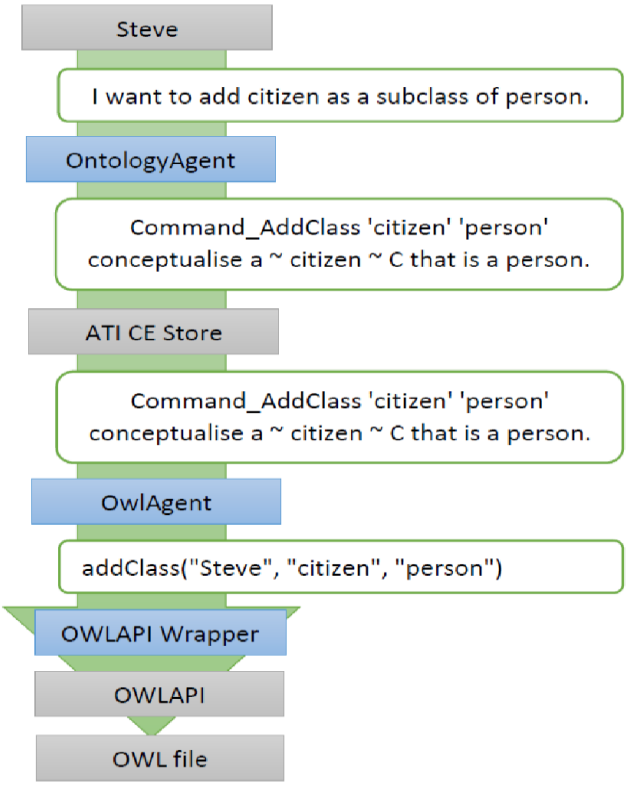
Consider an example. The user, who is logged in as 'Steve', types "I want to add Citizen as a subclass of Person." into the conversational interface. The OntologyAgent is part of

the CUBRC software and conducts some basic natural language processing to identify that the string matches one of the allowed forms for stating that that one class X should be made a subclass of another class Y. The user could have just typed "citizen is person" and gotten the same result. The interface is designed to be very forgiving about what it accepts. After getting confirmation from the user, the

OntologyAgent passes the user's command into the CE Store using a controlled English statement

```
there is a request card named 'msg_1' that is from
the service 'OntologyAgent' and is to the
individual 'OwlAgent' and has and has 'citizen' as
argument one and has 'person' as argument two and
has 'Steve' as human and has 'Command_AddClass' as
command and has 'conceptualise a ~ citizen ~ C
that is a person.' as content.
```

This string, when it enters the CE Store is recognized as a



legitimate CE statement. Because it is recognized as a card to the individual 'OwlAgent', the OwlAgent coded as part of the CUBRC software is passed the string through a Java method. The OwlAgent then parses the string to find that it should check whether 'citizen' can be added as a subclass of 'person'. It does so by making calls to a layer of the CUBRC software that keeps track of which ontologies are being edited by which humans and limits the kind of ontology changes permitted. This OWLAPI wrapper then makes calls to a publicly available Java API called OWLAPI, which is more complex and permits a much wider range of operations on ontologies, especially those stored as OWL files.

Note, however, that the OntologyAgent sends redundant information to the OwlAgent. In the content string it sends a statement that obeys the syntax and semantics of ITA-CE:

```
conceptualise a ~ citizen ~ C that is a person.
```

In the command, argument one, and argument two strings, it conveys in effect the same information:

```
Command_AddClass 'Citizen' 'Person'
```

The reason for passing two strings to the OwlAgent that express the same command is that (1) it is easy for the OntologyAgent to generate a tiny amount of additional text to express the same command in different formats, and (2) any agents that are coded to receive commands from an OntologyAgent can be coded to respond to whichever format is easiest to parse. In our conversational interface software, the Java methods called by the OwlAgent correspond exactly with the command strings that the OntologyAgent sends to the OwlAgent. The correspondence exists because there are only a small number of types of ontology alterations that it is reasonable to allow. Because OntologyAgents and OwlAgents understand that adding class X as a subclass of class Y is something a user probably wants to do, an OntologyAgent could encode that command in ITA-CE and have an OwlAgent decode it, but it is a bit simpler for the CUBRC software (and no harder for the CE Store) to have the OwlAgent ignore the ITA-CE altogether and check the command string and its arguments directly.

This raises interesting questions about what role, if any, controlled English can or should play in a conversational interface for editing ontologies. In the current version of the software, ITA-CE strings are used to provide a logically unambiguous yet human-readable confirmation prompt, but are not needed for processing the ordinary English input. Three questions in particular arise. Should some CNL be used even for the confirmation prompt? Should some CNL be used in this software for transmitting messages between the OntologyAgent and OwlAgent? And is ITA-CE the best form of CE to be used if we do use CE for user confirmation?

Concerning the first, the simplicity and lack of ambiguity provides a reason to use it for a command confirmation. But this alone is not a strong reason because it is often possible to make CNL statements even clearer in meaning by dropping some of the formality. For example, instead of the formula in ITA-CE:

```
conceptualise a ~ citizen ~ C that is a person.
```

the computer could offer

```
conceptualise a citizen that is a person.
```

without any loss of content. The tildes and the variables just indicate which concept is being defined, which is often clear from context or can be emphasized with boldface if desired. A stronger reason for using CNL is that it helps to coach the user so that the next time the user wants to enter the same kind of request, the user can use the sort of phrasing that appeared in previous prompts.

Concerning the second question of whether CNL should be used for the transmission of messages, there are some good arguments for its use. In our particular software, we controlled both ends: the natural language processing and the ontology processing. But if we think more broadly about how the conversational interface could be used, we find that an OntologyAgent could potentially get clues about how best to advise the human user from a wide range of

additional sources beyond what is kept in the edited ontology. Given the existing software framework, one could code agents that respond to an OntologyAgent's information requests by looking at a relational database, dictionaries, or even special purpose ontology advisors. Or one could code agents that edit ontologies stored in Open Biological and Biomedical Ontology (OBO) files rather than OWL files. The separation of the functions requires some sort of common communication format. Because CNL is relatively easy to parse and easy to extend in order to add additional functionality, it could serve well in this role. A further benefit of allowing CNL to play this role would be that one could easily set up logging and auditing software to work with the CE Store in order to keep track of which ontology changes were being made and by whom. Such information could be handy for later analysis to ascertain common ontology editing patterns, which could then be used to automate revisions to ontologies.

Concerning the third question, it is important to keep in mind that ITA-CE is severely restricted in its core lexicon and syntax. For example, the words 'every', 'some', 'or', and 'not' are not considered grammatical. While these same ideas can be expressed indirectly, the resultant statements and rules necessary for that expression tend to be too awkward for casual users (e.g., 'or' is defined in terms of complex statements involving numerous cases of 'and' and 'it is false that'). Thus, extending the lexicon and syntax of ITA-CE would greatly facilitate user understanding if these terms could appear in the confirmation prompts and in any communication to reduce complexity. In this way, ITA-CE could emulate some of the existing CNL-based ontology editors mentioned above, insofar as these editors utilize CNL constructs that more closely resemble OWL constructs.

V. USE CASES: MODIFYING THE SENSOR ONTOLOGY

A. *Adding New Classes*

It is easy to add a new class to an ontology using the conversational interface. The easiest way to do this is for the user to use the "is a" form that is standard in Basic Formal Ontology. The user types

```
An animal is an object.
```

At this point, the OntologyAgent sends a test message to the OwlAgent to see if the class "animal" can be assigned as a direct subclass of "object". If it can, the OwlAgent lets the OntologyAgent know this, and the OntologyAgent asks the user for confirmation. If the user answers, "yes", the OntologyAgent tells the OwlAgent to add the appropriate axiom and the OwlAgent does so, logging the change in case the user later wants to undo it. If the user answers "no" or doesn't respond or issues some alternative statement, The OntologyAgent will forget about the attempt to add "animal".

The user has other ways to communicate the desire to add a class. The following statements are equivalent to "An animal is an object."

```
animal is object
Animals are objects.
I would like to make animal a subclass of object.
Make animal a kind of object.
```

The software also allows users to insert a new class in between two existing classes, one of which is a direct subclass of the other.

```
Insert the class organism between object and
animal.
```

B. *Adding Existential Restrictions*

The other main capability of the software allows the user to add a new existential restriction on a class. The user can type something like

```
Every sensor observation is about some detected
material entity.
```
or
```
Sensor observations are about detected material
entities.
```

When receiving such a statement, the OntologyAgent sends a test message to the OwlAgent to see if the class "sensor observation" can have an existential restriction with the relation "is about" and the class "detected material entity". If it can, the OwlAgent lets the OntologyAgent know this, and then the OntologyAgent asks the user for confirmation. If the user answers, "yes", the OntologyAgent tells the OwlAgent to add the appropriate axiom and the OwlAgent does so, logging the change in case the user later wants to undo it. If the user answers "no" or doesn't respond or issues some alternative statement, the OntologyAgent will forget about the attempt to add the existential restriction.

C. *Adding Existential Restrictions in the Face of Obstacles*

In order to demonstrate some of the more sophisticated capabilities of our application, we created a use case with the following vignette:

> Your current version of the CUBRC Sensor Ontology includes a class called 'Detected Material Entity' but does not include any additional information about it.
>
> You want to introduce a class called 'Sensor Observation' and make the ontology understand that a sensor observation is the kind of thing that is about detected material entities. The ontology already has an 'is about' relationship but does not know the term 'Sensor Observation'.
>
> Your goals, then, are (1) to add a new class for sensor observations, (2), correctly situate that class within the existent hierarchy of classes, and (3) describe its relationship to detected material entities.

