Proceedings of the 2009 IEEE Symposium on Computational Intelligence
in Security and Defense Applications (CISDA 2009)

Goal-Driven Semi-Automated Generation of Semantic Models

Anthony P. Stirtzinger, *Member, IEEE*, Craig S. Anken, and Bruce McQueary, *Member, IEEE*

Abstract—**The approach taken with OGEP is to parse relevant domain data in the form of unstructured content (or corpus) and use that knowledge to generate and/or evolve an existing ontology. OGEP creates a constant conversation between the corpus parser and a reasoning mechanism (corpus reasoner) that continually formulates potential ontology modifications in the form of hypotheses. These hypotheses are weighted towards contextual relevancy and further reasoned over to provide a confidence measure for use in deciding new assertions to the ontology. The new assertions generated from the corpus reasoner can either be automatically asserted based on confidence measure, or can be asserted by OGEP interacting with a user for final approval. This paper describes the OGEP technology in the context of the architectural components and identifies a potential technology transition path to Scott AFB's Tanker Airlift Control Center (TACC), which serves as the Air Operations Center (AOC) for the Air Mobility Command (AMC).**

I. Introduction

As Department of Defense (DoD) technology platforms have evolved away from monolithic systems and applications towards net-centric Service Oriented Architecture (SOA) implementations, the issue of semantic interoperability, or interoperability based on the aligning the meaning or intent of the information exchange, has increasingly become a critical element in successfully deploying solutions. Prior to the highly distributed and service oriented systems that are now being developed for the DoD, semantic interoperability was not much of an issue because the systems and applications each contained "implied" semantics. Since data integration was limited to agreed upon message formats, application programming interfaces (e.g. in the form of shared libraries), and database schemas, it was very clear what data was exchanged, to whom the data was exchanged with, and how the data would be used. The boundaries of data ownership and data usage were very closely aligned and all stakeholders understood the how, when, and where of data usage. This combination of implied model and implied ownership insured that the data was used correctly but also limited the usefulness of data in terms of leveraging it with other sources of information.

C. Anken, is with the Air Force Research Laboratory, 525 Brooks Road, Rome, NY, USA 13441
A. Stirtzinger is with Securboration Corporation, 1050 W. Nasa Blvd., Melbourne, FL 32901 USA (phone: 321-409-5252; e-mail: tstirtzinger@securboration.com)
B. McQueary is with Securboration Corporation, 1050 W. Nasa Blvd., Melbourne, FL 32901 USA (phone: 321-409-5252; e-mail: bmcqueary@securboration.com)

Net-centricity has exposed this deficiency as systems and applications have become decomposed into a web of smaller specialized services integrated together to share information towards accomplishing common goals. As functionality is distributed and services are used that can contribute information in a variety of contexts, an implied static model no longer exists. Data and services are being used in an array of contexts requiring knowledge about how a single piece of data is used across each of those contexts. Simple syntactic integration approaches, successful in transforming, translating, and aligning the structure or content of the data, but ignorant of the semantic meaning of the data, are no longer acceptable. Today's SOA environments require an approach that includes a semantic model to maintain the integrity of the data, consistent with the owner's intent and usable by a variety of services whether computing, analyzing, or visualizing. The models used to declare this semantic context can be created using ontology languages.

Extending the notion of data or object models, ontologies can provide rich semantic definition to both the meta-data and instance data of domain knowledge. While the value and potential of a well structured ontology is rarely debatable, deciphering data, especially in unstructured document form, into this domain related knowledge is a difficult task. Approaches leveraging semantic models come burdened with the overhead process for the creation of those models. The need for specialized elicitation, modeling, and ontology development skills is still required and in limited supply. An additional complexity with the use of semantic models is the fact that most ontology development efforts end at the creation of the ontology, publishing the ontology for users and applications to "use as needed". Ontology development in this manner produces a static snapshot of the domain or application space and does not account for ontology evolution in the form of tuning, extending, refinement, or practical use.

