CHAMPION: Intelligent Hierarchical Reasoning Agents for Enhanced Decision Support

Ryan E. Hohimer, Frank L. Greitzer, Christine F. Noonan, Jana D. Strasburg
Pacific Northwest National Laboratory
Richland, WA

***Abstract* —** We describe the design and development of an advanced reasoning framework employing semantic technologies, organized within a hierarchy of computational reasoning agents that interpret domain specific information. The CHAMPION reasoning framework is designed based on an inspirational metaphor of the pattern recognition functions performed by the human neocortex. The framework represents a new computational modeling approach that derives invariant knowledge representations through memory-prediction belief propagation processes that are driven by formal ontological language specification and semantic technologies. The CHAMPION framework shows promise for enhancing complex decision making in diverse problem domains including cyber security, nonproliferation and energy consumption analysis.

Keywords — Semantic Graphs, Description Logic Reasoning, Belief Propagation, Memory-Prediction Framework, Case-Based Reasoning, Ontological Engineering

I. Introduction

A major challenge for information analysis is to develop *joint cognitive systems*, described by Woods [1, 2] as systems in which humans interact with another, artificial, cognitive system. Cognitive systems are goal-directed, using knowledge about "self" and the environment to monitor, plan, and modify actions in pursuit of goals. They are both data-driven and concept-driven. Woods observed that "developments in computational technologies (i.e., heuristic programming techniques) have greatly increased the potential for automating decisions" and for "… the support of human cognitive activities…." [1] A single, integrated system was envisioned at that time that could be composed of both human and artificial cognitive systems working collaboratively to perform complex decision making tasks. In the quarter-century that has passed since this vision was described, many different types of intelligent systems and processing frameworks have been proposed and developed, though it is not clear that the vision of joint cognitive systems has been realized. The current research and development effort represents a serious attempt to bring us closer to this vision utilizing semantic modeling.

II. Background

Understanding how the human brain works is one of science's grand challenges [3]. A great deal of effort has been devoted to the development of data-driven approaches to information analysis, inspired by neuroscience, in particular the neuron. A neuron is a cell in the brain whose principal function is collection, processing and distribution of signals. These signals are propagated through networks of neurons controlling brain activity and formulating the basis for human learning and intelligence including perception, cognition and action. Artificial intelligence (AI) as a field of inquiry has been around for decades and currently encompasses a large number of subfields intersecting biology, engineering and complex systems [4-6].

Properties of biological memory systems motivate the sub-field of artificial neural networks (ANN), one type of computational model representing a bottom up or data-driven approach [7]. Feed-forward or recurrent ANNs learn by example and are able to model nonlinear systems. They require data for training the network, which is not always available. From the decision support perspective they have the disadvantage of being "opaque" to the user [8]—that is, the distribution and weights of the neural network connections are not sufficiently specified to offer insight into their operation; and this clearly doesn't facilitate collaboration of joint cognitive systems.

Machine learning is a mature field focused on programming computers to optimize performance based on past experience. The goal with this type of research is to develop general purpose systems that can adapt to new circumstances and domain knowledge [9]. A disadvantage of machine learning approaches, when coupled with human decision makers in a joint cognitive systems context, is similar to that described above for ANNs and connectionist solutions to the extent that the workings of the machine learning component are not readily understood or communicated to the human decision maker.

In contrast to these data-driven approaches, research in knowledge-based/expert systems has focused more on concept-driven or top-down reasoning. Top-down reasoning tries to mimic the brain's functions such as memory. This area of AI is concerned with thinking; how knowledge is represented symbolically and manipulated and how it contributes to intelligence.

Bayesian Network (BN) modeling approaches have become a rapidly growing area of research aimed at modeling human cognitive and decision making behavior, reflecting a perspective that use of probabilistic models and associated computational power of the Bayesian mathematical framework greatly facilitates the representation of human performance within a rational decision making framework. BN models can be viewed graphically to represent probabilistic relationships in a given domain; hence they are more readily comprehended by users. Nevertheless, there are un-answered questions regarding the appropriateness of using the Bayesian probability construct, which reflects the assumption that human decision processes may be explained in terms of rational/normative models [10].

Logic-based/rule-based systems comprise a structured collection of rules. A long-standing top-down approach is the use of logic, as represented in rule based expert systems. A major difficulty in implementing such knowledge-based systems is the difficulty of collecting expert knowledge that must be represented in the collection of rules that comprise the knowledgebase. The use of semantic web technology provides an expressive knowledge representation using ontologies, along with the application of Description Logics, which provides a formal knowledge representation language that facilitates generation of conclusions or predictions.

Unlike most problem solving techniques in artificial intelligence, case based reasoning (CBR) is memory based. Solving a problem using the classic CBR cycle involves four major components - retrieve, reuse, revise and retain (see Figure 1) [11, 12]. CBR systems are concept-driven and rely on the recognition of previously-learned (hard-coded) or experienced representations to determine the system's response to new information. A challenge for the CBR approach is the development of efficient and effective methods to search the repository of cases (stored in case memory).