This task is more complicated than simply adding an existential restriction, but nonetheless quickly doable for the user who interacts with the OntologyAgent. The process

begins when the user enters a natural language expression that informally captures his request to add an existential restriction. The user could type anything like the following:

```
I want to make every sensor observation be about
some detected material entity.
I want to make sensor observation be about
detected material entities.
I want sensor observations to be about detected
material entities.
Sensor observations should be about detected
material entities.
Sensor observations are about detected material
entities.
a sensor observation is about a detected material
entity.
```

With such a request, the OntologyAgent queries for the class 'sensor observation'. Discovering – as the vignette stipulates – that there is no class by that name in the ontology, the OntologyAgent begins to search for any clues that will help interpret the user's intention. The OntologyAgent first checks the various parts of the user's phrase. In this case, it sees that 'observation' is a word in 'sensor observation' and checks whether there is a class 'observation' that could be a superclass for 'sensor observation'. When the OwlAgent tells the OntologyAgent that 'observation' is also not in the ontology, the OntologyAgent looks for restrictions upon relations (e.g., domain or range restrictions) in this case, whether the 'is about' relation already has any existential restrictions. (Although we have not coded further capabilities into the OntologyAgent yet, it would be easy to extend the CUBRC software to also look for synonyms of class names and annotations like developer comments, definitions, and class labels.) When searching the 'is about' relation for restrictions, the OwlAgent reports to the OntologyAgent the following clue: there is an axiom about the class *Information Content Entity* to the effect that *Information Content Entity* is equivalent to 'is about some *entity*' (a generic class comprising all other classes). Since the user is trying to express that sensor observations are about detected material entities, and since being an *Information Content Entity* is equivalent to being about some *Entity*, the OntologyAgent formulates a guess that (a) we need to insert a new class called *Sensor Observation*, and (b) the class *Sensor Observation* needs to be a subclass of *Information Content Entity*, though not necessarily a direct subclass. So, the OntologyAgent gets the direct subclasses of *Information Content Entity* from the OwlAgent and presents to the user the following text.

```
OntologyAgent: I understand you as wanting to add
an existential restriction. You are trying to
express that every sensor observation is about
some detected material entity. I think you want me
to issue the following command:
conceptualise a ~ sensor observation ~ C that is
about the detected material entity R.
However, the ontology is not aware of what 'sensor
observation' means. Because of the 'is about'
relation, a sensor observation should probably be
assigned to be one of the following:
1. A descriptive information content entity
2. A designative information content entity
3. A directive information content entity
4. An information content entity
If is one of these, just let me know the number,
or you can write something of the form, "a sensor
observation is a ___."
```

As it turns out, the user Steve decides to situate *Sensor Observation* under the parent class of *Descriptive ICE*, so he types

```
1
```

The OntologyAgent responds by adding 'sensor observation' as a direct subclass of the 'descriptive information content entity' class. Once the OwlAgent reports that that change has been made, the OntologyAgent knows that the obstacles have been cleared to allow the existential restriction to be added. So the OntologyAgent immediately sends the command to the OwlAgent without needing further confirmation. When the OwlAgent reports that this further command has succeeded, the OntologyAgent reports to the user the following:

```
OntologyAgent: I successfully added 'sensor
observation' as a direct subclass of 'descriptive
information content entity'. I successfully added
the existential restriction you requested.
```

In summary, Steve submitted a single sentence to convey his desired ontology change. After that, Steve only needed to select a multiple choice answer in order to clarify where the previously unknown 'sensor observation' class should be placed in the hierarchy, and all the necessary changes were made without any further effort.

In practice, all ontology changes would be audited before being integrated into widely shared OWL files.

VI. FUTURE WORK

Our research on this topic was merely exploratory because of resource limitation and a number of future efforts are warranted. Greater editing functionality should be added first because it provides clear value and is not an ambitious task. The harder task will be to improve the software's ability to help the user with the most appropriate questions and advice. Still, even these improvements can be incremental and piece-meal, as the natural language is meant only to help guide and coach when the user does not know the appropriate controlled English to use.

Earlier, we showed that although our application extends the capability of the CE Store software, it does so without necessarily using ITA-CE to represent ontological claims. The role played by ITA-CE might easily be played by a more expressive CNL or a standard data format like XML or JSON. There is potential for ITA-CE to serve as a common format for capturing the semantic intention of natural language user inputs. Its value in this regard will largely depend on how much other infrastructure uses ITA-CE and how well the information being passed back and forth can be leveraged to automate some ontology development.

At this stage, our application is limited to assistance with OWL-based ontology editing. It is our hope that this project will eventually be extended further, so that ITA-CE

sentences can be exploited to mediate user interaction with other ontology languages (e.g., OBO) and other formats (SQL, relational databases). Thus, in dividing labor between the OntologyAgent and OwlAgent, we have left open the possibility of mapping ITA-CE to these formats and of writing additional back-end agents, whose jobs, like the OwlAgent's, would be to exchange information with the OntologyAgent and edit the appropriate document on behalf of the user. We also leave open the possibility of alternative OntologyAgents, who pass messages in ITA-CE only.

To illustrate this proposal, consider the following two scenarios. In *Scenario #1*, a user interacts with a different OntologyAgent that outputs ITA-CE strings only, not the command strings described earlier, and sends the messages to an OwlAgent. The problem is that the OwlAgent does not understand ITA-CE strings; it ingests command strings only. In this scenario, it would be desirable to introduce a further intermediary agent to translate ITA-CE strings into command strings amenable to the OwlAgent's work. By introducing this further layer, we allow the OwlAgent to continue interpreting messages in terms of command strings. At the same time, this makes room for ITA-CE to act as middleware for various other agents.

In *Scenario #2*, a user interacts with the OntologyAgent, who passes messages in turn to an OwlAgent that understands ITA-CE strings only, not command strings. This new OwlAgent would have access to a relational database and would query that database to provide the OntologyAgent with information about the user's intended request. In this scenario, it becomes indispensable that the OntologyAgent transmit not just command strings, but ITA-CE strings as well. Thus, if ITA-CE is to be the common format for various agents and information sources, we ought to ensure that OntologyAgents are conversant in ITA-CE.

These scenarios highlight another potential benefit of employing ITA-CE as middleware. If ITA-CE is used as a common format, then additional agents could be programmed to provide OntologyAgents with answers to his queries which are based on the access they enjoy to various information sources. In this case, ITA-CE appears to be promising as a common format for the exchange of information among agents.

If ITA-CE were to be harnessed in these ways, then in will be necessary to augment the present ITA-CE core lexicon and syntax. At present, ITA-CE does not allow the terms 'every', 'some', 'or', and 'not', which inhibits users from facile comprehension. Thus, an extension of this project would be to pursue methods laid out by Mott and Hendler [2], in which new layers of generic syntax are added to the core ITA-CE syntax. Mott and Hendler illustrate this with the adverb 'only'; to add 'only' to the syntax of ITA-CE, they invented a language, which defines 'only' in terms of the unaugmented core syntax. Further transformations could add common quantifiers and connectives such as 'every', 'some', 'or', and 'not', thus enhancing users' interaction with ITA-CE expressions by rendering them more natural.

Acknowledgments

Work on ITA-CE as a middleware for ontology editing was supported by the US Army Research Laboratory. We also thank David Mott for his consultations on ITA-CE.

References

[1] Bock, C., Fokoure, A., Haase, P., Hoekstra, R., Horrocks, I., Ruttenberg, A., Sattler, U., Smith, M. 2012. "OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax," 2nd

[2] Mott, D. Hendler, J. 2009. "Layered Controlled Natural Languages," *Proceedings of the Third Annual Conference of the International Technology Alliance*.

[3] Braines, D., Mott, D., Laws, S., de Mel, G., Pham, T. 2013. "Controlled English to Facilitate Human/Machine Analytical Processing," in Proceedings of SPIE.

[4] Braines, D., Preece, A., de Mel, G., Pham T. 2014. "Enabling CoIST Users: D2D at the Network Edge," in 2014 Proceedings of the 17th International Conference on Information Fusion (FUSION).

[5] Poteet, S., Xue, P., Kao, A., Mott, D., Braines, D., Giammanco, C. 2013. "Controlled English for Effective Communication during Coalition Operations," Proceedings of ICCRTS.

[6] Preece, A., Pizzocaro, D., Braines, D., Mott, D., de Mel, G., Pham, T. 2012. "Integrating Hard and Soft Information Sources for D2D using Controlled Natural Languages," *Proceedings of the 15th International Conference on Information Fusion*.

[7] Rector, A., Drummond, N., Horridge, M., Rogers, J., Knublauch, H., Stevens, R., Wang H., Wroe, C. 2004. "OWL Pizzas: Practical Experience of Teaching OWL-DL: Common Errors & Common Patterns," *European Conference on Knowledge Acquisition* (EKAW-2004), Whittlebury, UK.

[8] Bernstein, A., Kaufmann E. 2006. "GINO – A Guided Input Natural Language Ontology Editor," in Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L. (eds.), ISWC 2006. *Lecture Notes on Computer Science*, Vol 4273.

[9] Bernstein, A., Kaufmann, E., Kiefer, C. 2009. "Querying the Semantic Web with Ginseng – A Guided Input Natural Language Search Engine," *Searching Answers: Festschrift in Honour of Michael Hess on the Occasion of His 60th Birthday*, Clematide, S., Klenner, M., Volk, M. (eds.). Munster: MV-Wissenschaft.

[10] Kaufmann, E., Bernstein, A. 2007. "How Useful Are Natural Language Interfaces to the Semantic Web for Casual End-Users?" *Lecture Notes in Computer Science*, Vol. 4825, 281-294.

[11] Funk, A., Tablan, V., Bontcheva, K., Cunningham, H., Davis, B., Handschuh, S. 2007. "CLOnE: Controlled Language for Ontology Editing," *Lecture Notes in Computer Science: The Semantic Web*

[12] Hart, G., Johnson, M., Dolbear, C. 2008. "Rabbit: Developing a Controlled Natural Language for Authoring Ontologies," *The Semantic Web: Research and Applications, Lecture Notes in Computer Science*, Vol. 5021, 348-360.

[13] Kaljurand, K. 2008. "ACE View – An Ontology and Rule Editor Based on Controlled English," *Proceedings of the Poster and Demonstration Session at the 7th International Semantic Web Conference* (ISWC 2008), CUER Workshop Proceedings.

[14] Wroblewska, A., Kaplanski, P., Zarzycki, P., Lugowska, I. 2013. "Semantic Rules Representation in Controlled Natural Language in FluentEditor," *The 6th Annual International Conference on Human System Interaction* (HSI).

[15] Smart, P.R. 2008. "Controlled Natural Languages and the Semantic Web. Technical Report ITA/P12/SemWebCNL, School of Electronics and Computer Science, University of Southampton.

[16] Horridge, M., & Patel-Schneider, P.F. 2012. "OWL 2 Web Ontology Language Manchester Syntax," 2nd edition.

