Graphical Methods for Real-Time Fusion and Estimation with Soft Message Data

Kedar Sambhoos
CUBRC
Buffalo, New York, USA
sambhoos@cubrc.org

James Llinas
Center for Multisource Information Fusion
State University of New York at Buffalo
Buffalo, New York, USA
llinas@eng.buffalo.edu

Eric Little
Center for Ontology & Interdisciplinary Studies
D'Youville College
Buffalo, New York, USA
little@dyc.edu

***Abstract* – Fusion of observational data acquired by human observers and couched in linguistic form is a modern-day challenge for the fusion community. This paper describes a basic research effort examining various strategies for associating and exploiting such data for intelligence analysis purposes. An overall approach is described that involves Latent Semantic Analysis, Inexact Graph Matching, formal ontology development, and Social Network Analyses. Not all the methods have yet been employed but the exploitation of the developed ontology and graphical techniques have been implemented in a working prototype and preliminary results have shown promise. Planned future research will complete the implementation of the methods described herein and add yet further enhancements.**

Keywords: Graphical methods, linguistic data, message processing, text extraction, graph matching

1. Introduction

Traditional fusion systems have been effectively developed for applications where the observational data have come from the typical array of Intelligence, Surveillance, and Reconnaissance (ISR) calibrated electronic sensor systems. In designing the fusion process the fusion system designers typically take advantage of the well-known error characteristics of these sensors, as well as other well-specified characteristics. Modern conflict however has shown that human-based observation, reported in unconstrained natural language, has become an important and sometimes dominant observational data source, recently labeled as "soft" data to distinguish it from the "hard" data from electronic sensors (analogous to the "fuzzy" and "crisp" distinctions). Often, the linguistically-couched observational data become framed as messages cast in digitized text, resulting from an audio-to-text conversion. The "strictly-soft" data fusion problem then becomes one of designing a fusion process capable of combining the observational data from multiple digitized textual data streams. While there has been some research on the approach to data fusion in such contexts (e.g., [1] [2]), soft data fusion has not been a central research topic in the fusion community. There is the extended problem of hard plus soft data fusion when textual data must be combined with those from the typical ISR sensors; this is a yet more complex challenge and is being addressed by some scientists in the fusion community.

The project reported on herein is a "strictly-soft" fusion basic research project funded by the U.S. Army Research Laboratory, called the "Soft Target Exploitation and Fusion (STEF)", where soft is used in two ways—signifying the objects of interest—"high value individuals" such as leaders of terrorist groups—and signifying the softness of the message-based textual data. A particular challenge we have undertaken is to try and design a real-time, dynamic fusion processing approach; so that alerts and threats can be estimated as the messages are arriving, enabling the realization of "actionable intelligence". This project is the first phase of a planned multi-phased approach to this difficult application area, and so does not take on all of the complexities inherent in this type of problem.

2. Source Characterization

Among the first things a fusion process designer must do is to characterize the nature of the input sources and the data they provide, an activity usually labeled "source characterization". One of the primary aspects of this process is the analysis and specification of the error characteristics for each data operation prior to the data entering the Common Referencing or Alignment function usually considered the first function in the fusion functional chain of Common Referencing—Data Association—State Estimation. The cumulative errors must clearly be known at some level of specificity to assure that the fusion functions can employ that knowledge and not be susceptible to significant biases or other errors. For a human observer reporting observations orally, with the audio being converted to text, a representative error-chain is shown below:

Note that if the observations are the result of interrogation, or coming from third parties (e.g., cooperative civilians), additional errors can be present. Modeling these error processes and their parametric dependencies on observational context and other factors (as typically done for radars and other electronic ISR sensors) is an extremely difficult task. We have studied these issues but for prototype development have ignored these errors; we use the simulated message text as correct observables.

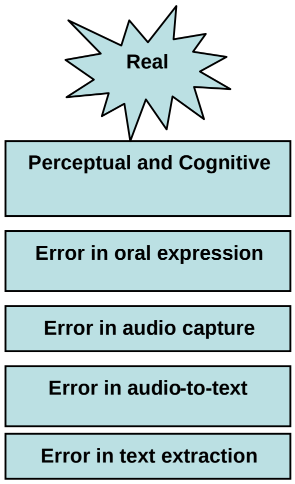

Figure 1. Representative Error Chain.