The Ontology Generation and Evolution Processor (OGEP) mitigates the amount of manual labor involved with ontology maintenance by bringing together multiple facets of semantic knowledge in order to enhance and evolve an existing ontology. The goal of OGEP is to improve the automation of the ontology development process, only requesting user input when un-resolvable ambiguous situations occur. The ability to grow ontologies allows OGEP to adjust to new inputs and stay relevant to its domain as the data space broadens. Thus OGEP addresses a key issue which limits automated ontology generation: the ability to understand the meaning of objects (i.e., entities, events) and the relationships these objects have within the domain of interest.

II. Automated Ontology Generation

The classic definition of ontology is Gruber's "A specification of a conceptualization" [1]. In practical terms, an ontology extends the concept of a hierarchy (taxonomy) with rich semantic relationships among classes and types. A common representation used to model ontological concepts is the W3C Object Web Language (OWL), an XML-based ontology language that can represent concepts (called classes), data types, instances (called individuals), relationships (called properties), and restrictions. Restrictions include assertions about what can be inferred if a class is of a particular type (necessary restrictions), and assertions about how the properties of an entity can identify that entity (class or individual) to be of a specific class type (necessary and sufficient restrictions). OWL has three increasingly rich sub-languages: Lite, DL (Description Logic), and Full. OWL DL is most often used in solutions that employ reasoning. Previous research explains the different reasoning techniques, when they are applicable and general rationale for using ontologies for semantic reasoning [2].

Automated and semi-automated ontology generation is not a new concept. Ding [3] has surveyed research efforts addressing the problem of ontology generation. While these attempts are quite different in implementation, they all share two basic concepts. First, each of the efforts required a seeding of vocabulary or terms (either explicit or implicit within the corpus) as a starting point for the ontology generation. The corpus in each case was parsed using some free text parsing mechanism to extract parts of speech (POS) entities and match them to the seeded terms via natural language algorithms. Second, semantic relationships between terms were also discovered. These relationships are discovered by spatial approximation between the terms. Spatial approximation is where the efforts vary the most, each using different specialized algorithms to discover the relationships during the parsing process. The shortcoming of the efforts, as documented by Ding, is the failure to find appropriate and efficient ways to detect or identify relations either automatically or semi-automatically. Ding also cautioned about the problem of mapping to an existing ontology.

An approach taken later than Ding's report, thus not documented by him, is the mapping of the Suggested Upper Merged Ontology (SUMO) to WordNet. This work addresses Ding's concern about the problem of mapping to existing ontology by using SUMO along with the concept of a Mid-Level Ontology (MILO) as a bridge for mapping other ontologies; SUMO containing the abstract ontology concepts and MILO containing the domain specific concepts. Using this upper and midlevel ontology, a mechanism is available to relate or map other ontologies at the required level.

However, still not addressed in any of the works mentioned above is a solution for Ding's primary concern of automatically or semi-automatically discovering semantic relationships. OGEP addresses this deficiency by using a goal-driven approach to aid in the discovery of semantic relationships. Foundation domain concepts and corresponding goals are seeded in an ontology. This provides the core to focus discovery needed to evolve the ontology. The infrastructure ontology called Semantic Grounding Mechanism (SGM) [4] aids in the process by providing the structure for expressing goals as domain independent semantic relationships among domain entities.

III. OGEP OVERVIEW

OGEP attempts to mimic the process that humans use to read and reason about the meaning of a corpus relevant to an end goal. The corpus parser is used to read the corpus while the corpus reasoner is used to continually formulate potential ontology modifications in the form of hypotheses which are compared to the foundation concepts and goals. These hypotheses are weighted towards contextual relevancy and further reasoned over to provide a confidence measure for use in deciding new assertions to the ontology. The new assertions generated from the corpus reasoner can either be automatically asserted based on confidence measure, or can be asserted by OGEP interacting with a user for final approval. Fig. 1 shows the process flow through the OGEP components.