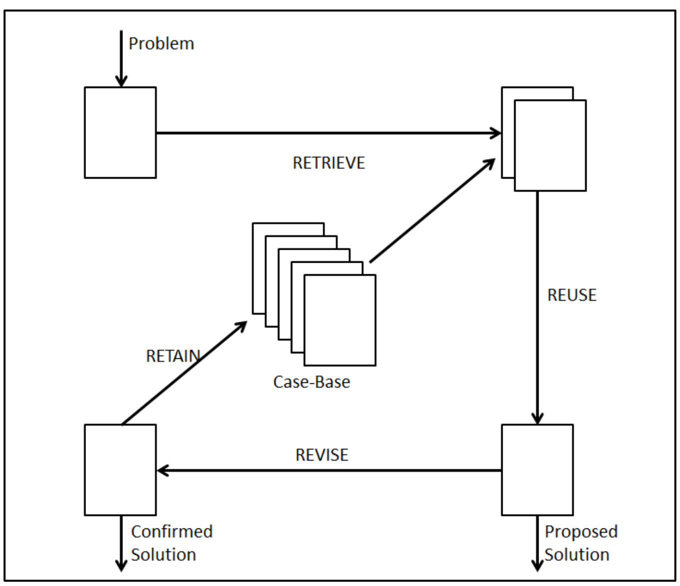

Figure 1. The CBR Cycle, adapted from [13].

A relatively recent top-down approach showing great promise is the memory prediction framework (MPF) [14, 15]. The MPF defines how the neocortex uses a feedback loop to store memory patterns which can lead to prediction of future events. These higher level concepts of cognitive processing have been applied in our work in development of the CHAMPION system.

We advance many of the aforementioned artificial intelligence concepts through extensive use of semantic technologies. With our modeling architecture, we separate domain knowledge from the reasoning framework. This is done to maintain flexibility with domain knowledge, allowing it to be updated as needed, and to ensure domain agnosticism, allowing the system to be implemented in many fields of inquiry.

III. System Design

The neocortex was the *inspirational metaphor* for the design of our reasoning framework, called CHAMPION (for Columnar Hierarchical Auto-associative Memory Processing In Ontological Networks). This metaphor serves as a representation for a functional (not structural) design adopting the following requirements :

- Stores sequences in an invariant form
- Stores sequences of patterns
- Stores sequences in a hierarchy
- Retrieves sequences auto-associatively

The CHAMPION architecture incorporates a significant variation on knowledge intense case based reasoning (KI-CBR) depicted in Figure 2. Modifications to the traditional CBR cycle were invented in order to meet the functional requirements of this metaphor.

- Instead of iteration through the case library to find a useful solution, our system uses semantic expressions to represent the criteria for a case belonging in the case library. We consider this an invariant form of a concept belonging to the set of cases.
- The functional requirement to store sequences of patterns is met by representing the problem and solution spaces in the form of semantic graphs. The nodes and edges constitute the patterns.
- The architecture uses the query/construct capabilities of SPARQL and programming pattern paradigm of "Publish and Subscribe" to implement an auto-associative mechanism.
- The domain ontology of the system addresses the functional requirement to store the concepts in a hierarchy.

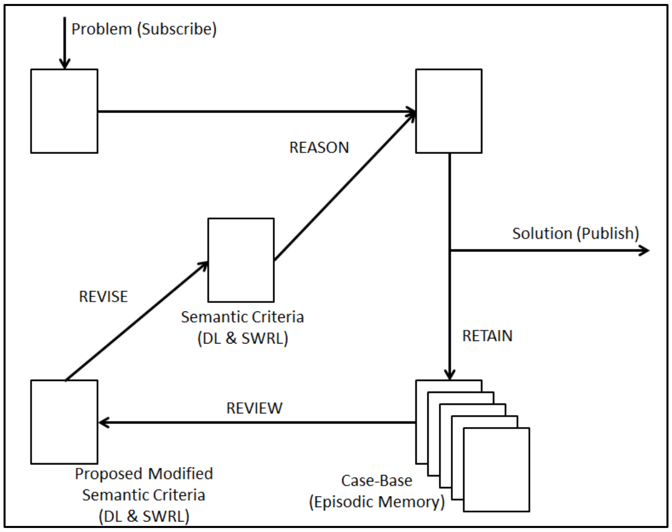

Figure 2. The CHAMPION modified CBR cycle

The CHAMPION reasoning framework consists of a hierarchy of reasoning agents called Auto-associative Memory Columns (AMCs). The hierarchy is formed as each agent subscribes to subgraphs of interest from a base graph and publishes subgraphs back to the base graph (i.e. making the base graph an inference graph).