3. Common Referencing

In our prototype development we are dealing with a single message stream although the individual messages can come from multiple people i.e., multiple sources, and so the Common Referencing function should attempt to account for interpersonal differences in the chain of Figure 1. Although we also ignore this function since in our simulation we have no specification of the observer's identity and personal characteristics, we have also studied some aspects of Common Referencing peculiar to this problem. One factor we think should be considered to reduce the extent of ambiguity due to the extensive range of synonyms for many if not most words (the reason for Thesauri), is to have a front-end processor that does synonym filtering. Brief explorations into the possible strategies for such synonym-filtering suggest that the methods of Latent Semantic Analysis (LSA) and Singular Value Decomposition techniques might serve the purpose. LSA assumes that there is some underlying or latent structure in word usage that is partially obscured by variability in word choice. As part of the possible LSA approach, a truncated singular value decomposition (SVD) would be used to estimate the similarities in word usage across messages. LSA/SVD methods have been employed for real-time applications [3] but a particular strategy for this application has not been fully explored.

4. Data Association

As suggested in Figure 1, the fundamental data entering the fusion processing chain is the extracted digital text. In our prototype, we have employed a generously-donated commercial tool, the Attensity Corporation Text Analytics Suite that is focused on the extraction and analysis of text in unstructured formats. The extraction process produces Resource Description Framework (RDF) Triples comprised of phrases containing a *subject*, a *predicate*, and an *object*. These triples can easily be represented in a graph structure. The direction of the arc (the arc representing the predicate) in the triple-graph is significant: it always points toward the object; the nodes of an RDF graph are its subjects and objects. Any single message may of course yield a set of triples according to the phrases contained in the message, and as messages arrive and cumulate, the triple-graphs can grow large, imputing a requirement for computational efficiency of any algorithm that might operate on these data.

Since (for now) we consider the triple data as validated, correct observations, the Data Association process is rather direct and deterministic, simply linking commonly-named nodes in the overall cumulative set of triples (future research will entail Source Characterization and more rigorous Data Association). The culmination of these operations results in what we are calling a "Data Graph", i.e., the cumulative, to-the-moment set of associated triples; a sample of this graph might look as follows:

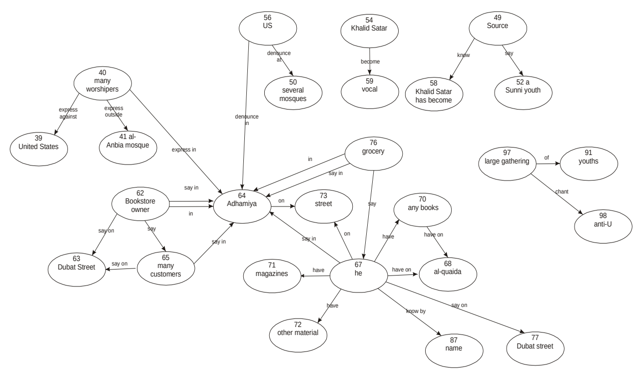

Figure 2. Notional Data Graph of Observables.

Note that the nodes (subjects and objects) may be connected by more than one arc, i.e., by multiple relationships (i.e., predicates, P_i) when triples of the form S-P_1-O and S-P_2-O exist. The ability to express observed relationships is a distinctive aspect of human and linguistic reporting; by and large electronic ISR sensors are feature-based and do not generally provide information on relationships—this is important as regards Level 2 inferencing which is about entities and relations.

5. State Estimation

The Data Graph, as mentioned above, captures the collective set of observations to the moment but does not exploit any a priori deductive model-type knowledge that is typically available and employed in a fusion estimation process (e.g. the target dynamic models employed in Kalman Filters, etc). To integrate such knowledge, we conducted an ontological analysis of the targeted problem space (terrorism and hostage-taking type problems), and developed a prototype ontology that in effect represents such a priori modeling knowledge of the domain; Section 10 describes the ontology development effort. The prototype ontology, developed with the "Top Braid Composer" ontology development toolset, also results in an ontology-graph. As we had the observational data captured in graphical form, we developed a graphical means to integrate the ontological knowledge into the space of "known" data. By examining all the nodes in the Data Graph, we can assert a connection of a node to the Ontology Graph if the observed node is an entity present in the ontology. However, this creates a design choice of how deep to query the ontology from that common node. We chose to make this depth-of-connection choice an analyst option, and labeled this capability as an "n-hop" connectivity or enhancement of the Data Graph, and designed it as a GUI allowing the analyst to choose the depth of connection. This operation results in what we call an "Enhanced Data Graph", which now not only represents real-time, cumulative observational data but also cumulative asserted knowledge drawn from the ontology; an Enhanced data Graph might look as follows:

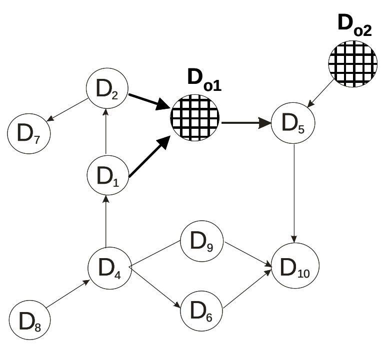

Figure 3. Enhanced Data Graph

In Figure 3, the unshaded nodes and arcs represent observational data as reported in messages; the shaded nodes and related arcs represent either/both of superordinate or subordinate nodes drawn from the ontology that are just one-arc or hop away from the associated observed node. In essence, logic is at work that says "if I see this, and it exists in the ontology, then I will assert that the one-hop neighbors from the ontology exist even though I do not have direct observational evidence of their presence"—this because the ontology is an a priori asserted world model that we believe is valid. This Enhanced Data Graph then represents all that we know at a given time; it is the cumulative direct observational evidence and asserted evidence that describes the "constructed reality" of the real-world state of affairs.

The question now is how do we analyze this (rather large, complexly interconnected) graph to assess whether any conditions of interest are present or not? To do this we engage a Subject Matter Expert (SME) in a knowledge elicitation exercise to develop what we call a "Target Graph". The Target Graph is a graphical representation of entities and relationships of interest to the analyst, depicted also in a graph structure. This process is notionally shown in Figure 4; note that there can be many Target Graphs or relationships of interest.

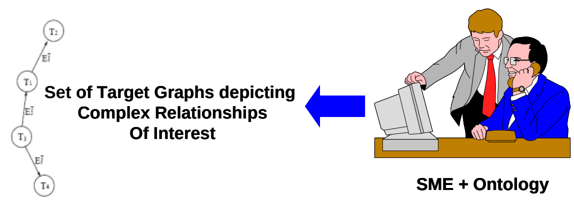

Figure 4. Target Graph Development Process

The question now is to develop a technique to examine whether any instance of a Target Graph (a sub-graph) is present in the Enhanced Data Graph. To do this we employ quantitative methods of Inexact Graph Matching, the details of which are described in Section 9. The inquiry is to compute the degree to which a match occurs. Since both the Target Graph and Enhanced Data Graph may involve complex sub-graph structures, it is of course likely that part of a Target Graph may match a portion of an Enhanced Data Graph; this is the quantitative challenge, to figure out a reliable way to make these degree-of-match assessments (see Section 9.2 for details). The notional graph-matching inquiry is depicted in Figure 5.

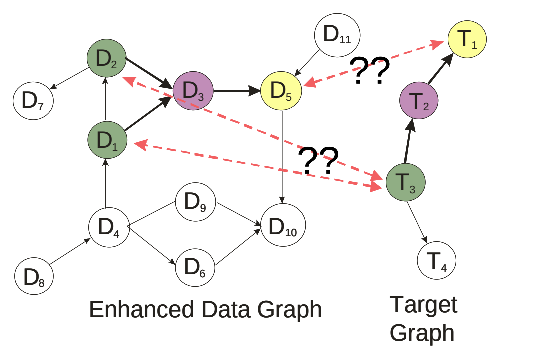

Figure 5. Graph-Matching Challenge.

The matching process yields a set of ranked hypotheses about whether and to what degree the relationships of interest are implied by the observational data and a priori, asserted ontological knowledge; we call this capability a "Relationship Discovery Service", yielding state estimates of the likely relationships in the domain.

6. Social Network Analysis Methods

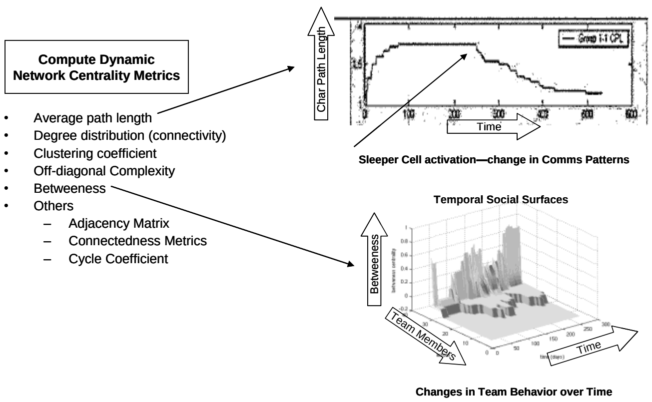

Figure 6. Examples of Dynamic Social Network Analyses [4] [5].