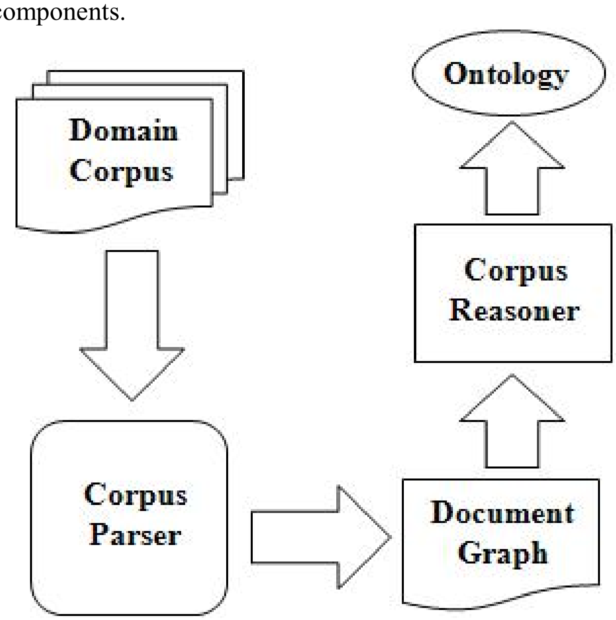

Fig. 1. OGEP Components

The OGEP system starts with two assumptions: First, the goals of the semantic model are defined in ontology language using the SGM ontology patterns; and second, a corpus of information exists that can confirm the semantic model, contradict the semantic model, or extend the semantic model. OGEP's Corpus Parser consumes the content of the corpus and creates one or more Document Graphs (DG) [5], [6]. The DGs provide a semantic graph representation of the content. The Corpus Parser annotates both domain independent semantic relationships along with domain semantics by referencing the existing ontological concepts to interpret the content during graph generation. Each node and relationship in the DG is attributed with a numeric value representing the relevance of the node/relationship to the existing ontology. The DGs are then passed to the Corpus Reasoner for processing. It is the job of the Corpus Reasoner to evaluate the concepts within the DGs to compute a confidence level for each concept as to the impact on the ontology. Concepts that are determined to be "matches" increase the confidence level of the ontological concept. Concepts that contradict ontological concepts are converted to hypotheses, weighted for confidence, and considered for ontological modifications. Extensions to the existing ontological concepts are also handled through hypothesis processing. As previously noted, hypotheses can transition to assertions either automatically or through user interaction depending on the configuration of the OGEP system.

IV. CORPUS REASONER

The Corpus Reasoner is the core of the OGEP system; its purpose is to interact with both the Corpus Parser and the existing ontology to synthesize the content of each into an assimilated, fused resulting ontology. All of the lexical analysis components within the Corpus Reasoner, along with the ontological assertions are available to the Corpus Parser so the iterative interaction can occur when processing domain content. Corpus Reasoner is comprised of three main components: Semantic Processing Engine (SPE); Semantic Memory Controller (SMC); and Hypothesis Reasoner. Fig. 2 shows the components of the Corpus Reasoner.

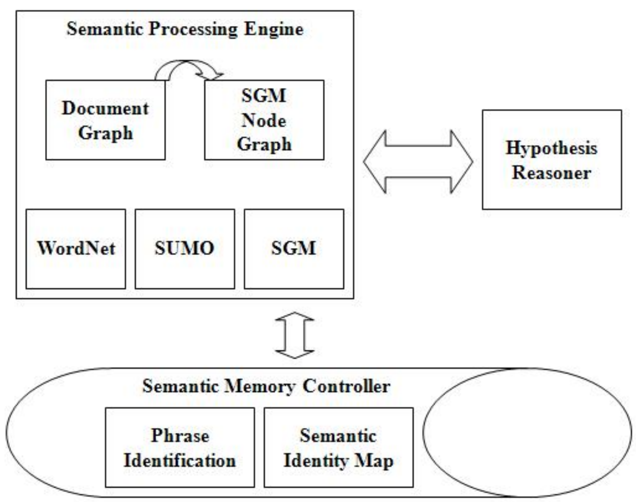

Fig. 2. Corpus Reasoner Components

A. Semantic Processing Engine

The Semantic Processing Engine is the primary processing element of the Corpus Reasoner. It is comprised of a series of semantic processing components that follow a chain of responsibility in analyzing the DGs generated by the Corpus Parser. As a DG is passed through the chain of responsibility, each processing component is given the opportunity to identify, modify, or remove semantic entities and relationships. This cascading responsibility allows the Semantic Processing Engine to build upon itself and extend the semantic knowledge of the system. The semantic processing components make use of WordNet, SUMO, and the SGM; descriptions of each follow.