[17] Cregan, A., Schwitter, R., Meyer T. 2007. "Sydney OWL Syntax – toward a Controlled Natural Language Syntax for OWL 1.1," presented at *OWLED 2007, OWL: Experiences and Directions, Third International Workshop*, Innsbruck, Austria, 6-7th, June 2007.

[18] Sowa, J.F. 2007. "Common Logic Controlled English," http://www.jfsowa.com/clce/clce07.htm.

[19] Mott, D. 2010. "Summary of ITA Controlled English," https://www.usukita.org/papers/5658/details.html.

Toward Representing and Recognizing Cyber-Physical Elements in Competition Using Event Semantics

Alonza Mumford, Duminda Wijesekera, Paulo Costa
George Mason University
amumford@gmu.edu, dwijesek@gmu.edu, pcosta@gmu.edu

***Abstract*—The Federal Aviation Administration (FAA) is observing an increasing number of incidents involving recreational drones, and imagining a future where every drone will be equipped with an *Automatic Dependent Surveillance-Broadcast* (ADS-B) transponder that communicates and cooperates with the FAA's Next Generation (NextGen) Aviation Cyber-Physical System in order to help mitigate aerial collision risk [1]. This exemplar application involves human or autonomous agents interacting within some sort of cyber-physical system where competition or cooperation between cyber-physical elements exist. We anticipate that the use of higher-level abstractions will be required for modeling human or autonomous agent's interactions within these type of systems in order to make sense of the observations derived from sensor-data. In this paper, we articulate an approach that uses event semantics to represent the temporal, spatial, factor, and outcome features of activities generated by competing or cooperating agents functioning within a cyber-physical environment. We use those semantics, along with observations of activity, to model higher-level activity abstractions and to help perform strategy recognition from a concrete, competition-oriented scenario reflected in a real-world, game data set comprised of more than a half million events involving nearly 8500 unique agents. The strength of the approach is grounded in a specification of event semantics for our concrete multi-agent, competitive game ontology using Resource Description Framework Schema (RDFS) and Ontology Web Language (OWL). By leveraging these Semantic Web languages, we anticipate that the use of event semantics to describe cooperative or competitive agent interactions within cyber-physical systems will become more predominant in the future.**

***Index Terms*—Agent-Based Model, Human Agent, Autonomous Agent, Unmanned Aerial Vehicle, Gridiron Football, Semantic Web**

I. Introduction

Cyber-physical systems (CPS) are at the outset of completely changing how society interacts with the physical world around it. These systems measure different features across the physical environment (e.g., the location of an agent) and enable computational models that interact with a cyber-core (i.e., the computing and communications backbone of the CPS) and with their corresponding physical environment to provide some desired benefit or utility. In most cases, sensors provide the cyber-core with the primary mechanism for recognizing events or changes in the physical environment. The actions and interactions between human or physical autonomous agents and the cyber-core are captured through sensors. Consider some real-world applications where the activities of human and physical autonomous agents are identified by sensors coupled to a cyber-physical system:

- In the National Football League (NFL), each football player and stadium is equipped with RFID sensors and receivers permitting the league to track fine-grained location data for each play. In this case, the Internet-of-Things (IoT) and CPS has been incorporated into operations and management of professional sports venues [2].
- In the recreational Unmanned Aerial Vehicle (UAV) market, some manufacturers have equipped their drones with ADS-B sensors, which is a type of sensor for aerial cooperative collision detection and avoidance [3]. ADS-B is an element of the Federal Aviation Administration (FAA)'s Next Generation Air Transportation System (NextGen), which has been described as a airborne network instantiation of a cyber-physical system [1].

These examples also illustrate a specific characteristic present in some cyber-physical systems where competition or cooperation exist between human or physical autonomous agents. Subsequently, this paper reflects an interest in these type of cyber-physical systems. This research effort is narrowly focused on identifying higher-level abstractions such as strategies used by agents from observations derived from sensors in the CPS. Further, we focus on an approach for the representation and modeling of competitive actions and interactions of agents in cyber-physical systems.

II. Methodology

The high-level methodology for this research activity has been decomposed into five components. First, an exemplar for our experiment is identified. Gridiron Canadian and American Football is distinguished as an elaborate, competitive game activity that involves multiple agents, which are organized into two teams for the purpose of executing a series of offensive/defensive advances intended to score points and win the game. This scenario is selected to match the CPS exemplar identified in [2]. A NFL play-by-play data set that offers a likeness or model of the kind of RFID-derived data we would expect in [2] is acquired for the experiment. Second, the information within the data set is conceptualized based on the domain knowledge of Gridiron Football and modeled for

its semantic relationships using event semantic abstractions. The result is the contributed *Gridiron Football Ontology*, which is a conceptual vocabulary of American and Canadian football, in the namespace http://www.ncsu.org/kr/fball. Third, the data set is extracted from its previous form, transformed into the Resource Description Framework (RDF) metadata model (including serialization) according to the *fball* ontology and loaded into a public-accessible SPARQL Protocol and RDF Query Language (SPARQL) Endpoint and RDF Store. Fifth, we integrate or point a public-accessible SPARQL Endpoint Explorer for Expressive Question Answering to the Gridiron Football Event Endpoint in order to interrogate the nearly 40 million triples generated for indications of tactics or strategies being employed during game events. Further details are provided in each respective section of the paper.

III. Proposed Solution

A. Data

Our research effort was presented with a considerable challenge pertaining to the acquisition of rich context data that would comprise human or agent activities, and that would simulate the type of data we would expect to acquire from the scenario identified in [2].

(7:41) D.Williams left guard to PIT 13 for 6 yards (A.Branch; G.Grissom).

Fig. 1. An example illustration of play-by-play text generated by a NFL statistician, which translates to: a RUSH play event occurred at the "7 minute and 41 seconds" timestamp of the quarter; D. Williams rushed the ball in the direction of his Left Guard and progressed 6 yards up to the 13 yard line of the opposing Pittsburgh Steelers team; and where he was jointly tackled by A. Branch and G. Grissom.

Prior to recent introduction of RFID sensors into NFL game stadiums [2], the data capture of movements of players on a football game field demanded a human-in-the-loop to observe the execution of plays across the field and to sequentially report the specifics of each event as illustrated in figure 1. During games, this reporting data is manually generated by a distributed network of league statisticians, and immediately propagated from individual stadiums to hundreds of websites in a span of seconds [2]. In turn, the data set used in the experiment is provided by a sports analysis firm that uses web extraction and other techniques to generate a structured data set from disparate web post left by these league statisticians. The data set contains NFL play-by-play events from 2000 to 2013. Further, we assert that this data is characteristic of the type of complex yet coarse- or fine-grained event data that can be expected from observations of human or autonomous physical agent's activities as they engaged in competition within a cyber-physical system.

B. Domain Knowledge Acquisition

An effort was made to grasp an understanding of the physical environment in which our football player agents operate. Domain knowledge of concepts such as game field, players, game timeline, driving the ball, alternating possession of the ball, kickoffs, free kicks, scoring and penalties associated with *Gridiron Football* was acquired primarily through a literature review of popular publications such as *Football's Matchup Zone Coverages* [4], *The Art of Place-Kicking and Punting* [5], *Defending the Spread Offense* [6], *Offensive Football Strategies* [7] and *Winning Football* [8]. Familiarity of these concepts was used to formulate ontological abstractions designed to codify types of events with associative kinds of entity-attribute-models with agent, temporal, spatial, factor and product classes and relationships in the *Gridiron Football Ontology*.

C. Formal Ontology Modeling

The contributed *Gridiron Football Ontology* expresses a formal representation of knowledge within the Gridiron (or American and Canadian) Football domain via a set of concepts and inter-concept relationships. The ontology is engineered according to a methodology that partitions the ontology into two levels: theory ontology (or upper ontology) and domain ontology (or lower ontology). The theory ontology is abstract and compact. It focuses on concepts such as time, space, goals, etc. The domain ontology provides a formal description of the classes (i.e., concepts) and relationships between classes that exist in a domain. In this manner, the *Gridiron Football Ontology* is conceived as a two-level ontology with an upper level that abstracts a football game as a sequence of organized spatio-temporal events and a lower level that provides a concrete specification of the ontology components (i.e., individuals, classes, attributes, relations, etc.) associated with Gridiron Football. By concrete specification, we mean that the ontology is machine-readable and understandable. The result is that *RDF Schema (RDFS) Language* and *Web Ontology Language (OWL)*-based ontological software components were developed for all items (370+ unique key-value pairs) in the NFL Play-by-Play data set. Ontological modeling and engineering efforts were assisted using several Semantic Web tools to include CMAP Knowledge Modeling Environment [9] and Stanford Protege [10].

At the center of the upper ontology is the notion of an event, and we reuse the existing *Event Ontology* developed at Queen Mary, University of London [11]. The ontology is partitioned into a set of classes and properties identified as: *event:Event*, which is an arbitrary classification of a space/time region, which may have participating agents, passive factors, products, and a location); *event:sub_event*, which provides a mechanism to partition a complex event; *event:agent*, which relates an event to an active agent such as a person; *event:time*, which relates an event to a time object such as a duration of time. *event:place*, which relates an event to a spatial object; *event:product*, which relates an event to something produced during an event; and *event:factor*, which relates an event to something that contributes to its result such as a cause.