Since the problem domain here is about people and people-to-people, people-to-organization, and organization-to-organization relationships, we inquired as to the applicability of Social Network Analyses (SNA) as an additional tool that could be used for analysis. In exploring the SNA literature, it was observed that most of the SNA work has the following characteristics:

- It is forensic and retrospective, developing analyses from large sets of cumulative data—a "batch" type analysis process
- The analyses treat the observational data, typically framed in a large graphical structure of interrelated nodes akin to our Data Graph, as valid or error-free; i.e., no accounting of observational error seems to be applied[1].
- The analyses proceed from the computation and examination of "Network Centrality Metrics" computed over the batch-set of the cumulative graph structure.

We believe such methods can be effectively applied to this problem domain but we desire to apply them dynamically, in runtime. Very limited research seems to exist in the SNA literature but the small works there seem promising. Examinations of the dynamics of Network Centrality Metrics in [6] have been used to assess sleeper-cell activation and terrorist-group behavioral changes as shown in Figure 6.

The overall prototype analytical toolkit we are currently developing is summarized in Figure 7; our thoughts about our future research and prototyping directions are described in Section 11. The next sections describe the formalisms of the graph-matching methods and the approach to development of the prototype ontology.

7. Perspectives on Evaluation of Soft Data Processes

A completely rigorous test and evaluation process will require a solid Source Characterization analysis of the front end input side as well as an analysis of the errors in Data Association and State Estimation, so that the contributions of each on overall performance can be understood. As indicated previously, these aspects are beyond the scope of this initial research. However, our synthetic Use Case permits the specification of the "Truth" state from which the observations are drawn, and so allows for an evaluation approach that can compare Estimated fused states to Truth states for purposes of basic overall performance assessment of the total process. Our plan for evaluation, to be implemented in our next phase, is to exploit the extensive research we have done on test and evaluation methodologies for data fusion processes [13]. These methods employ formal statistical experimental design techniques and corresponding

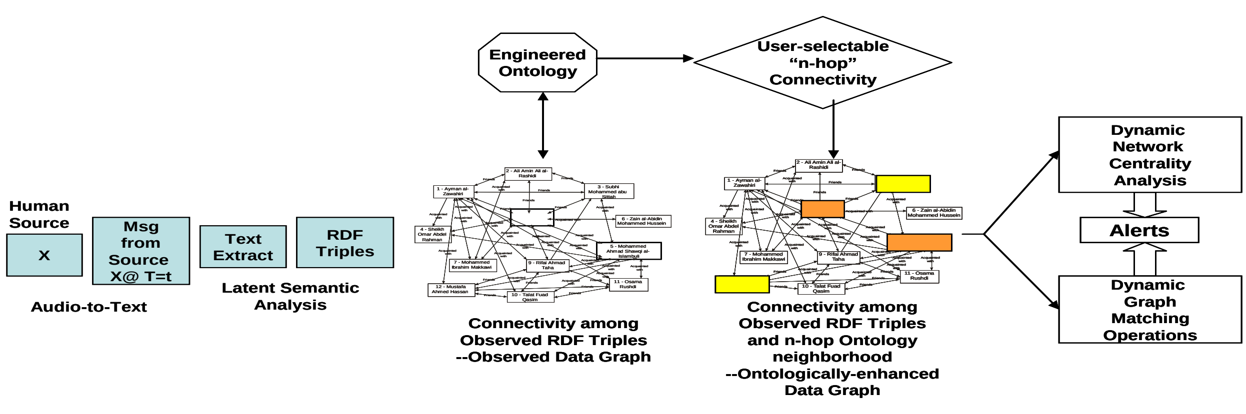

Figure 1. Overall Processing of Current Prototype.

analysis methods that allow statistically-grounded assertions to be made about fusion performance (e.g., analysis of variance, response surface methods, etc.). In order to evaluate the core graph-matching processes, we

[1] We too have not accounted as yet for such errors but fully intend to do so

plan to use Precision and Recall metrics that are usually employed in query-retrieval systems, since the matching process can be viewed as "retrieving" the matching sub-graphs from the enhanced Data Graphs. Since the other innovation we are experimenting with is the ontological enhancement strategy, we are planning to conduct parametric studies that would vary the degree of specificity in the ontology (i.e., the ratio of class-level to instance-level knowledge), and also to vary the number of hops in the "n-hop" enhancement scheme to see how performance is affected by these parameters. As the project expands to include the Social Network Analysis aspect, additional evaluation schemes will be developed.

8. Preliminary Results

The current project is a basic research project primarily focused on better understanding of the Soft Data type problem and its implications for information fusion processes. So in our synthetic use case, the message data (observations) drawn from the specified "truth" state is processed and represented as rdf triplets.