WordNet [7] is a large lexical database of English, developed under the direction of George A. Miller. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations. Each word in WordNet is comprised of a series of one or more Senses. A Sense is a set of synonyms, a definition, and information about related Senses. Senses in WordNet have many associations with other Senses. One such association is the concept of a hypernym. WordNet provides OGEP with a rich set of nouns, verbs and adjectives, correlated with synonyms, hyponyms, and hypernyms. This allows OGEP to reason about the context of a word and also a classification of the word. WordNet's support of automatic text analysis allows us to reason about the context of the word usage. The grouped sets of cognitive synonyms are referred to as synsets. Each synset expresses a distinct concept. These synsets are interlinked by means of conceptual-semantic and lexical relations. Each word in WordNet is comprised of a series of one or more Senses. A Sense represents a set of synonyms, a definition and a set of relationships to other Senses. WordNet allows us to reason about the context of the word usage through the use of the Senses it offers. The parts of speech attached to the DGs as a result of the Corpus Parser can be "looked" up individually using WordNet to find which Senses the word may match. Those that are nouns, verbs or adjectives will most likely result in one or more Senses. Anything that results in more than a single Sense will have further analysis to disambiguate the word's usage. WordNet also allows us to discover a domain for each Sense. Using this information, we can then determine the corresponding SUMO classes associated with each Sense we obtained from WordNet. Gathering up the Sense synonyms and SUMO class names, we can build a hypothesis for reasoning against the SGM ontology.

SUMO [8] is an upper level ontology that classifies worldly entities. It is a standard ontology that promotes data interoperability, information search and retrieval, automated inferencing and natural language processing. Currently it contains over 4,000 assertions and 1,000 concepts and has been compared to a dictionary or glossary but with greater detail. It is considered to be a foundation ontology, which means that it describes general concepts across many domains. As foundation ontology, it is limited to concepts that are Meta, generic, abstract and philosophical, and therefore general enough to address a broad range of domain areas. SUMO is also useful because each of the nouns, verbs and most of the adverbs found in WordNet are mapped to a constrained set of SUMO elements. OGEP leverages the SUMO classifications, in concert with the WordNet classifications to both identify new terms to the ontology, and also to reason about the properties of an existing term in the ontology. SUMO provides a high-level ontology of the WordNet data space. Each of the nouns, verbs, and most of the adverbs found in WordNet are mapped to a constrained set of SUMO elements. Since so many WordNet Senses are mapped to such a small set of SUMO elements, generalizations can be easier to make. For example, SUMO has the notion of a SentientAgent that includes all humans and organizations. This type of mapping does not exist directly in WordNet and would require several different mappings to accomplish this level of generality.

Finally, SGM is a mechanism for defining semantic patterns about a domain using an ontology language. As opposed to pure class hierarchy and relationship reasoning that is available via SUMO and to some extent from WordNet, the SGM provides a set of structured semantic concepts that are related to an ontological entity to further describe the context of that entity. The SGM provides an infrastructure ontology that is entirely focused on the definition of semantic relationships. Fig. 3 SGM Semantic pattern shows the basic ontological pattern that SGM uses to describe a semantic concept.