Agents interpret data in a similar fashion as subject matter experts. The lowest level agents in the hierarchy interpret the rawest form of data, and pass their interpretation of that data up the hierarchy. Primitive data goes in the bottom and higher level interpretations come out the top.

A basic premise adhered to is the separation of the domain knowledge from the reasoning framework. If domain knowledge is hardcoded within the reasoning framework, then the framework's source code must be changed and recompiled frequently as domain knowledge is updated. Equally important is the fact that this separation of domain knowledge from the reasoning framework maintains the domain agnostic quality of the system, which enables its application to diverse problems without modification to the reasoning framework. We use the Ontology Web Language (OWL) as our knowledge representation language, to implement the ontologies and knowledgebases of the system.

The main components of the CHAMPION system, shown in Figure 3, are:

- *Ontologies*, used for representing the specialized domain knowledge.

- *Reifiers*, used for ingesting the primitive data as individuals of the types specified in the domain ontologies.

- *Memory*, used to store the facts asserted from the primitive data and the facts inferred by the reasoning system.

- *Auto-associative Memory Columns (AMCs)*, reasoning components used to interpret the data assertions and infer new assertions.

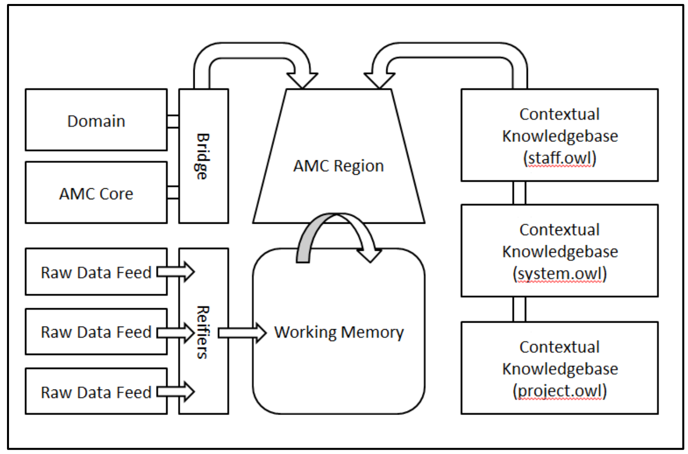

Figure 3. The components of the CHAMPION system

A. CHAMPION Ontologies

There are four key ontologies in the CHAMPION system, each having a unique purpose: Domain Ontology, Core Ontology, Bridge Ontology, and a collection of Rules Ontologies.

1) Domain Ontology

The content in the domain ontology is the knowledge of the subject matter expert in the domain of discourse to be reasoned about. It is expected that the specialized terminology of interest be captured in this T-Box ontology. If the domain of interest is Insider Threat, concepts used by experts in this field are defined here. Concepts specifically about aspects of trusted persons, their access, privileges, roles, responsibilities, and authorities would be defined. Additionally, concepts of the enterprise within which they function would be defined, such as concepts related to the infrastructure and business systems.

2) Core Ontology

The content of the core ontology is the knowledge of the reasoning framework and its elements. The definitions that describe what the necessary components of the AMCs are encoded into this ontology. The primary concept defined in this ontology is the AMC. The AMC is the primary reasoning agent of the framework and the class definition of the AMC is found in the core ontology.

3) Bridge Ontology

The bridge ontology associates concepts in the domain ontology with concepts from the core ontology. In other words, this is the place where domain concepts are assigned an AMC to reason about them.

Continuing with the Insider Threat domain, let's assume the concepts of *access* and *unauthorized access* are defined in the domain ontology as Access and UnauthorizedAccess respectively. In this example, Access is the superclass of UnauthorizedAccess. In the bridge ontology we encode that an AMC is assigned to reason about UnauthorizedAccess (the AMC class is subclassed to be an UnauthorizedAccess). The UnauthorizedAccess AMC is further defined to subscribe to

Access individuals, and publish UnauthorizedAccess individuals. Later in this paper, we will see that this is a subsumptive AMC.

4) Rules Ontologies

An AMC in the reasoning framework is to publish the appropriate assertions that are entailed in the local AMC's graph. Two governing ontologies are applied to the local AMC, 1) the domain ontology, and 2) an AMC specific ontology which contains knowledge that is relevant to the local AMC only. The consequence of having an ontology at the AMC granularity is that a rules ontology must exist for each AMC.

B. Knowledgebases

In addition to the ontologies, the following knowledgebases are required: Working Memory, AMC Knowledgebases (Binning Queue, Case Library), and a Contextual Knowledgebase.

1) Working Memory

The Working Memory knowledgebase is the semantic graph containing the state of the base-graph and the inference-graph assertions. This is the location of all the individuals from reifiers and from AMCs.

2) AMC

Each AMC has to have a local knowledgebase over which it can reason. The local knowledgebase directly imports the bridge ontology, which in turn indirectly imports the core and domain ontologies. Additionally, each AMC has a dedicated ontology that contains semantic expressions specific to this AMC. These expressions include SWRL rules that the local AMC's description logic reasoner evaluates.