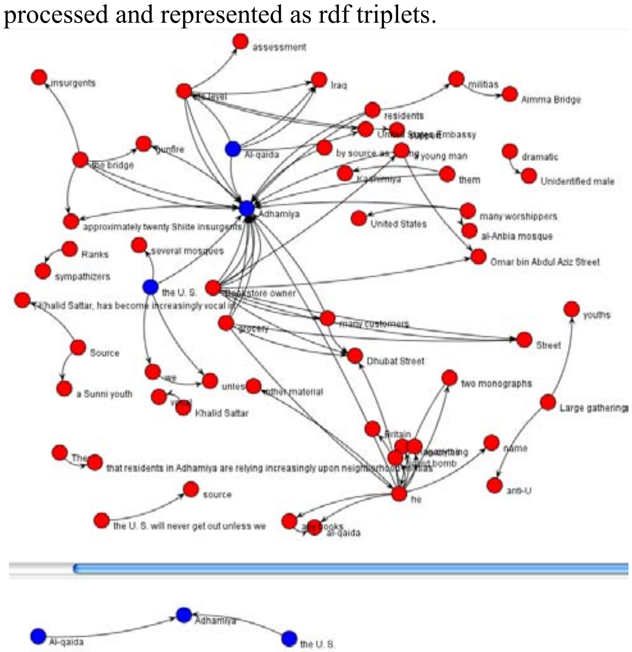

Figure 8. Target graph match in data graph.

An example result with a match of a target graph in the enhanced data graph is shown in Figure 8. A small target graph with 3 nodes and 2 links is matched against the enhanced data graph with 57 nodes and 74 links. The details of the matching process are explained in Section 9.

9. Formalisms of Inexact Graph Matching [8]

9.1 Introduction, definitions and notation

Many different graph matching techniques can be applied to search for an isomorphism that will exactly match the target graph and some portion of the data graph generated from the message data and ontology. However, in asymmetric problem environments with high uncertainties, the isomorphism condition can be too strong in many real problems and cannot be expected between both graphs. In these cases, such problems call for inexact graph matching, and the aim on these is to search for the best homomorphism possible. The inexact graph matching problem has been proved to be NP-hard, and therefore heuristic algorithms that provide an approximation to acceptable solutions are required; in the following sections we describe our particular approach

A graph is represented by an attributed structure as $G = (V, E, A, a_V, a_E)$, where V is a set of nodes, E is a set of arcs, A is a set of node attributes, and a_V: $V \rightarrow A$, a_E: $E \rightarrow A$. Furthermore, we assume that the graph is directed. The data graph in our problem is denoted as $G_D = (V_D, E_D, A_D, a_{VD}, a_{ED})$, where $V_D = \{D_1, D_2,..., D_n\}$(n = number of data graph nodes) and $E_D = \{E_1^D, E_2^D, ..., E_{n'}^D\}$(n ′= number of data graph edges). Similarly, the template is represented as $G_T = (V_T, E_T, A_T, a_{VT}, a_{ET})$, where $V_T = \{T_1, T_2... T_m\}$ (m = number of template nodes) and $E_T = \{E_1^T, E_2^T... E_{m'}^T\}$ (m ′= number of template edges). Figure 9 shows a simple example of a data graph and a template, which is used as an illustrative example for our solution methodology.

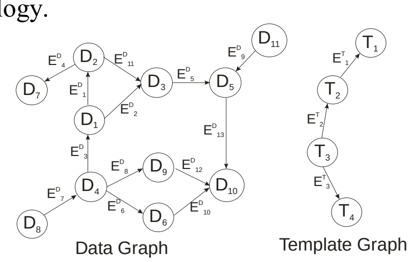

Figure 9. An example of data graph and template.

9.2 Similarity function

In all graph matching approaches, there is a need of measuring the semantic similarities of the attributes in the nodes and arcs of the templates and the data graph. The most common similarity function is by the aggregation of the values of a graph structure that take into account the one-to-one weights of nodes and arcs. Let $v_i \in V_T$ and $v_j \in V_D$, then there is a similarity weight between the node on the template and the node on the data graph, defined as $S^n(v_i, v_j)$. Correspondingly, given $e_{ih} \in E_T$ and $e_{jk} \in E_D$, there is a similarity weight between these two arcs, defined as $S^e (e_{ih}, e_{jk})$. These similarity functions depend on the problem and semantics. We assume that these similarity functions are developed and specified via collaboration with SME's.