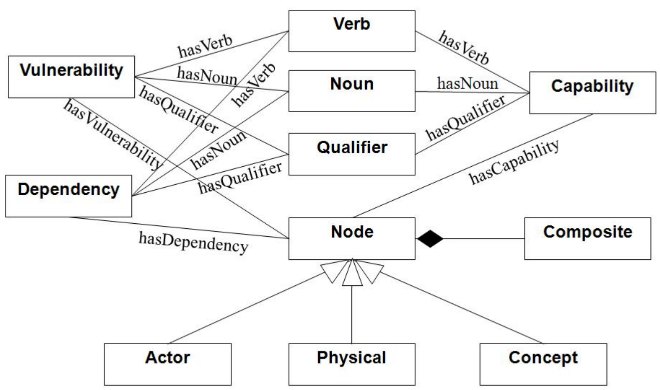

Fig. 3. SGM Semantic Pattern

Each concept is classified as a "Node" type (physical, actor, or conceptual). SGM has been merged with SUMO to form a semantic model that provides the ability to derive semantic meaning from a wide range of categories. This is accomplished by linking the SGM Nodes to SUMO classes for "real world" classification. In addition to the classification of Nodes, SGM provides for a set of properties to be defined that describe each semantic concept. These properties are representd as relationships to "dependency", "capability", and "vulnerability" classes. Each dependency, capability, and vulnerability consists of a relationship to another SGM Node along with a verb-noun-qualifier tuple associated to that relationship. The tuple provides linkages to WordNet for verb sense disambiguation, synonym matching, hyponym traversal, and other lexical analysis used to compare the DGs created by the Corpus Parser to the SGM Nodes resident in the existing semantic model. As semantic entities are discovered within the corpus, they are added to the SGM where semantic relationships are identified, leading to further discovery of semantic entities.

The process of intelligently selecting which discoveries of the corpus are valid to be committed to the semantic model as changes is handled by the Hypothesis Reasoner. The SPE using WordNet, SUMO, and SGM convert DGs into potential semantic sub-graphs in the form of SGM Nodes and relationships. These sub-graphs are then handed to the Hypothesis Reasoner for processing.

B. Hypothesis Reasoner

The Hypothesis Reasoner evaluates each of the semantic sub-graphs created by the SPE to determine their weight of applicability to evolving the existing semantic model. Hypothesis Reasoner uses the verb-noun-qualifier tuples resident within the semantic sub-graph to calculate a probabilistic value or weighted confidence value as to how closely the semantic sub-graphs align with the current SGM Node graphs in the semantic model. Direct matches of Nodes and tuples are ranked highest, matches using synonym or hyponym relationships are ranked lower, matches discovered through indirect relationships are ranked lower. Markov models are being implemented to represent hypothesis as related weighted semantic sub-grpahs with attached probabilistic values. The probabilistic values provide the system or user with a ranked set of hypotheses suggesting how to evolve the current semantic model. Initial criteria for the measurement of the hypotheses to be calculated using the Markov models has been selected and tested to validate their reasonability.

C. Semantic Memory Controller

Remembering decisions that have been made, understanding why they were made, and using that knowledge in future decision making is one crucial piece of learning. It allows an individual to encounter a situation that might lack information but still make "reasonable" deductions. The Semantic Memory Controller (SMC) is responsible for remembering all user and algorithmic decisions made by the system. This allows the system to deduce a solution that it has previously encountered where only ambiguous elements are present. A particular solution may be putting added weight to a particular Sense, or by outright choosing a Sense to continue analyzing with. As OGEP progresses through document analysis, semantic elements and the relationships between those elements are identified. These entities and their relationships provide an arsenal of data for axiomatic mining. The SMC evaluates semantic knowledge as it is added to the system and seeks commonality and generalizations through the use of patterns. For example, during processing we detect that certain aircraft types tend to always run late. The SMC would be used to realize those situations occur only at specific airfields, narrowing down the solution space and potentially identifying a deficiency.

The SMC utilizes two sub-components, Phrase Identification and the Semantic Map. The Phrase Identification Module remembers previously identified phrases such as "Osama bin Laden" or "Dar es Salaam". In basic terms, it is a dictionary designed for use with multi-word phrases. The Semantic Map identifies previously mapped words and phrases which have been mapped to another entity, whether that entity is another word or an ontological entity. An example of this would be mapping the phrase "Muhammed Atef" to the ontology element Terrorist. We would also want to make a correlation between the phrase "Abu Hafs" and the phrase "Muhammed Atef" since "Abu Hafs" is an alias for "Muhammed Atef".