3) Contextual Knowledge

Additional knowledge beyond the streaming problem data under analysis or search is stored in contextual knowledgebases. This type of knowledge needs to be accessed by the AMC in order to do informed searches or analysis. For example, to correctly reason about an activity associated with a username, the AMC must be able to access information about that username, such as the roles and access controls that are associated with that user.

C. Auto-associative Memory Columns

The analysis of real world data presents a challenge to computationally analyze very large graphs. The difficulty is not so much a *data reduction* problem as it is a *data interpretation* problem. A traditional approach to analyzing large graphs is to build the graph and then conduct reasoning over the entire graph. In contrast, the CHAMPION hierarchy of reasoners comprises a "stack" of individual AMCs which reason over the data as it is introduced into the system in much smaller graphs than the entire dataset. The larger graph structure is built as data are analyzed; this produces a dynamic belief propagation network that takes in primitive data and pushes the interpretation of that data up the hierarchy. We can think of this as interpreting the current structure in the data and simplifying with abstracting semantics. Just as we can stack the AMCs, we can stack collections (*regions*) of AMCs that address reasoning or pattern recognition for different domains. Similarly, even higher level collections of AMCs enable reasoning across such regions, providing a natural mechanism for high level information fusion and analysis.

Using a hierarchical framework of reasoners allows us to constrain the requirements of each reasoner to a narrowly-defined purpose. There is almost a one to one relationship between AMCs and the classes defined in the domain ontology. With a well-formed domain ontology, we can overcome computational intractability by performing reasoning on *subsets* of the semantic graph. Rather than implementing a monolithic reasoner that is required to reason over all the concepts represented in the semantic graph, each reasoner in the hierarchy is only required to reason about a small set of relevant concepts.

The belief propagation network performs a transformation of the low level literal inputs into higher level abstractions. Ingesting and properly formatting the input data for a given domain is performed by a *reifier*, which instantiates the input from a data source and packages the information into an OWL representation called an *individual*. In turn these individuals are instantiated in Java objects called *abstractions.* The *abstractions* are added to the *Working Memory* of the CHAMPION system.

D. Reifiers

Reifiers are responsible for asserting *individuals* (primitives) into the Working Memory via *abstractions*. Although AMCs are domain agnostic, this is not possible with the reifiers. The reifier takes in raw literal data and forms an *individual* that is defined by the domain ontology. When raw data needs to be reified, specific code is required to convert the raw data into a data-type defined in the domain ontology.

E. Provenance Information

Provenance has been defined as the description of the origins of data and the process by which it came to exist [16, 17]. Clearly this is an important requirement for the system that will facilitate the decision maker's understanding of the reasoning process. The system has two locations where provenance information can be stored. The first is in the asserted individuals added to the graph. Reified individuals (i.e. individuals from a reifier) and inferred individuals (i.e. individuals from an AMC) can have data properties asserted specifying their time and source of instantiation. The second location for storing provenance information is the episodic memory of the AMCs. Each AMC has an instantiation history of all the individuals that it has classified as being a member of its governing class. This constitutes its case library, comprising each inference graph the AMC has asserted into the base graph.

To date we have not focused on collection of provenance information. However, in future research we wish to use provenance information for two significant purposes: 1) intelligent rollback to a point of logical consistency, and 2) adaptive machine learning of higher level class resolutions based on case library analysis.

IV. An Agent's Purpose

A. Initial Base Graph Assertions are "Primitives"

The first assertions into the base graph are defined as "primitives." These are not primitives in the same sense as how programming languages define them, but in the sense that they are defined by a subject matter expert. These primitives are nodes that are believed to be assertions with very low uncertainty. For example, the data reified into the base graph could be computer workstation events such as security events, application events, and system events. No assumptions are made about the events; they occurred and the information is reified into the base graph. However, as reasoning agents infer new assertions based upon these primitive assertions, uncertainty can be introduced into the graph.

B. Inference Graph Assertions are "Abstractions"

The AMCs are in fact "classifiers". Each AMC in the hierarchy is configured by an ontology that defines classes that are the types of things in the domain of interest. In other words, the ontology contains the class definitions of the domain concepts. Class definitions are the abstract data types of the domain. Concepts are recognized by CHAMPION reasoners that have been configured to detect them. This means that for each AMC in the hierarchy there is a class definition in the governing ontology.

The purpose of each AMC is to recognize the existence of an individual of the type that belongs to its assigned class. If the individual does exist, the agent publishes the appropriate assertions.

V. The Taxonomy of CHAMPION AMCs

There are several types of AMCs in the CHAMPION system. Each AMC has the job of classifying the individuals that exist in the system. To deal with different kinds of concepts, it is necessary to define different kinds of reasoners within the AMCs. We have defined the following types of reasoning agents:

- Subsumptive
- Composite
  - Aggregate
  - Existential

We will discuss each of these in the following sections.