Although such an algorithm is extremely efficient, it is prone to a number of mismatches that could be problematic in a real application. Our strategy expands on the triplet idea and what we define as 1-Hop neighborhoods. These 1-Hop neighborhoods consist of a root node and all other nodes of edge distance 1 away. The similarity function stated previously results in identical sets of triplets. However, using 1-Hop neighborhood approach, we have the benefit of dissimilar sub-structures. For a pair of template node and data graph node, a linear assignment problem is solved over their neighborhood. The linear assignment problem takes the score matrix with individual elements as the average of the neighboring node-to-node score and connecting edge-to-edge score. So each 1-Hop neighborhood score will be a unique score depending on its corresponding neighborhood.

We define two parameters of the procedure to increase the flexibility of the solution and allow the algorithm to be configured for different domains of data.

1. α **(root weight score)**: weights the value of the match between neighborhood root versus the value of the assignment of its neighbors.
2. t **(score threshold)**: The algorithm will not return matches with a value below this threshold. The higher this threshold is set, the fewer 1-Hop neighborhood assignments would be determined, which in turn leads to improved performance (but potential loss of optimality).

Step 1 of the procedure is to compute a node-to-node score (C_{ij}) for each node in the template graph to each node in the data graph. For each node we then sort this list in descending order. Using the threshold value (t) as input to the algorithm, we can prune the amount of assignments we must run for this template node by $\frac{t-1+\alpha}{\alpha}$. Root node scores which are below this value do not have the possibility of having an overall score above the threshold even when there is a perfect neighborhood assignment score.

Step 2 of the procedure is to compute the scores for the 1-Hop neighbors of each root node pair. This returns the optimal assignment of neighbors of the root node in the template graph to the neighbors of the root node in the data graph. Then the neighborhood score between template graph node i and data graph node j is given by ($\alpha \times C_{ij} + (1 - \alpha) \times W_{ij}$), where C_{ij} is the score of the root node pair and W_{ij} is the score of the neighborhood assignment, which is given by Equation 1.

$$W_{ij} = \frac{\text{Sum of neighborhood assignment scores}}{\text{Number of neighbors to root template node}} \qquad - [1]$$

1-Hop neighborhood score takes care of "node-to-node" assignment as well as "edge-to-edge" assignment. The score of neighborhood assignment is solved using a linear assignment problem between the root nodes with the adjacent nodes and edges forming the solution matrix.

9.3 Matching Process Details: The TruST Algorithm

A truncated search tree algorithm, TruST [7][8][9], which is akin to beam search algorithm, is a heuristic search algorithm that is an optimization of best-first search. The disadvantage is that such methods are not guaranteed to yield an optimal solution. However, the reached sub-optimal solution is a very good one, for most of the times. Through the control of the truncation parameters one can obtain various efficiency tradeoffs.

Zhang [10] in his paper has shown that truncated branch-and-bound as a heuristic gives better results than most of the heuristics. TruST is based on a similar truncated branch-and-bound approach which works on graphs. A typical example of this algorithm is shown in Figure 10. The search tree is developed dynamically during the search and initially consists of only the root. At each iteration of this algorithm, a subproblem is selected for exploration from the pool of *live* subproblems using the scores of the current match. We use here a strategy which is similar to the breadth first search strategy found in the literature.

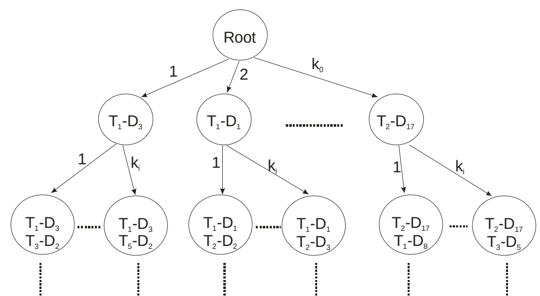

Figure 10. An example of TruST algorithm.

In what follows, we consider the branching rule for the selected subproblem. Each subproblem is developed by adding one pair to its parent problem. Topology is the most important factor considered in this step. Those (template, data graph) node pairs which are qualified to be added should be connected respectively to at least one template and one data graph node in the parent problem. The way in which a newly added template node connects with the existing template node must be exactly the same as the way that connects the data graph nodes in the corresponding pairs. When a new data graph node is added, the score at that level is calculated using the average of the scores of all the node pairs and edge pairs connecting them, at that level. So as the level increases the new nodes are added and the score is revised in correspondence with the new pair score.