V. RESULTS

Initially the OGEP process was implemented from end-to-end with manual intervention throughout the steps. As research has progressed, the manual steps have been incrementally replaced with automated mechanisms. The automation in OGEP is not intended to "replace" the need for domain experts, subject matter experts, or knowledge engineers. However, the automation is intended to aid these experts and engineers, increase their productivity, and ultimately produce a better ontology product. A number of key technologies have been implemented to promote the automated derivation of semantic relationships and evolution. As mentioned, the combination of SUMO, WordNet, SGM and the Corpus Reasoner provide a robust set of tools for gaining semantic knowledge. Results to date include development of a WordNet interface for high performance access and caching of the WordNet lexical database. A SUMO to WordNet bridge has been developed that brings together the general categorization of SUMO and the more specific lexical definitions of WordNet. Additionally, an OGEP prototype has been developed that is capable of identifying singular "Sense" phrases that allow user resolution of conflicting or ambiguous definitions. In further efforts to phase out user level dependencies, different NLP parsers were tested. The goal of this testing was to determine how much the quality of the NLP results beyond simple POS findings affected the ability to infer the semantic context of words. This testing revealed that the "training" required for the NLP parsers to provide better results beyond POS was not an acceptable prerequisite for automated ontology generation. As an alternative, an approach has been adopted to use Document Graphs. As discussed previously, the DG's add a level of semantic relationships on top of the POS information, but do not require domain specific training beyond what is already captured in the SGM. The DG's also proved to be very complimentary to the SGM Node Graphs and make it possible to form the basis of the OGEP hypothesis analysis. As an example, a small ontology was developed containing concepts of an "Arms Trade". Test corpus were created and processed through the DG parser. Fig. 4 shows a small segment of the related nodes produced.

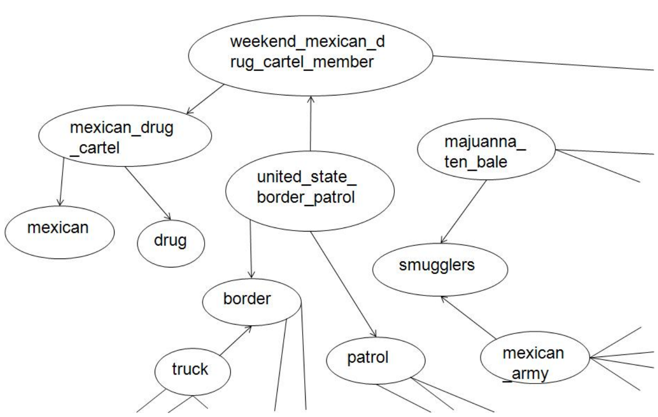

Fig. 4. Sample DG Segment

Using the ontology in the DG can be specialized by changing generic "related_to" relationships to more specific relationships. For example the existing DG has "weekend_mexican_drug_cartel_member" related to "united_state_border_patrol". Corpus Reasoner is used to recognize the verb and replace this relationship with the actual action which is "arrest". Similarly the "related_to" relation between "truck" and "border" can be made more specific by "continue toward" relationship. Continuing to refine "drug" and "majuanna_ten_bale", they may be related with each other and then related to "smugglers". This is due in part to the notion of an "arms trade" being already defined in the ontology and the fact that both "drug" and "arms" can play the role of trade products with respect to "smugglers" - smugglers may be buying drug or using drug as a currency for trade.

In Fig. 5 the bold segment contains relations added from the segments "arm buyers-> buy-> arms" in the ontology contains relations added from the segment "arm sellers -> access -> arms".