In figure 2, domain-specific classes and properties are created through subsumption of the *event:Event* class to model the many type of events associated with *Gridiron Football*. For example, *fball:GameEvent* class conceptualizes

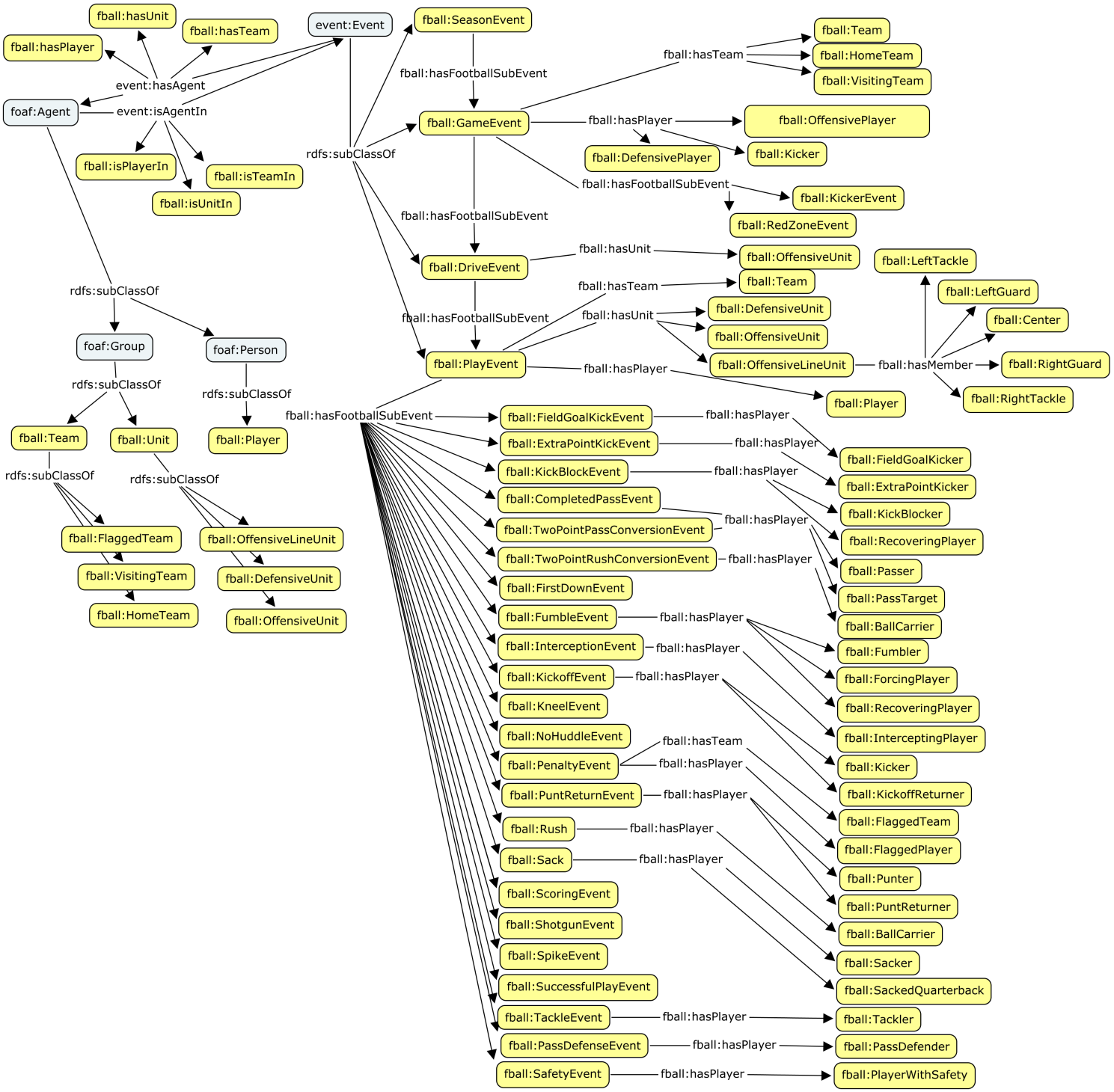


Fig. 2. A detailed specification of the *fball* ontology's Active Agent description using portions of the *Event Ontology* and *Friend Of A Friend* (FOAF) vocabularies. In this context, an active agent is a person or machine that performs in an event. Ontological components that are members of the upper ontology are illustrated in blue color whereas the members of the *Gridiron Football* domain ontology are colored in yellow.

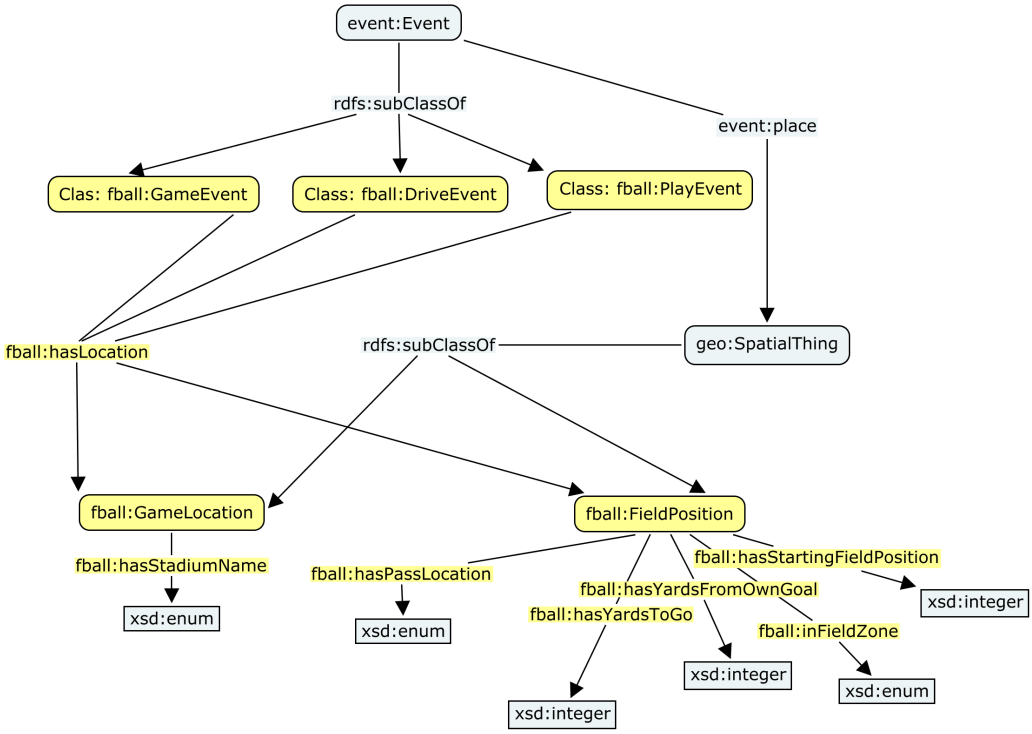


Fig. 3. A detailed specification of the *fball* ontology's active agent description using portions of the *Event Ontology* and *Basic Geo* vocabularies.

a football game whereas *fball:SeasonEvent* represents a football season. In the case of the NFL, a season event is comprised of 16 football games over a duration of 17 weeks. The *fball:hasFootballSubEvent* property is devised through subsumption of the *event:hasSubEvent* property and used to partition a Season event and a Game event into a collection of games and collection of plays respectively. The *foaf:Agent* class and its sub-classes, *foaf:Person* and *foaf:Group* are extended to model the various types of football playing positions (e.g., *fball:LeftTackle*), the membership of particular playing positions to certain football sub-groups (e.g., a Left Tackle has membership of the *fball:OffensiveLineUnit*). The *fball:Player* is established to represent the idea of a generic football player that has datatype properties (i.e., attributes) such as *fball:hasFullName*, *fball:hasPrimaryPosition*, *fball:hasHeight*, *fball:has40YardDashTime* and *fball:hasBenchPressWeight*. The *fball:hasPlayer*, *fball:hasUnit* and *fball:hasTeam* properties are created by means of subsumption of the *event:hasAgent* property to relate football persons and groups to kinds of football events. For example, the *fball:hasPlayer* property is used to establish a relation between the *fball:FumbleEvent* and one of three player roles expected or required during a fumble type of event: *fball:Fumbler*, *fball:ForcingPlayer* and *fball:RecoveringPlayer*.

Domain-specific classes and properties are devised by way of subsumption of the *event:Product* class to model the many types of outcomes that result from different types of football events associated with *Gridiron Football*. For example, *fball:KickoffOutcome* is related to *fball:KickoffEvent* using the *fball:hasOutcome*. Though not shown, the *event:Product* and *event:hasProduct* are subsumed to generate the various types of football outcomes and *fball:hasOutcome* property respectively. In addition, domain-specific classes and properties are devised by way of subsumption of the *event:Factor* class to model the many types of factors that affect different types of football events associated with *Gridiron Football*. For example, *fball:WeatherFactor* and *fball:FieldConditionFactor* are related to *fball:GameEvent* using the *fball:hasEventFactor*. The *event:Factor* and *event:hasFactor* are subsumed to generate the various types of football factors and *fball:hasEventFactor* property respectively.

In addition, portions of the Time [OWL-TIME] [12] and The WGS84 Geo Positioning Ontology [Basic Geo] [13] vocabularies are used. The primary classes and properties include: *time:TemporalEntity*, which is a parent class that relates temporal information to an event; *time:Interval*, which is a subclass of TemporalEntity and temporal things with extent that have interior points; and *geo:SpatialThing*, which is a parent class that relates spatial information to an event. For brevity, the time and spatial ontological components used in the *fball* ontology are not described; however, spatial ontological components are illustrated in figure 3.

D. Metadata (RDF) Creation

This work involved creating efficient RDF representations for the NFL Play-by-Play data set using the *Gridiron Football Ontology*. The effort experimented with reasoning software

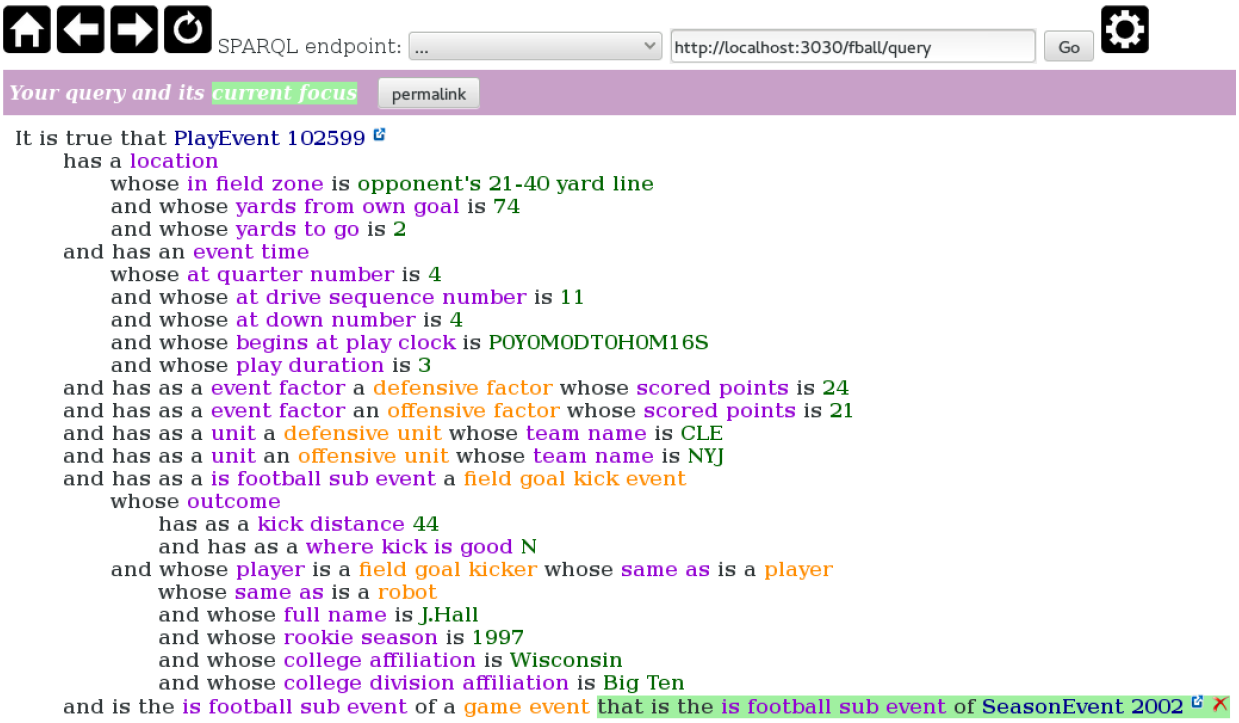


Fig. 4. An illustration of the *SPARKLIS* interface/engine being used to interrogate the RDF triples generated from the *Gridiron Football Ontology* and NFL data set. In this example, the attributes of location, time, factor, outcome and agent entities associated with this particular NFL play event are explored.