Note that each subproblem at level i (i = 0, 1, 2 ...) has exactly i pairs (one data graph node and one template node). In summary, the more pairs exist in a subproblem, the more neighboring nodes should be considered. That causes the search tree to explode exponentially as it goes deeper. In order to avoid such a problem, we do not consider all the feasible pairs. We use a parametric mechanism to control the state space growth. For a subproblem at level i, we only choose at most k_{i+1} best child subproblems. In our work, we first set k_0 to be a fairly large number. The reason to do so is to make the starting point cover the data graph as much as possible. And then we set $k_i = k$ ($\forall$ $i \geq 1$) with k being relatively small in order to reduce the exponential growth. However the search tree still remains very large. For instance, at level i, the number of sub-problems is given by $k_0 k^{i-1}$, which is extremely large even if the values of k and i are relatively small. Therefore, we introduce two additional parameters β_i and δ. The parameter β_i controls the breadth of the search tree, i.e., the total number of sub-problems at each level is at most β_i. The parameter δ controls the depth of the search tree, i.e., the search progress stops at level δ with only part of the template explored. If two or more sub-problems at the same level have exactly the same matched pairs, we only retain one of them and fathom the others. It is preferable to let δ be equal to the number of nodes in the template graph for complete exploration. An illustrative description of this state space is presented as follows.

```
ParentBundleSet = Initial Dummy Parent Bundle;
AvailableMappings(InitialDummyParentBundle) = Initialize and sort
mappings;
genTreeLevel(ParentBundleSet);
for each ParentBundle in ParentBundleSet {
   if (TreeLevel ≥ δ OR TreeLevel > NumTemplateNodes) {
      Output(ParentBundle);
   }  else {
      if (no adjacent mappings) {
         NewParentBundleSet = InitialDummyParentBundle;
         AvailableMappings(InitialDummyParentBundle) =
         AvailableMappings(ParentBundle);
      }  else {
         for (k best available mappings taken from adjacent mappings) {
            NewChildBundle = ParentBundle + NewMapping;
            if NewChildBundle = = Previously Generated Bundle {
               fathom(NewChildBundle);
            }  else {
               AvailableMappings(NewChildBundle) =
               AvailableMappings(ParentBundle);
               findAdjacentMappings(NewChildBundle);
               NewParentBundleSet += NewChildBundle;
            }
         }
      }
      NewParentBundleSet.sortDescending;
      if (NumNewChildBundles > β) {
         NewParentBundleSet = NewParentBundleSet.subset(0, βi);
      } genTreeLevel(NewParentBundleSet);
   }
}
```

After running this algorithm, each branch yields a series of matched pairs. Then the data graph nodes in the matched pairs form a subgraph, which is a final match for the template. There are a bunch of such resulting subgraphs with various matched values and topologies, referred to as *leaf nodes*. At each level in the algorithm, the best βi leaf nodes are selected. If any of the leaf nodes cannot be extended at level i, then there are no adjacent nodes in the data graph corresponding to the template graph. So a penalty is added to the node and the tree is expanded with some non-adjacent node having a lower 1-Hop neighborhood score. The worst-time complexity of full enumeration of the algorithm is O (m n^m), but in terms of user parameters the complexity is: O ($k_0 \beta^{\delta}$).

10. Ontology Specifications

The ontology provides an a priori model that specifies the domain in question by defining abstract metaphysical objects, processes and relationships (represented by upper-level categories such as Substance, Dependent Attribute, Spatial Region, Temporal Process) as well as concrete, domain-specific objects and relationships (represented by the lower-level categories gathered from subject matter experts and domain-specific resources) [11]. The items contained in the upper-levels of the ontology are the product of formal metaphysical reasoning and are applicable across multiple domains. They are captured as basic categories in the ontology's hierarchy. The items specific to applications such as hostage-taking or bomb-making are gleaned from domain research, interaction with SME's, and ingestion of other related sub-ontologies. These are captured in the ontology as lower-level categories or instances (i.e., particular items such as named individuals). An example of such a lower-level ontology segment is shown in Figure 11. The graph structure of the ontological data provided by the TopBraid Composer tool allowed for efficiency in linking the ontology to the Data Graphs to enable the ontological enhancements spoken about in Section 5.

A critical first step in the ontology development process involves some manual extraction of domain-specific terms, which can be used to construct a lexicon that contains all of the relevant items from that domain, along with their definitions (which can also be inserted into the ontology as annotations) [12]. It is important to capture as much information about the domain as possible, in order to build an appropriately complex ontology model. The lexicon must also remain an open research item, which can be developed further over time.

Utilizing the combination of abstract and specific categories, the ontology can provide contextual information about many kinds of items discovered in the Attensity natural language processor, and in turn, provide these kinds of contextual relations to the graph-matching algorithm. The level of specificity of the ontology is an issue, since it is unlikely that the ontology will contain massive amounts of instance-level (i.e., highly specific) information. Instead, the ontology is more adept at providing links to related categorical items and attributes. Furthermore, one must make a decision as to whether to allow the ontology to provide inferential RDF triples (i.e., those produced by its internal reasoner) to the graph-matching process, or whether only asserted triples in the ontology (those assumed as a priori facts about the world) will be allowed.