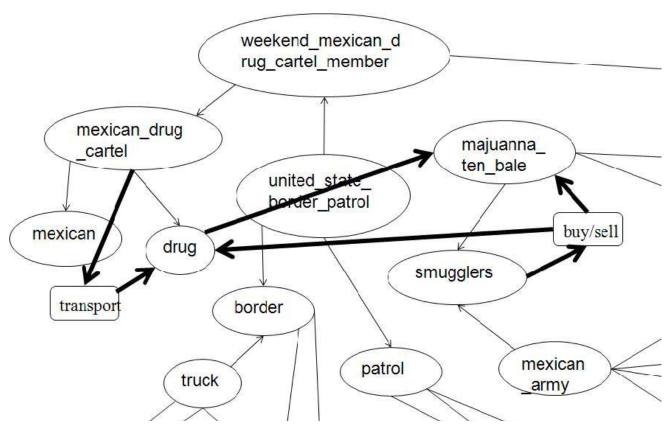

Fig. 5. New Relationships

Finally, many unused relationship can be removed if those relationship: 1) are not connecting to many other nodes; 2) state the obvious or 3) have been replaced by the relationships in the ontology. For example: several "truck -> related to -> truck" can be removed because it states the obvious; as does "large wooden crate cargo-> related_to -> large wooden crate"; and some "related_to" relationships can be deleted because we have already added more specific relationships from the ontology. Those relationships have noted by "x" symbols in Fig. 6.

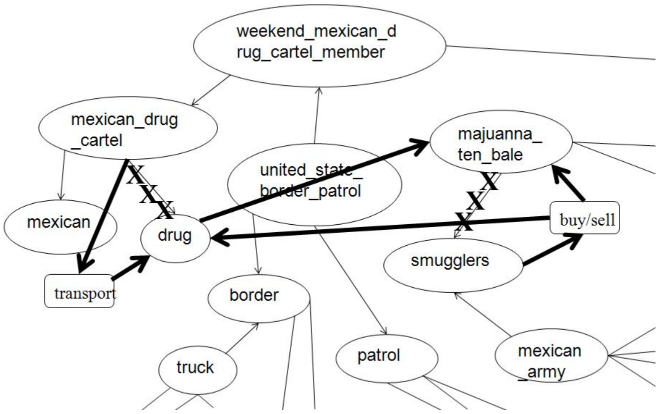

Fig. 6. Deleted Relationships

Work is continuing in the development of the Hypothesis Reasoner, along with additional Corpus Reasoner capabilities such as: discovery of new nodes and discovery of deleted nodes. Finally, implementation of an ontology evaluation mechanism has also been initiated.

VI. OGEP TRANSITION

An initial target customer for possible transition of OGEP is Scott AFB's Tanker Airlift Control Center (TACC). Scott's TACC serves as the Air Operations Center (AOC) for AMC. As such, TACC is responsible for the monitor, assess, plan and execute (MAPE) phases of airlift missions in order to meet requests for moving military equipment and personnel associated with pop-up contingencies, time-critical changes, and preplanned missions. There are numerous complicated, interrelated factors across MAPE phases. Problems or changes with any number of the factors will result in a delay or deviation in the planned mission. Problems from changes or deviations are amplified by a lack of an integrated environment to provide situational awareness to AMC operators. Current Mobility Air Force Command and Control (MAF C2) advanced technology demonstration (ATD) programs, have focused on enhancing cognitive aspects of situational awareness. The goal of MAF C2 is to provide AMC operators global visibility into their airlift fleet allowing them to quickly recognize the impacts of changes during the mission [9]. A challenge within this process has been the extremely difficulty in translating knowledge contained within Diplomatic Clearances (DIPS), Flight Information Publications (FLIPS), and crew regulations into a data model that can be exploited by automated tools. A primary contributor to this challenge is the unstructured format used to define and document the DIP and FLIP knowledge. Manual and error prone mechanisms are required to 1) interpret the DIP and FLIP regulations; 2) apply the regulations to current missions; and 3) recognize impacts to the missions when new regulations are posted. For example, current approaches require a user to review and manually extract information from DIPS and put it into an Excel spreadsheet before that knowledge can be used within an automated environment. Any change to FLIPS or DIPS regulations requires a tedious, labor intensive reentry process, to transfer the knowledge from unstructured format to machine readable form. Additionally, the key concepts that relate DIPS, FLIPS, and crew regulations (the critical semantic relationships between these regulations) are implicit and require experienced users to make the associations. DIPS, FLIPS, and crew regulations are tightly coupled and changes in any one can impact the others and negatively affect cargo capacity flow. While current MAF C2 applications are capable of depicting this knowledge (e.g. DIPS Warnings for a mission based on airspace time), advancements are needed to integrate and semantically structure the underlying data to understand all the consequences of changes or deviations on Airspace, Aircrew, Ground, Aircraft, and Airfield requirements.