A. Subsumptive Reasoning Agents

Subsumption is rather straight forward. The knowledge representation language (OWL) used to implement our governing domain ontology specifically defines the predicates for subclassing and superclassing. A subsumptive agent examines the state of subscribed subgraphs and determines if the subgraph is subsumed by a higher level class defined in the ontology. Consider the following example:

A subsumptive reasoning agent would be used to recognize that an asserted Vehicle was in addition to being a Vehicle a Motorcycle as well. The reasoning agent would subscribe to individuals of type Vehicle, examine the state of that individual, and determine if the state of the individual meets the criteria for being a motorcycle. For instance, the Vehicle may have two wheels and handlebars, thus qualifying it as a Motorcycle. The reasoning agent would then publish the added assertion that the Vehicle was also a Motorcycle.

B. Composite Reasoning Agents

Composite reasoning agents are less straightforward. Unlike subsumption which is supported by explicit subclassing and superclassing predicates of standards based ontology languages, the composite reasoner examines user defined predicates to determine if the classification is valid. Subsumption only requires that a new typing assertion on an existing individual be made, not the creation of a new individual. A composite reasoner on the other hand may need to create a new named individual, not just new assertions on existing individuals.

C. Aggregation Composite Reasoning Agents

These agents must recognize when the requisite parts to an individual are present, and if so, create the new individual. An example of this kind of reasoning follows:

Continuing with the Vehicle example, a composite reasoning agent would subscribe to subgraphs that represented parts of a Motorcycle. These would be individuals of type Wheel and Handlebar. When the reasoning agent recognizes that all the requisite parts of a specific Motorcycle exist it creates a new individual and makes the appropriate object property assertions.

An important aspect of this aggregation process is the concept of making sure that the pieces are all parts of the same whole. In the CHAMPION system we refer to this notion as a "binning property." This property can be thought of as a Vehicle Identification Number (VIN) on an automobile. The VIN is a number that is used to keep track of the parts that belong to a specific automobile. It is not true that any four wheels, any engine, any fender, or any two bumpers sensed as inputs are the parts that make up an automobile. There has to be a mechanism to assure us that these parts all belong to the same car. This is the purpose of the binning property of a CHAMPION Composite Reasoning Agent, to make sure that the parts are recognized as being parts of a specific whole.

D. Existential Composite Reasoning Agents

Existential reasoning agents are very similar to aggregation reasoning agents in the fact that they have the capability to create a new individual if it is appropriate to do so. However, the aggregate reasoning agent is looking for the sum of a whole, looking to entail the existence of a thing if its necessary parts exist. An existential reasoning agent is looking to entail the existence of a thing based on evidence that it should exist. As an example of existential reasoning, if we know that a traffic ticket exists which identifies a particular license plate, we can infer that a vehicle exists. In contrast, an example of aggregation reasoning would be if we watched for vehicle parts and when we found the parts necessary to make a vehicle we could infer a vehicle exists.

The assertion that a traffic ticket exists carries little uncertainty. The inference that a vehicle exists based on the assertion of the traffic ticket carries with it a level of higher uncertainty than the existence of the traffic ticket. There could not have been a violation without the vehicle, but it may have been destroyed as a result of the violation. If we assert that it exists based on the fact that a traffic ticket refers to it, we are propagating a level of uncertainty.

VI. AMC Clockworks – Making AMCs Tick

CHAMPION AMCs comprise several components. The main component is a modified CBR mechanism. We have customized a traditional approach to CBR in order to meet the design criteria established early in our implementation.

A. *Traditional Case Based Reasoning Cycle*

A traditional CBR cycle iterates through instances of cases in a case library. As a new case is considered in traditional CBR it is compared to each of the cases in its case library. If a match is found it is considered to be a solution/match to the new case. If an exact match is not found in the case library, the closest match is modified to see if it can be made to match. If it can it is considered a solution and the modified case is added to the case library.

B. *CHAMPION's Modified Case Based Reasoning Cycle*

We chose to alter the traditional CBR cycle because the iterations through the case library to find an exact match do not fit our functional requirement to use an ***invariant form*** to characterize solutions.

The CHAMPION CBR cycle doesn't iterate through instances of cases in a case library. As a new problem case is considered it is compared to semantic expressions to see if qualifies (i.e. it belongs to the appropriate class) to be in the case library. A Description Logic (DL) reasoner is used to examine the state of the new case, if that state entails that the classification is true, the new case is added to the case library, and published to the working memory (see Figure 2).

In traditional CBR the case library is used as a repository for cases that will be iteratively compared to new input cases. This is not the purpose of the case library in our modified version of CBR. The CHAMPION system maintains the case library for the purpose of statistical analysis. The results of the statistical analysis can be used to improve the semantic expressions that define whether or not the abstractions belong in the case library.