for the purpose of providing an inference reasoning capability to the project's software components for deriving new RDF triples (knowledge) from the instances generated directly from input data and its related ontology. This activity involved software development using the *Apache Jena* Open Source Java Framework for Semantic Web Development [14] to covert the elaborate NFL play-by-play data set, which is comprised of more than a half million game events and nearly 8500 unique agents, to a semantic graph containing 44,676,644 RDF triples.

E. Metadata Storage and Retrieval

Specifically, this effort consisted of the deployment a SPARQL End-Point web server and RDF-based triple store using the *Apache Fuseki/TDB* suite [15]. Ingest of triples into the triple store are primarily made by scripts or the upload feature in the *Apache Fuseki* web client. Queries are made through also made through the SPARQL interface within the Fuseki client as well as the *Sparklis: a SPARQL Endpoint Explorer for Expressive Question Answering* [16] web service. In Figure 4, an illustration of *SPARKLIS* being used to interrogate the generated NFL Football triples is given.

IV. Activity and Strategy Recognition

In this section, we apply our approach of ontology-based activity recognition to the *Gridiron Football* domain and try to show how event semantics may be used to help identify the base offensive scheme being used by a particular team during a football game. In *Gridiron Football*, an offensive scheme can be thought of as an offensive strategic system that a team uses to counter his opponent's defensive attack. Here, we show multiple components that may be decomposed from a team's overall offensive scheme according to [17]:

- Running Component: *Man/Power Blocking*, *Zone Blocking* and *Flex Blocking*
- Passing Component (Setup Mode): *Run to Setup the Pass*, *Pass to Setup the Run* and *Take What the Defense Gives You*
- Passing Component (Tempo): *Normal Tempo* and *Hurry-Up Tempo*
- Passing Component (Huddle): *Normal Huddle* and *No Huddle*
- Passing Component (Length of Passes): *Short to Intermediate* and *Vertical Intermediate to Deep Passing Game*
- Passing Component (Quarterback Position): *Under Center* and *Pistol Depth*, *Shotgun Depth*
- Passing Component (Route Assignments): *Route Tree Assignments (Air Coryell)*, *Group Assignments (Erhardt-Perkins)*
- Passing Component (How, Where, When): *Predetermined Pass to Spot Before Break*, *Predetermined Pass to a Person after the Break* and *Option Pass to a Person after the Break*

In our experiment, we attempt to identify these components of an offensive scheme. This is at least partially achieved by integrating a natural language-to-RDF query engine to our *fball* project's SPARQL-endpoint/RDF triple store. The integration of these two technologies allowed our research team

to interrogate the *fball* event knowledgebase for a collection of propositions (i.e., statements that are either true or false) that may be supported by the facts represented in the triple-store knowledgebase. Principally, a collection or sequence of statements would be derived to simulate a strategy-recognition pattern that matches against the RDF-encoded triples in the store and provide evidence of certain offensive scheme components. Identifying certain offensive components such as *Pass Component (Huddle): No Huddle* and *Passing Component (Tempo): Hurry-Up Tempo* was fairly simple and straight forward to accomplish. This ease to identify a particular offensive component was due primarily to completeness of data and that offensive component being directly available within the *fball* knowledgebase to directly support that question. In other cases, even with the detection of an offensive component or combination of offensive components it was not sufficient for identifying a more complex offensive strategy. To illustrate that point, here we focus on a particular type of offensive strategy called a *West Coast Offense* strategy. As background domain knowledge, we offer *football passing theory* that describes the *West Coast Offense* as the concept of using short passes to replace some of the running attacks [18]. Moreover, the short pass receiver is expected to run for good yardage after the completion. Therefore, a *West Coast Offense* strategy is a composite strategy of at least two of the components identified in the previous enumeration: the *Passing Component (Setup Mode)* and *Passing Component (Length of Passes)*.

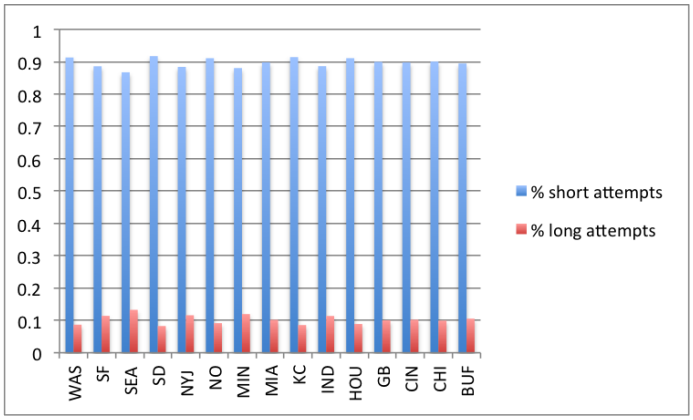


Fig. 5. An illustration of the average percentage of short-to-intermediate-distance pass attempts versus intermediate-to-long-distance pass attempts by NFL teams known to incorporate a *West Coast Offense* Strategy.

In figure 7, we show an example of the type of natural language query used by our researchers for detecting pass completions by a particular team during a single football season where the receiver or pass target caught the pass within 5 yards of the goal line and net gained 15 yards or greater after the pass completion. It follows that this particular team, which is the 2013 Miami Dolphins, coached by Joe Philbin and offensively coordinated by Mike Sherman, was known for using a *West Coast Offensive* strategy during the 2013 game season [19]. In figure 8, we show the results of the query.

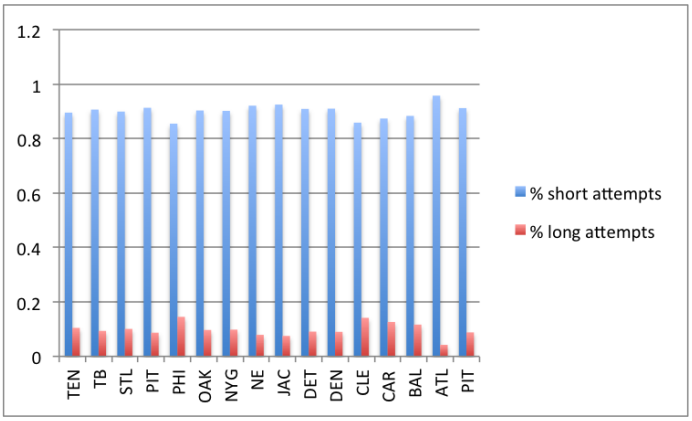


Fig. 6. An illustration of the average percentage of short-to-intermediate-distance attempts versus intermediate-to-long-distance pass attempts by NFL teams known to not incorporate a *West Coast Offense* Strategy.

Our researchers asked the following question: "In the time duration of a football game or season is this team using a *west Coast Offense* strategy as part of its offensive scheme? It follows that the research team proposed the following hypothesis, which is two-fold:

- In regard to the *Passing Component (Setup Mode)*, if the team demonstrates a higher percentage of *Pass to Setup the Run* attempts (i.e., passing attempts) than a team that demonstrates a higher percentage of *Run to Setup the Pass* attempts (i.e., rushing attempts); and
- in regard to the *Passing Component (Length of Passes)*, if a team demonstrates a higher percentage of *Short to Intermediate* pass attempts as compared to the percentage of *Vertical Intermediate to Deep Passing Game* passes.

In the evaluation of our detection pattern for the use of a *West Coast Offense* strategy in the base offensive scheme for a particular team, we were not able to identify teams that were definitely employing this strategy. In figure 5, we illustrate the average percentage of short-to-intermediate pass attempts and intermediate-to-long pass attempts that were executed by NFL teams that were known to use a West Coast Offense strategy as part of their base offensive scheme during the 2013 game season. In figure 6, the same statistics are illustrated; however, in this case we show the statistics of teams that were not known to use a West Coast Offense strategy during the 2013 Season. A quick visual examination of these bar charts show that there is not any major difference in the percentage of short- to long-distance pass attempts between the two category of teams (i.e., those known to use a West Coast Offense and those that do not). In addition to the statistics on *Passing Component (Length of Passes)*, statistics related to the *Passing Component (Setup Mode)* also did not show a major difference between teams known to use a *West Coast Offense* versus those that are not known to use that type of strategy.

V. Preliminary Results & Discussion

A number of insights were made as a result of our research effort. First, we believe that the complexity and

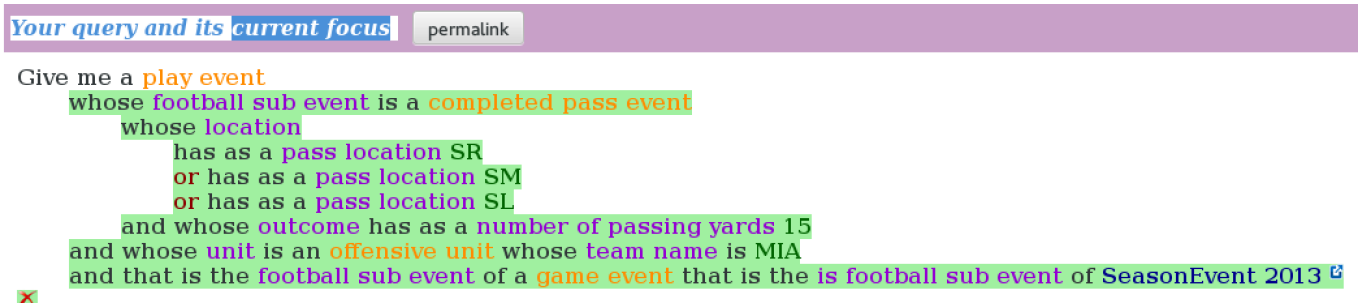


Fig. 7. An illustration of a natural language query using the *SPARKLIS* engine/interface that models the pattern of West Coast Offense scheme .