OGEP can fulfill a critical need in bringing full situational awareness to the AMC operator. The problem OGEP solves in this context is not only text extraction, but also recognizing the delta changes within DIPS, FLIPS, and crew regulations and updating the semantic model to reflect the changes in a semi-automatic or automatic fashion, alleviating the problem of manually understanding deltas, and reentering data when there is a change. OGEP also allows for semantic model learning so that the rationale and mechanisms performed by "man-in-the-loop" modification operations will be intelligently stored and "learned" for automation of these steps in future model updates. OGEP technology can transform the unstructured DIPS, FLIPS, and crew information into rich semantic ontological constructs that encode knowledge about their complex interrelationships. The resulting underlying semantic model that can be applied to MAF C2 applications will contain knowledge from the unstructured sources related to knowledge currently contained in those applications.

VII. CONCLUSIONS

Our initial research and testing of OGEP shows that semi-automated ontology generated is achievable if a seeded set of assertions pre-exist to help guide the generation. To transition OGEP towards automated generation and evolution work is currently focused on creating a valid mathematical basis for the parsing using Markov diagrams representing the discovered hypotheses. Further refinement and integration of this algorithm for parsing text and creating probabilistic Bayesian Knowledge Bases (BKB) [10], [11] is ongoing.

REFERENCES

[1] Gruber, T., "A translation approach to portable ontologies, knowledge acquisition," 5(2):199-220, 1993.

[2] T. A. Breeden, A. P. Stirtzinger, "Ontology-driven reasoning and prediction", IC-AI 2007.

[3] Ding, Ying, Foo, and Schubert, "Ontology Research and Development Part 1 – A Review of Ontology Generation," SAGE Publications, Available: http://jis.sagepub.com/cgi/content/abstract/28/2/123.

[4] S. A. Stirtzinger; A. P. Stirtzinger, "Information discovery – NLP parsing using semantic ontological patterns," in *Proc. 2008 ICAI,* pp. 636-643.

[5] E. J. Santos, A. A. Mohamed, and Q. Zhao, "Automatic evaluation of summaries using document graphs," *in Proc. 2004*, Text Summarization Branches Out, ACL-04 Workshop, Spain, pp. 66-73.

[6] E. J. Santos Jr., Q. Zhao, H. Nguyen, and H. Wang, “Impacts of user modeling on personalization of information retrieval: An evaluation with human intelligence analysts,” presented at Fourth Workshop on the Evaluation of Adaptive Systems, Edinburgh, UK, July 24 - 30, 2005 pp. 27-36.

[7] http://en.wikipedia.org/wiki/WordNet

[8] I. Niles and A. Pease, “Linking lexicons and ontologies: Mapping WordNet to the suggested upper merged ontology,” in *Proc. 2003 International Conference on Information and Knowledge Engineering*, Las Vegas, Nevada, 2003. Available: http://citeseer.ist.psu.edu/niles03linking.html

[9] Mobility Air Forces Command and Control Functional Needs Analysis, 2006.

[10] E. J. Santos, A. Hussein, “Comparing case-based bayesian network and recursive bayesian multi-net classifiers,” *in Proc.* IC-AI 2004, pp 627-633.

[11] E. J. Santos, H. T. Dinh, “Consistency of test cases in validation of bayesian knowledge-bases,” *in Proc.* ICTAI 2004, pp 468-475.