C. *Processes of the AMCs*

The semantic expressions which define the class of objects recognized by the reasoning agents are implemented in the form of Semantic Web Rule Language (SWRL) and equivalent class expressions in OWL. The Reasoning Agents use a DL Reasoner to examine the state of the subscribed abstractions and modify the data and object properties of the abstractions.

A basic flow of the processes of an AMC:

1. Accept subscribed abstractions into local memory.
2. Acquire the requisite/relevant knowledge from contextual knowledgebases and assert into local memory.
3. Apply SWRL rules to abstractions to check and modify their state (i.e. their data and object properties).
4. Check to see if the abstraction can be classified as the targeted type of the Reasoning Agent based on equivalent class expressions in the domain ontology
5. If the DL reasoner has typed the abstraction as the targeted type, publish the abstraction to memory and add it to the case library of this agent.

The purpose of the AMCs is to process abstractions (subscribed input) and decide if it is appropriate to publish additional assertions. The additional assertions are not limited to existing individuals, meaning that the AMCs can assert new named individuals if deemed appropriate.

VII. AMC Regions

The reasoning framework arranges the AMCs in a hierarchy. The lowest levels of the hierarchy contain AMCs that subscribe to the abstractions published to the working memory by the reifiers. The AMCs of the system have a publish and subscribe relationship with working memory (see Figures 4 and 5).

When a low level AMC publishes an abstraction, a higher level AMC may be a subscriber of that type of abstraction. This is the method in which abstractions propagate up the hierarchy. As mentioned earlier, at the lowest levels in the hierarchy one expects that the abstractions contain very little uncertainty. As the AMCs are placed higher in the hierarchy the more uncertainty is likely in their output abstractions.

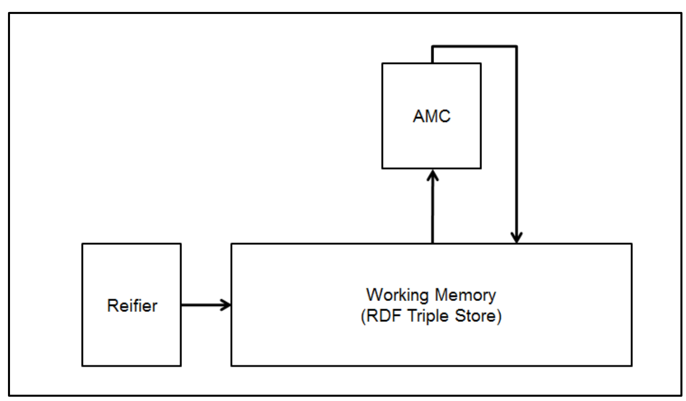

Figure 4. AMCs Publish and Subscribe to and from Memory

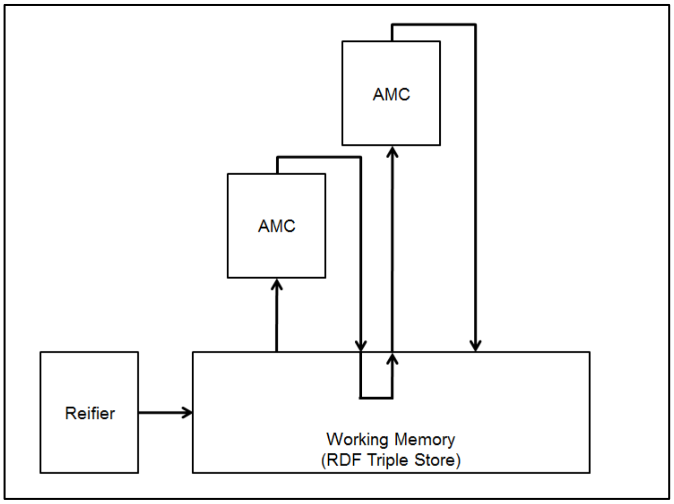

Figure 5. Abstractions passing up the AMC hierarchy

VIII. Applications

The CHAMPION reasoning framework is being applied to a variety of advanced decision making problem domains, including cyber security/counterintelligence, counterterrorism/ weapons nonproliferation, and smart grid power consumption analysis. A cybersecurity/counterintelligence application focusing on countering the insider threat is illustrative.

The insider threat refers to harmful acts that trusted individuals might carry out that may cause harm to the organization or those which benefit the individual. The insider threat is manifested when human behaviors depart from established policies, regardless of whether it results from malice or disregard for security policies. The annual e-Crime Watch Survey conducted by Carnegie-Mellon's CERT program reveals that for both the government and commercial sectors, current or former employees and contractors pose the second greatest cybersecurity threat, exceeded only by hackers; the financial impact and operating losses due to insider intrusions are increasing [18,19]. Modeling employee computer behaviors of concern using knowledge engineering methods serves as a framework to explore the insider threat. A key to the identification of an insider threat is to understand the signatures of suspicious activity and to disrupt it in its early stages. The main objective of our research is the development, validation and improvement of knowledge discovery automation tools for cyber security personnel that will significantly reduce the amount of manual analysis while simultaneously improving the quality of perceived threat indicators [20].