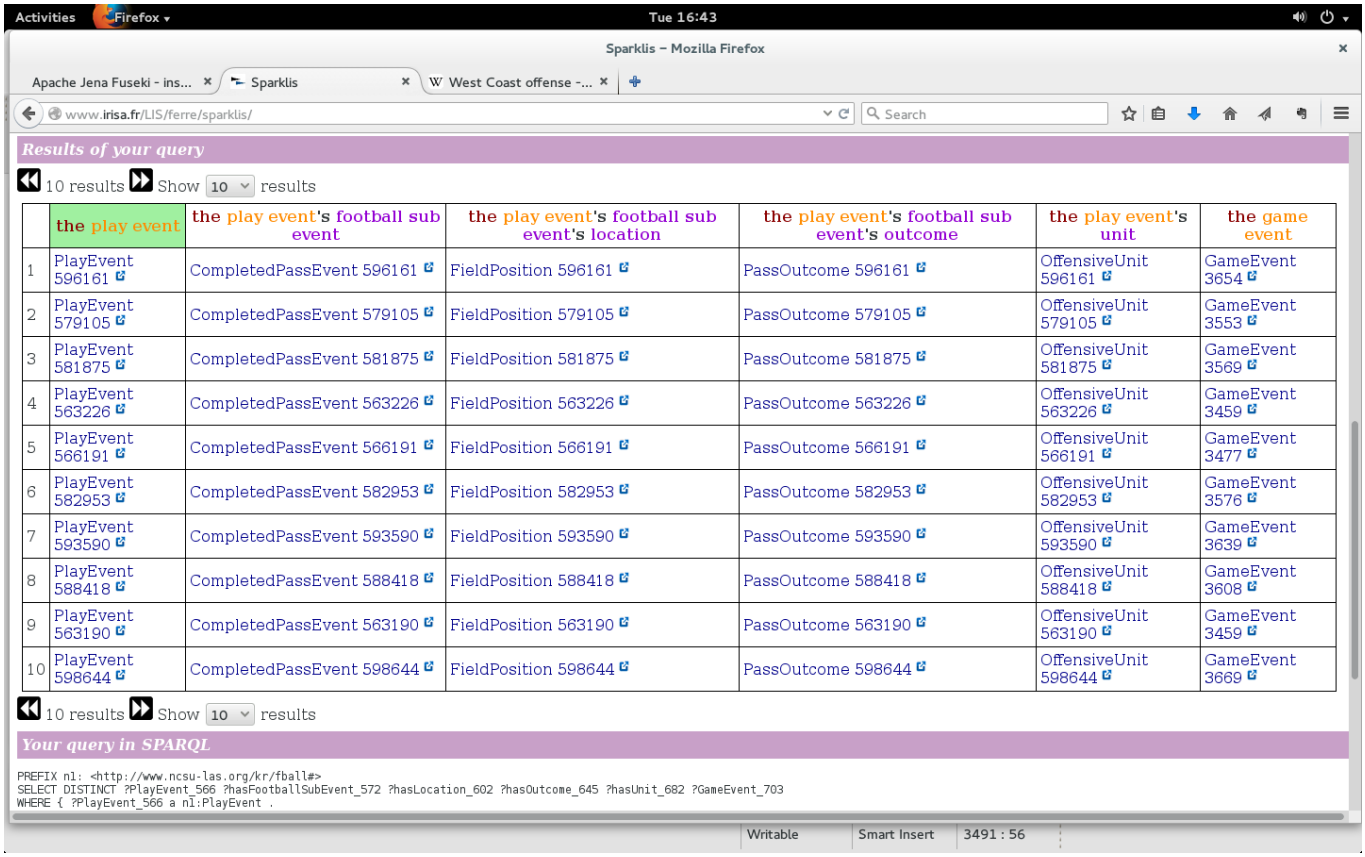


Fig. 8. An illustration of the query results from the West Coast Offense pattern-based query. Note the illustration shows observations of the *West Coast Offense* strategy being used by the Miami Dolphins football team during a game event in 2013

fine-granularity of the real-world, NFL Play-by-Play data set provided our team with an intermediate step for evaluating the application of semantic-based event models to the observations of human or autonomous agents engaged in competition at the scale we would expect within a cyber-physical system. As stated previously, NFL football players and stadiums are equipped with RFID sensors and we assert that the type of sensor data that we expect to be derived by an NFL Stadium Venue's cyber-physical system (CPS) will be similar to the data used by our research team. This type of activity data provided the research team with a unique challenge in effort to properly reflect in each RDF statement the appropriate semantic using the *Gridiron Football Ontology*. We expect that the modeling of other applications or domains involving human or autonomous agents competing or cooperating within a cyber-physical system would present a similar challenge.

The primary strength of the ontology-based activity (strategy) recognition approach is the relative simplicity and straightforwardness involved in incorporating domain knowledge and heuristics into the recognition models. The use of event semantics was especially beneficial in this regard. For instance, the The Event Ontology provides the ability to model and interrogate the NFL event knowledgebase based on five dimensions: event type, time, location, factor and outcome. Additionally, the upper ontology also provided an abstraction (i.e., *event:hasSubEvent*) for describing an event

that is composed of other events. This abstraction allowed the research team identify what type of football sub-events were associated with a particular play event, drive event, game event or season event. For example, a particular play event may be comprised of a penalty event and a pass event.

Though challenging, the research team determined that developing semantic queries that can detect certain strategies being used by NFL competing agents is possible. Initially, the research team exclusively relied on the development of SPARQL queries. Thereafter, the team learned that the guidance of an expressive Natural Language-to-RDF query builder such as *SPARKLIS* is useful for formulating straightforward queries for answering particularly complex hypotheses. The weakness of solely using the ontology-based strategy recognition approach is the lack of learning ability in terms of identifying patterns that can identify certain complex strategies. In this case, machine learning techniques for performing statistical and probabilistic reasoning may have been useful. However, the logical model approach (i.e., ontology-based activity recognition) can certainly play a dominant role when it is integrated along with techniques for learning patterns as well as dealing with the inability of the logical model to represent fuzziness and uncertainty. We offer that our approach and contribution is an intermediate step that can be further extended to include using an instance of an event ontology as a seed ontology for statistical and probabilistic strategy recognition. In some cases, this seed ontology may be used to develop a more comprehensive ontology using ontology learning techniques.

VI. Conclusion and Future Work

In this paper, an approach is given for capability that uses event semantics to represent the temporal, spatial, factor, and outcome characteristics of events generated from the observations of agents engaged in a competitive activity between each other. Further, we have described the likeness of the data set used for this experiment with the kind of data set we would expect to be generated from the type of "cyber-physical game" scenario identified in [2]. The approach extends existing modular vocabularies and is based in the specification of event semantics for our contributed *Gridiron Football Ontology* using Resource Description Framework Schema (RDFS) language and Ontology Web Language (OWL). Our future work has already begun and includes: extending the event ontology to an applicable data set for autonomous physical agents cooperating or competing within a cyber-physical system. Moreover, our effort seeks to integrate aspects of game theory analysis with the ontology-based strategy recognition approach to account for concepts such as payoffs and tensions between the different strategies that may be used by agents.

Acknowledgment

The authors would like to acknowledge the *Sparklis* research activity under Dr. Sebastien Ferre supported at IRISA, Universite de Remes, Rennes cedex, France. The authors would also like to acknowledge *Arm Chair Analysis* as the source of the NFL data set used in this research project.

References

[1] K. Namuduri, Y. Wan, M. Gomathisankaran, and R. Pendse, "Airborne network: a cyber-physical system perspective," in *Proceedings of the first ACM MobiHoc workshop on Airborne Networks and Communications*. ACM, 2012, pp. 55–60.

[2] T. Olavsrud, "The internet of things comes to the nfl," Jul 2015.

[3] S. Trimble, "Google targets low-cost ads-b out avionics market," 2015.

[4] J. Durham, *Football's Matchup Zone Coverages*. Harding Press, 1998. [Online]. Available: https://books.google.com/books?id=qhcKAAAACAAJ

[5] D. Jennings, M. Bahr, R. Danmeier, and D. Herbst, *The Art of Placekicking and Punting*. Linden Press/S&S, 1985. [Online]. Available: https://books.google.com/books?id=Z31YAAAAYAAJ

[6] J. Rice, *Defending the Spread Offense*. Coaches Choice, 2010. [Online]. Available: https://books.google.com/books?id=lOkXRAAACAAJ

[7] A. F. C. Association, *Offensive Football Strategies*. Human Kinetics, 2000. [Online]. Available: https://books.google.com/books?id=pHbvMbMZKjQC

[8] B. Ramseyer, *Winning Football*. Human Kinetics 1. [Online]. Available: https://books.google.com/books?id=IuxRAgAAQBAJ

[9] A. J. Cañas, G. Hill, R. Carff, N. Suri, J. Lott, T. Eskridge, G. Gómez, M. Arroyo, and R. Carvajal, "Cmaptools: A knowledge modeling and sharing environment," in *Concept maps: Theory, methodology, technology. Proceedings of the first international conference on concept mapping*, vol. 1, 2004, pp. 125–133.

[10] N. F. Noy, M. Crubézy, R. W. Fergerson, H. Knublauch, S. W. Tu, J. Vendetti, M. A. Musen *et al.*, "Protege-2000: an open-source ontology-development and knowledge-acquisition environment," in *AMIA Annu Symp Proc*, vol. 953, 2003, p. 953.

[11] Y. Raimond and S. Abdallah, "The event ontology," Technical report, 2007. http://motools. sourceforge. net/event, Tech. Rep., 2007.

[12] J. R. Hobbs and F. Pan, "Time ontology in owl, w3c working draft 27 september 2006," *W3C Working Draft*, vol. 27, 2006.