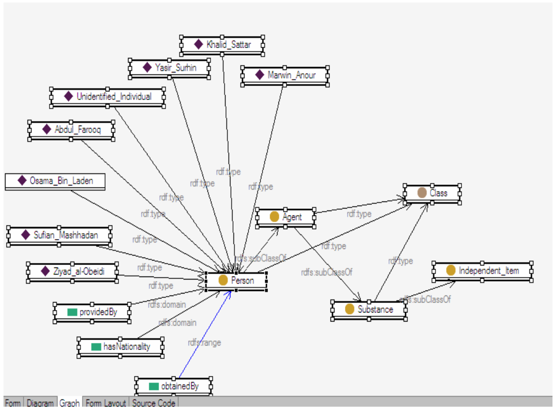

Figure 11. Instances of person in the Ontology.

11. Conclusion

This basic research project is a first step toward developing a robust information fusion process and capability to deal with the new and challenging issues of fusing soft/linguistically-based observational data in real-time. Our approach, which we think will be typical of such approaches, involves the integration of a variety of different stepwise processing modules that have typically been standalone functions, e.g., text extraction tools, social network analysis tools, and fusion tools. From the outset, the design was focused on constructing a capability to handle the dynamic, real-time requirement to generate actionable intelligence, and has shown promising real-time performance and in fundamental inferencing/state-estimation via quantitative techniques based on graph-theoretic principles. The design is unique in its integration of observational data with a priori knowledge in an efficient, graph-based representational scheme that has shown good scalability. However, much more needs to be done to address various other design challenges, to include multiple message stream inputs, modeling of input sources and processing errors, inclusion of dynamic social network analysis methods, and in formalized parametric testing.

Acknowledgment

This project was supported by the U.S. Army Research Laboratory (ARL) and the Tactical Data Fusion and Exploitation Branch; CMIF is appreciative of the efforts of Dr. Barbara Broome and Mr. Mark Thomas of ARL in sponsoring and guiding this project.

References

[1] Z. Chen and D. Arieh, *On the fusion of multi-granularity linguistic label sets in group decision making*, Computers and Industrial Engineering, Volume 51, Issue 3, pp. 526-541, November 2006.

[2] Y. Q. Zhang et al., *Granular neural networks for numerical-linguistic data fusion and knowledge discovery*, IEEE Transactions on Neural Networks, Volume 11, Issue 3, pp. 658 – 667, May 2000.

[3] P. Praks et al., *On SVD free latent semantic indexing for image retrieval for application in a hard real time environment*, Proc. IEEE Int. Conf. Industrial Technology, Maribor, Slovenia, pp. 466-471, 2003.

[4] T. Coffman and S. Marcus, *Dynamic classification of groups through social network analysis and HMMs*, IEEE Aerospace Conference, Volume 5, pp. 3197-3205, 6-13 March 2004

[5] P. A. Gloor, *Capturing team dynamics through temporal social surfaces*, Ninth International Conference on Information Visualisation, pp. 939-944, 6-8 July 2005.

[6] T. Frantz and K. Carley, *Relating Network Topology to the Robustness of Centrality Measures*, CASOS Technical Report, Carnegie Mellon University, May 2005.

[7] K. Sambhoos et al., *Enhancements to high level data fusion using graph matching and state space search*, Submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007.

[8] K. Sambhoos, *Graph matching applications in high level information fusion*, Ph.D. Dissertation, February 2007.

[9] M. Sudit et al., *A Graph-Based Framework for Fusion: From Hypothesis Generation to Forensics*, The 9th International Conference on Information Fusion, Florence, Italy, July 10-14, 2006.

[10] W. Zhang, *Depth-first branch-and-bound versus local search: A case study*, In Proc. 17th National Conf. on Artificial Intelligence, pp. 930–935, 2000.

[11] E. Little and G. Rogova, *Ontology Meta-Model for Building A Situational Picture of Catastrophic Events*, in Proceedings of the FUSION 2005-8th International Conference on Multisource Information Fusion, July 25-29, Philadelphia, PA, 2005.

[12] E. Little, *A Proposed Methodology for Application-Based Formal Ontologies*, Proceedings of the Workshop on Reference Ontologies vs. Application Ontologies, 15-18 Sept., University of Hamburg, CEUR-WS.org., 2003.

[13] K. Sambhoos and J. Llinas, *Designing a Performance Evaluation Methodology for Data Fusion-Capable Tactical Aircraft VI* (Oral Presentation), AIAA U.S. Air Force T&E Days, February 2008.