To create useful models, information is acquired from multiple sources including specialized reports, open literature, and subject matter experts. This information is captured via interviews with subject-matter experts (SMEs) and the development of concept maps based on domain expertise and literature analysis.

We conducted interviews of SMEs to capture information and priorities, to reveal how analysts intuitively conduct risk profiling, and to understand how they gather information about the purposes, goals and perceived risk mitigation outcomes of such activities. The information acquired is formally represented ontologically; some of the information is stored in contextual memory, and other information resides in ontologies that drive the AMCs and define the structure of the hierarchy of reasoners for this application. Figure 6 illustrates the CHAMPION system architecture within this application context.

Another interesting application for this technology is understanding nuclear proliferation. The nuclear fuel cycle is a large, complex process with many stages, dependencies, processes and signatures. In the coming year the team will use the CHAMPION framework to provide a mechanism for exploring the nuclear fuel cycle (NFC) and the logical relationships between the activities, processes, and materials involved. Working with SMEs, the team will encode the necessary knowledge into OWL to implement a proof-of-concept demonstration that will focus on a portion of the NFC. As development continues, broader coverage of the NFC will be encoded.

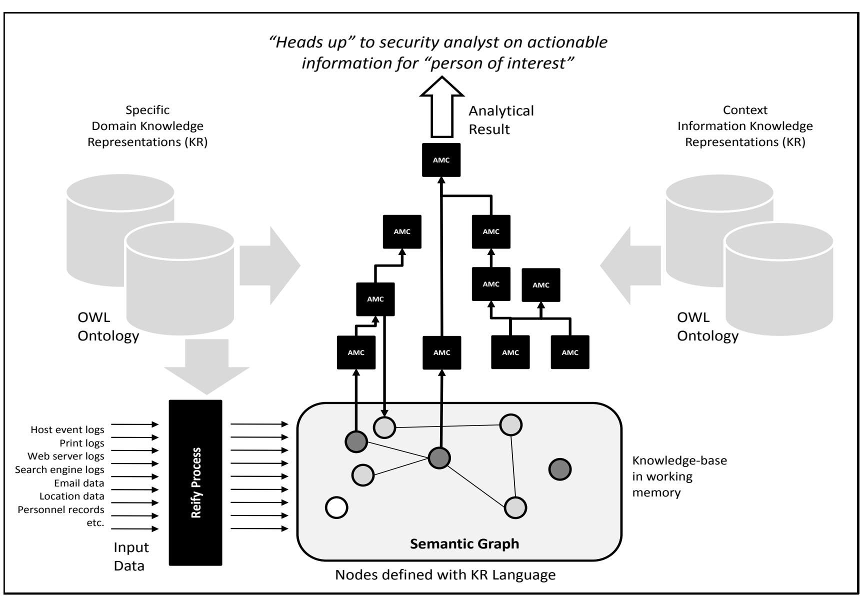

Figure 6. CHAMPION Framework in an insider threat monitoring application

IX. CONCLUSIONS

We have described a new approach to computational reasoning models that combines key aspects of belief propagation networks, semantic web, Description Logics, and Case Based Reasoning to yield a system best characterized as a memory-prediction framework. This framework is functionally modeled after an interpretation of how the neocortex performs pattern recognition. It is implemented as a hierarchy of reasoning agents that retain certain critical functional requirements that produce a domain-independent model that may be applied to a variety of decision making problems.

Earlier in this paper, we compared several extant approaches to problems in AI and noted the drawbacks of using rational decision making models to characterize human performance, such as represented in typical BN models that rely on probability theory constructs. Similar issues apply to models that apply other forms of probabilistic models such as subjective expected utility theory. Famous research programs conducted by Kahneman and Tversky [e.g., 21] demonstrate that human decision making is not rational and is rather characterized by the use of heuristics (or influenced by cognitive biases) that do not yield optimal decisions. The use of heuristics—and what has been described by Kahneman [22] as "system 1 cognitive processes" – exploiting intuition and experience rather than procedural knowledge – is sometimes cited as a critical survival mechanism that accounts for expert decision making by firefighters and other highly experienced individuals who do not have time to systematically calculate and compare outcomes of alternative responses [23]. A conceptual model that reflects this view is the "Recognition-Primed Decision Making Model" (RPDM) offered by Gary Klein and collaborators [24]. In this regard, the basic structure of the CHAMPION reasoning framework, rooted in the notion of the memory-prediction system, is very compatible with this view of expert decision making. Indeed, the CHAMPION framework represents one method of implementing an operational version of a RPDM model. It is our hope that such a model, fortified by recent computational methods adopted from semantic Web technologies, will provide a major advancement in realizing the vision for joint cognitive systems for decision support.