[13] D. Brickley, "Basic geo (wgs84 lat/long) vocabulary, 2006," *Cité en*, p. 52.

[14] A. Jena, "A free and open source java framework for building semantic web and linked data applications," *URL: http://jena. apache. org*, 2011.

[15] A. Jena-Fuseki, "serving rdf data over http," 2014.

[16] S. Ferré, "Sparklis: a sparql endpoint explorer for expressive question answering," in *ISWC Posters & Demonstrations Track*.

[17] G. Morris, "Football 101: Understanding basic nfl offensive concepts," 2014. [Online]. Available: http://www.bloggingtheboys.com/2014/8/11/5965033/football-101-understanding-basic-nfl-offensive-concepts

[18] T. Flores and B. O'Connor, *Coaching Football*. McGraw-Hill Education, 2006. [Online]. Available: https://books.google.com/books?id=mrsIBAAAQBAJ

[19] W. C. offense, "West coast offense — wikipedia, the free encyclopedia," 2015, online accessed 02-September-2015. [Online]. Available: https://en.wikipedia.org/wiki/WestCoastoffense

Position Paper

A Semantic Approach to Reachability Matrix Computation

Nicole Dalia Cilia
Dept. of Philosophy,
Sapienza University of Rome,
Via Carlo Fea 3, Rome, Italy,
nicole.cilia@uniroma1.it

Noemi Scarpato
University Telematica San Raffaele
RomaVia di Val Cannuta,247 Rome,
Italy,
noemi.scarpato@unisanraffaele.gov.it

Marco Romano
Epistematica S.r.l.
Via Ostiglia 10, 20133, Milano, Italy,
m.romano@epistematica.com

***Abstract*— The Cyber Security is a crucial aspect of networks management. The Reachability Matrix computation is one of the main challenge in this field. This paper presents an intelligent solution in order to address the Reachability Matrix computational problem.**

Keywords— CyberSecurity;Ontologies; Reasoning.

I. INTRODUCTION

In this paper we describe our contribute in the PANOPTESEC[1] project. PANOPTESEC aims to deliver beyond-state-of-the-art prototype of a cyber defence decision support system, demonstrating the benefit of a risk based approach to automated cyber defence. PANOPTESEC takes into account of the dynamic nature of information and communications technologies (ICT) and of the constantly evolving capabilities of cyber attackers in order to propose a solution based on knowledge representation and reasoning.

Recently, various studies provided progress in the Cyber Defence domain with data models and methods to focus the security problem in large networks. Morin et al. [1] have provided "a data model for security systems to query and assert knowledge about security incidents and the context in which they occur. This model constitutes a consistent and formal that an organization implements with an ICT system".

In order to better assess the effect of countermeasures to cyber-attacks and better rank countermeasures, PANOPTESEC provides a list of requirements for a system for mission impact assessment. Information about ICT assets and their vulnerabilities is used in order to compute known ways to attack a system (so-called attack graphs). Reachability Matrix is the input for the Attack Graph Generator. The PANOPTESEC approach to cyber-security maintenance support is based on a model of relations between business services and the supporting ICT assets. Business services represent the mission assessment. Information about ICT assets and their vulnerabilities is used in order to compute known ways to attack a system (so-called attack graphs). Reachability Matrix is the input for the Attack Graph Generator.

The scope of this paper encompasses data collection and correlation for the ACEA use case, but also provides a generalized approach for cyber security domain. We propose a semantic approach that applying the formalism of Description Logics [1] to the Cyber Security domain.

Following we describe the Reachability Matrix Correlator (RMC) component, the Reachability Matrix Ontology (RMO) and the reasoning task. RMC provides algorithms for computing reachability information. RMO describes the Cyber Security domain. Reasoning task uses the RMO ontology in order to compute the Reachability Matrix. Our approach foresees that RMC populates the RMO with input data provided by PANOPTESEC Data Collection and Correlation system (see section II and III for more details) and applies a set of SWRL rules and SPARQL queries to compute the Reachability Matrix (see section IV for more details). Reachability Matrix is employed in PANOPTESEC to determine if a node can reach another node (via ISO/OSI layer protocols), this information is crucial to risk management.

II. REACHABILITY MATRIX CORRELATOR

As shown in **Fig. 1** the PANOPTESEC architecture includes: Visualization System, data Collection and Correlation System, Dynamic Risk Management System, Integration Framework and Monitored System. The Data Collection and Correlation System (DCC) has the goal of providing suitable data to all other components required for building a cyber-security protection system. The Reachability Matrix Correlator component is part of the Data Collection and Correlation System.

The role of DCC in the PANOPTESEC project is to develop a data collection and a correlation engine for building an advanced cyber-security maintenance system.

The Data Collection and Correlation System, which also contains a model for describing the impact of cyber-attacks as well as corresponding countermeasures (based on a mission model), will provide the necessary input for other components of PANOPTESEC. DCC is composed by five main components: Data Collection Interface, Data Collection Collector, Low-Level Correlator, Reachability Matrix Correlator and Mission Impact Module (see **Fig. 1**). The DCC module needs to avoid duplicate data handling and complex synchronization principles. RMC provides the reachability matrix, useful for the Attack Graph Generation component of

[1]http://www.panoptesec.eu/

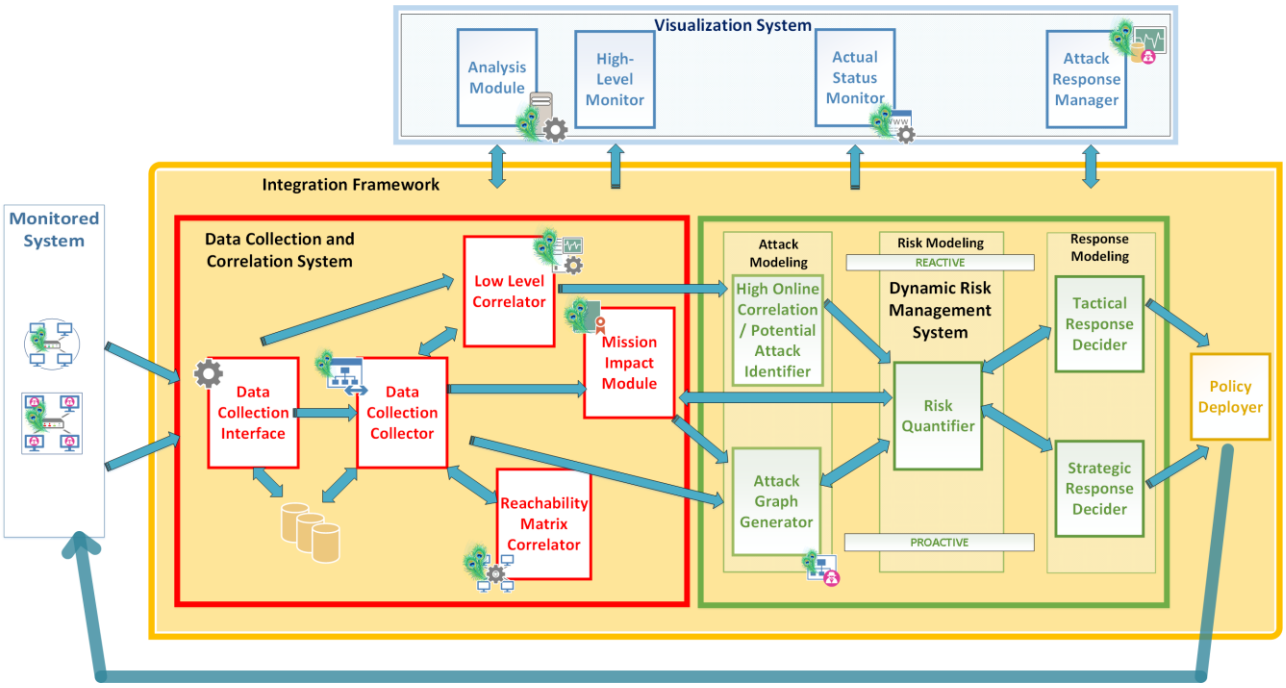


Fig. 1: PANOPTESEC global architecture and logical data flows.

the Dynamic Risk Management Response System. The reachability information are used to determine if a node can reach another node (via ISO/OSI layer protocols).

RMC performs the reachability computation across the monitored ICT network to deduct if two nodes are reachable from each other in the network, for all pairs of nodes representing ICT devices. To achieve the RMC goal we need to produce an abstract machine-readable representation of the knowledge, the RMO (see next section for details). As shown in **Fig. 2** RMO is imported from an external file and stored in the Graph Database Sesame [2] by the T-Box Loader, this operation happens once at the initialization of the RMC and successively only if RMO is changed. Then A-Box Loader populates the RMO with the information regarding the Network Inventory, Deploy Access Control Policy and Mitigation Action. Finally the reachability correlation engine computes the Reachability Matrix using information stored into Knowledge Base.

The RMC must:

- determine if a node is reachable from another node on a logical level. To provide at a logical level if a node can be reached from another node If in a network a node is reachable from another node, there is a possibility that an adversary might be able to infiltrate a network further. Such information is gatherable from, e.g., Firewall Rules, Mapping Rules, Firewall Logs and/or Traffic Captures.
- determine reachability in terms of Source-Port, Target-Port, Protocol to obtain a detailed view of reachability in a network and provide the most available information if a node is reachable over a specific port and protocol, there might exist vulnerabilities in such a protocol. Further a node might be reachable, but this reachability does not allow an adversary to progress further in a network.
- identify physical entities responsible for a reachability. Identify hardware entities, e.g. Firewalls, Switches, Routers, that route a reachability on a physical level. A logically non-existing hardware, e.g. a switch, but itself be prone to vulnerabilities, which might allow an adversary to broaden a reachability.
- consider that a node might be known via multiple addresses To identify a reachability on a logical level between unique devices in a subnetwork, entities might be addressed (e.g. IP) in another way, than from outside the subnetwork.

The requirements presented are secondary to:

a) Identifying and defining an adequate representation of knowledge (ontology), the proper Knowledge Base and the appropriate Knowledge Repository [2] to store the Knowledge Base,

b) Defining a proper mode to populate the Knowledge Base.

Several tasks are needed to achieve the RMC goal:

a) To study and produce a correct representation of the problem, using the most suitable available methods.

b) To perform applied research to determine the optimum methods to solve automatically the problem.

c) To guarantee that the representation of the knowledge, the representation of the problem and the solving method are abstract enough to be independent of any specific commercial networking devices or applications.