ACKNOWLEDGMENT

The authors wish to thank the CHAMPION team - Shawn Hampton, Samantha Curtis, Alex Gibson, Michael Henry, Dan Johnson, Gary Kiebel, Peter Neorr, Patrick Paulson, Yekaterina Pomiak, and John Schweighardt; and the Information and Infrastructure Integrity Initiative at Pacific Northwest National Laboratory which supported this work. The Pacific Northwest National Laboratory is operated by Battelle for the U.S. Department of Energy under Contract DEAC06- 76RL01830. PNNL Information Release Number PNNL-SA- 82834.

REFERENCES

[1] D. D. Woods, "Cognitive technologies: the design of joint human-machine cognitive systems," AI Magazine, vol. 6, pp. 86-92, 1985.

[2] D. D. Woods, Joint Cognitive Systems: Patterns in Cognitive Sytems Engineering. Boca Raton, FL: Taylor & Francis, 2006.

[3] Institute of Medicine Forum on Neuroscience and Nervous System Disorders. From Molecules to Minds: Challenges for the 21st Century. Washington, DC: National Academy of Sciences, 2008.

[4] B.G. Buchanan, "A (very) brief history of artificial intelligence," AI Magazine, vol. 26, pp. 53–60, 2005.

[5] N.J. Nilsson, Artificial Intellugence: A New Synthesis. San Francisco, CA: Morgan Kaufmann Publishers, Inc., 1998.

[6] R. Chrisley, ed. Artificial Intelligence: Critical Concepts, vols. 1-4. London: Routledge, Taylor & Francis Group, 2000.

[7] D.J.C. MacKay. Information Theory, Inference, and Learning Algorithms. Cambridge: Cambridge University Press, 2003.

[8] P. Smolensky, "On the treatment of connectionism," Behavioral and Brain Sciences, vol. 11, pp. 1-23, 1988.

[9] T.G. Dietterich, "Machine Learning," Annual Reviews in Computer Science, vol. 4, pp. 255-306, 1990.

[10] M. Jones and B.C. Love, "Bayesian fundamentalism or enlightenment? On the explanatory status and theoretical contributions of Bayesian models of cognition," Behavioral and Brain Sciences (in press).

[11] R. Lopez De Mantaras, et al., "Retrieval, reuse, revision and retention in case-based reasoning," The Knowledge Engineering Review, vol. 20, pp. 215-240, 2005.

[12] I. Watson and F. Marir, "Case-based reasoning: a review," Knowledge Engineering Review, vol. 9, pp. 355-381, 1994.

[13] A. Aamodt and E. Plaza, "case-based reasoning: foundational issues, methodological variations, and system approaches," AI Communications, vol. 7, pp. 39-59, 1994.

[14] A. Nouri and H. Nikmehr, "Hierarchical bayesian reservoir memory," Proceedings of the 14th International CSI Computer Conference (CSICC'09), pp. 582-587, 2009.

[15] J. Hawkins and S. Blakeslee. On Intelligence. New York: Henry Holt and Company, 2004.

[16] Buneman, P., S. Khanna, and W.C. Tan. 2001. Why and where: A characterization of data provenance. International Conference on Database Theory (ICDT), 316-330.

[17] Simmhan, Y.L., B. Plale, and D. Gannon. 2005. A survey of data provenance in e-Science. ACM SIGMOD Record, 34(3), Sept. 2005.

[18] CSO Magazine, U.S. Secret Service, Software Engineering Institute, CERT Program at Carnegie Mellon University and Deloitte. 2010 CyberSecurity watch survey - survey results.

[19] M. Keeney, et al. Insider Threat Study: Computer System Sabotage in Critical Infrastructure Sectors. U.S. Secret Service and Carnegie-Mellon University, Software Engineering Institute, CERT Coordination Center. 2005.

[20] F. L. Greitzer and R. E. Hohimer, "Modeling human behavior to anticipate insider attacks," Journal of Strategic Security, vol. 4, pp. 25-48, 2011. doi:10.5038/1944-0472.4.2.2.

[21] D. Kahneman and A. Tversky, "On the psychology of prediction," Psychological Review, vol. 80, pp. 237-251, 1973.

[22] D. Kahneman, "A perspective on judgement and choice: mapping bounded rationality," American Psychologist, vol. 58, pp. 697-720.

[23] G. Klein. Streetlights and Shadows: Searching for the Keys to Adaptive Decision Making. Cambridge, MIT Press, 2009.

[24] G.A. Klein, "A recognition primed decision (RPD) model of rapid decision making," in GA Klein, J Orasanu, R Calderwood and CE Zsambok, eds. Decision Making in Action: Models and Methods. Norwood, NJ: Ablex, pp. 138-147, 1993.