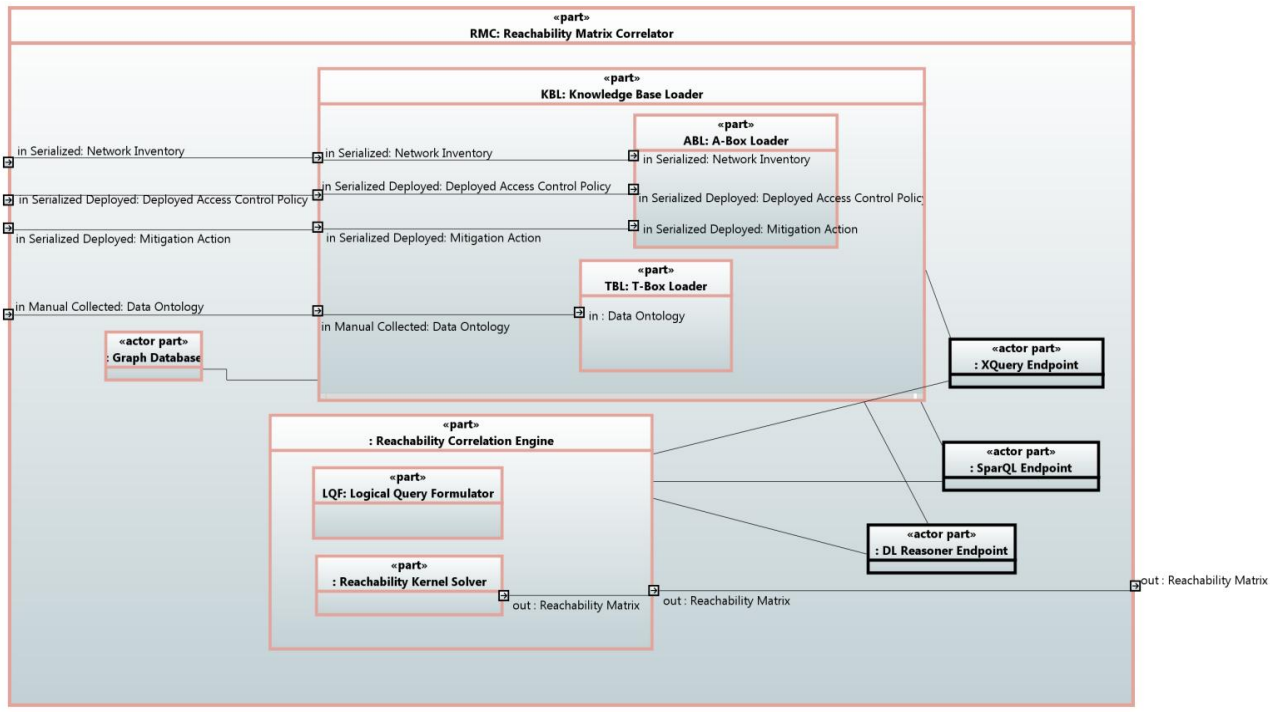


Fig. 2: Component View: Reachability Matrix Correlator (Internal Blocks Diagram)

d) To integrate the available tools and methods into a running prototype correlator that produces the correct Reachability Matrix.

e) To guarantee that the IT services provided by the Correlator are be aligned to the needs of the Attack Graph Generation component of the Dynamic Risk Management Response System .

f) To guarantee the compatibility of the technical solution with the overall PANOPTESEC system

g) To guarantee that the technical implementation of the solution within PANOPTESEC system is performing at an acceptable service level for a prototype within the project scope and quality requirements.

III. THE KNOWLEDGE BASE

As mentioned above RMO ontology [3] represents connectivity, Network Inventory, Access Control Policy and Mitigation Actions. RMO has the following characteristics:

- Capable of representing all kind of ICT devices, including terminal machines,
- Able to represent connectivity including sub-netting,
- Able to describe ICT devices grouping, according to filtering rules in machines,
- Capable of representing gateways and firewalling rules,
- Having the capacity to represent connectivity between every kind of nodes in the ICT network,
- Able to represent Deployed Access Control Policy Rules,
- Able to represent Deployed Mitigation Actions.

In order to design RMO used within the RMC component, we faced a thorough and deep study of IP networks in order to identify all objects that come into play in such a domain, with all their characteristics, the relationships between them, and the role each element plays in successful communications over IP networks. This study has gone in parallel with the analysis of the Data Model that we have been provided with, with special regards for the schemas concerning the Network Inventory and the Deployed Access Policies (which are basically the format of the data incoming into our component as its input), and the schema for the Reachability Matrix which determines the format for the data that exits our component.

The resulting knowledge representation, which is an OWL [4] ontology, provides a reconciled vision of these partial Data Models, in such a way that the ontology has “room” enough to receive all data from Network Inventory (and the other input files about routing tables, firewall rules and NAT rules) so as to compose a Knowledge Base (static T-box, plus the A-box reloaded over time), and to re-model the data so as to fit the output data format required by the Reachability Matrix data model. Of course, input and output formats described in the It Data Model do overlap, since are different views of the same matter. More precisely, the output that we produce contains a subset of all the information which is in the input that we receive, but enriched with some “new” information.

This is the information made explicit by automatic reasoning, thanks to the “logical embedding” of the additional knowledge about the functioning of IP networks (derived from our initial study) that is recorded in the ontology (in particular in the T-box). While designing the ontology, the classes described in the Data Model schemas, with all their attributes, needed to be re-modelled to fit the different representation

paradigm of ontologies. The most typical cause of intervention is the need to distinguish objects - and their relationships with other objects (possibly with different types of objects) - from simple values that express attributes of the objects (sort of terminal, minimal points of information about which is not possible to say anything else). Briefly, at the present stage of development, the ontology has an expressivity well within OWL-DL [4] expressivity (allowing for good performance of reasoning). It counts with 37 named classes (i.e. concepts in the ontology T-box) that collect the objects accounted for in the Network Inventory and the other input files. There also 37 different relationships (object properties according to the OWL terminology) to represent the possible relationships among the objects of this classes, and other 55 (datatype) properties to account for all other characteristics of the objects.

The most important part for the function of our module (which at present is focused on reachability at the layer 3 of the OSI model) is the part that accounts for: nodes identification, network interfaces, network they belong to, and routing instructions to reach other networks, i.e. the routes and the complex information to describe them: the source (node&interface), the destination (network), and the gateway to pass through. Besides the classes that collect the objects of these various types, a set of 12 object properties allow to logically model the reachability between nodes. These (object) properties deal with:

- the network interfaces belonging to some node
- the network that each interface is connected to
- and, as a consequence, the networks that a node belongs to.

But also they deal with the other networks that can be reached by passing through one or more gateways, based on routing instructions. Finally, a set of 4 SWRL [5] (Semantic Web Rule Language) rules "force" the reasoner [6] to compute, for every interface of a node, every other node it can reach to (further details on this regard in the next section).

IV. ABOUT THE REASONING

The very first reasoning service used with regard to our ontology is the consistency check of the T-box, which is run at the design time of the ontology. Of course, the ontology passed this check. Subsequent check is the validation of the entire knowledged base. Once the A-box is loaded along with the T-box, and the whole KB is loaded into the framework of our component, this second service checks whether the A-box – produced based on the input data (Network Inventory and other files) – is consistent with respect to the T-box. Normally, a fail in this check would highlight an error in the way the input data is translated into the A-box, hence still an error in the ontology design.

The most interesting part is the reasoning triggered by the SWRL rules that rely on information stored in the Knowledge Base. These rules are typical logical rules of the form:

IF condition1 and … condition N THEN consequence.

The rules provided along with our ontology describe all possible scenarios to be investigated in order to detect all the nodes that are reachable from any given couple made of a node (with its specific routing instructions) and any of its network interfaces (connected each to one particular network).

The first rule covers the case of all reachable nodes within the same domain to which a given node belongs. The second rule covers the case of all reachable nodes within some known networks for which special routing instructions are given. The third rule covers the case of all other networks not known in advance, yet reachable through a series of "hops" to default gateways. Though absolutely necessary, the fourth rules does not cover any special case. It only enforces the reasoning in such a way that the transitivity of the relevant relationships (object properties in the ontology describing the functioning of "hopping" through gateways) is properly taken into account by the reasoner [6]. The execution of the reasoning based on all the information within the Knowledge Base and the four SWRL rules, allows to produce the set of all pairs made of a network interface and the nodes it can reach (discovered by looking at every network interface of a node, its direct connections and the routing instructions given to the node it belongs to). Here we have all information needed to produce the reachability matrix (as it is at present stage, at layer 3 of the OSI model).

Last step to produce our output – the Reachability Matrix – for use on the part of the other components is to explicitly point out, for each network interface of any given node, the set of all and only the other nodes that it can reach. However, this is not properly speaking reasoning, since it is just retrieval of triples (the form in which data are declared in OWL), and it is achieved by firing some SPARQL [7] queries (actually embedded in the APIs [8] of the persistence environment that we adopt). Other similar queries retrieve the rest of information that is available in the Knowledge Base and is expected in the Reachability Matrix according to the output format.

V. ACKNOWLEDGEMENTS

This paper has been supported by Epistematica (http://www.epistematica.com/) within the PANOPTESEC project.

REFERENCES

[1] Morin, B, L Mé, H Debar, M Ducassé (2009). A logic-based model to support alert correlation in intrusion detection . *Information Fusion*, 10 (4), 285-299.

[2] Sesame (http://rdf4j.org/).

[3] Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing?. *International journal of human-computer studies*, *43*(5), 907-928.

[4] McGuinness, D. L., & Van Harmelen, F. (2004). OWL web ontology language overview. *W3C recommendation*, *10*(10), 2004.

[5] Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B., & Dean, M. (2004). SWRL: A semantic web rule language combining OWL and RuleML. *W3C Member submission*, *21*, 79.

[6] Haarslev, V., & Müller, R. (2001). RACER system description. In *Automated Reasoning* (pp. 701-705). Springer Berlin Heidelberg.

[7] Prud'Hommeaux, E., & Seaborne, A. (2008). SPARQL query language for RDF. *W3C recommendation*, *15*.

[8] Stellato A. OWLART API (http://art.uniroma2.it/owlart/).