HOUND : From Discourse to Actions

High-accuracy Ontology-based Understanding of Natural Domains

Committee Members

Jaime Carbonell

Eric Nyberg

Manuela Veloso

Vasco Calais Pedro

Language Technologies Institute

Carnegie Mellon University

Acknowledgments

I would like to thank all the people that made this possible. The path of research is sometimes lonely, but this was not the case.

My Grandmother Helena and my Mother, for always being there with a gentle hand and the certainty of a home; My Father and my second set of parents, João and Célia, for all their encouragement and support; In one way or another they define a big part of who I am and why I am that way.

My daughter Beatriz, which brings a new bright smile into my life with the beauty of new beginnings.

My love, wife and friend Joãozinha, my pilgrim soul in sun or rain; the long journey that we embarked in is not yet over, but it would have never started if it wasn't for her soft gentle voice in my ear. For every morning that I wake up by her side I thank the fortune of having found happiness. The world is better place because she exists.

Finally, the people that directly influenced this research with their guidance, advising and patience, Jaime Carbonell and Eric Nyberg, my advisors, without which the research would never have existed. In case they need reminding, there will always be a student in awe for the attention they devoted to me.

I am very fortunate to have so many people that fill my life with support, friendship and love. I hope that in some way I can deserve all that they give me.

Abstract

Correlating information from multiple large information sources is very expensive. The current communication infrastructures allow rapid information propagation and flexibility in the means by which that information is accessible. A considerable amount of information is distributed in the form of natural language documents, but the existence of tools that parse and correlate the information contained in large scale document streams is still somewhat inadequate. It is necessary to develop methods to reduce the amount of time spent in information analysis and facilitate the task of finding the relevant information nuggets. Furthermore, the existence of coherent and stable natural domains should allow for the automatic production of expected predictions in those domains, highlighting relevant facts subject to further exploration. The efficiency of any information analyst in areas such as Question Answering and Intelligence Analysis would greatly benefit from the existence of a dedicated framework for the analysis and reasoning with semantic information.

This thesis describes the motivations, conception and implementation of the first version of the HOUND (High-accuracy Ontology-based Understanding of Natural Domains) system. HOUND minimizes the effort dedicated to information analysis by correlating semantic information within multiple documents from various sources and predicting possible actions of the objects in the system. The correlation is based on the rule-based description of the natural domain we are addressing. Implemented in a server-client architecture type, HOUND fully justifies the reasoning behind any prediction with a semantic trace and allows the quick visualization of the information timeline in the form of the ordered set of documents that originated that prediction thus enabling the user to quickly reach the information nuggets within the documents.

1 Introduction

1.1 Motivation

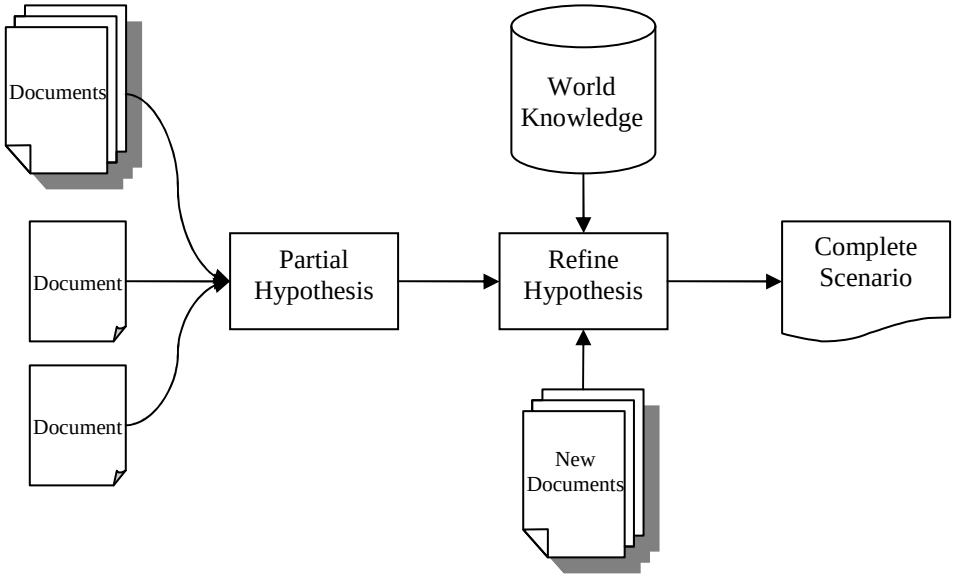

Fig 1: Concept Outline

Information Extraction is the mapping of short, unstructured natural language texts into predefined, structured representations, or templates which, when filled, represent an extract of key information from the original text [GH97].

When humans extract information from the text they do more then just extracting and representing the key information. They work with the acquired knowledge to reason, generate new hypothesis and to derive conclusions. Concept mapping and semantic nets, for example, are strategies that can be used in a complementary way by humans to perform tasks such as learn new information and teach this new information to others [Fis90].

There are many issues to consider when analyzing the human processing of information. Humans have robust parsing capabilities and extremely flexible ways to convert syntactic representations into semantic representations, thus solving anaphora and co-reference phenomena with ease. But context and world knowledge play a crucial role in this matter. They fill the gaps of the non-explicit information, and are the base upon which is possible to reason.

In view of this, if we want to achieve the same efficiency in these kinds of tasks, the representation of information presents only a first challenge. We need to be able to derive knowledge from knowledge and implement the approaches that will allow us to do that.

When the information that could lead to a conclusion is in different documents and the flow of information is bigger than our capability of analysis this becomes unattainable. The situation becomes critical when the information can have different levels of urgency. In situations like the stock market analysis or the intelligence community, the flow of information can have leads that could affect in a substantial way our next decision. It is when this happens that we need to automate the process to deal with the incoming reports, newswires, etc.

However, if our goal is to establish a process that does not pretend to emulate human thought but help humans navigate the flood of raw text, pointing out possible valuable information and possible consequent events based on the existing information and domain knowledge, then we just might have a chance to do it.

1.2 Problem Statement

The analysis of overwhelming amounts of crucial information requires enormous amounts of human effort.

If we consider an intelligence analysis scenario, where the analyst must connect the dots between the data gathered in surveillance reports, the task requires the analyst to examine and combine information from multiple reports over a period

of time. There is, in this case, a need to map relations between the events describes in the report in such a way that all possible relevant conclusions are drawn.

It is often observed the use of different expressions to express the certainty of the information.

Here is an example of a possible set of surveillance reports.

> *"Army CID Report [21 February, 2002]: report of theft of small arms ammunition, four M-16 rifles, six grenade launchers, and six Manpads. This theft occurred at a military installation in Colorado. [This theft is still under investigation]."*
>
> *"CIA Report [20 September, 2002]: the person identified as John H. in FBI Report was seen in London, UK on 12 June, 2001 in company with Lofti Raissi who may have helped train the four hijacker pilots involved in the Sept. 11, 2001 terrorist incidents. It seems that the name John H. is an alias for Omar K, a person known to be associated with AL-Qaeda."*
>
> *"FBI Report [8 July 2002]: Recorded conversation in Denver, Colorado on 1 July, 2002 between Homer W. of the Aryan Brotherhood of Colorado and a person who identified himself as John H. This conversation involved the sale of weapons of an unspecified nature. Homer W. is an Army sergeant being investigated in connection with the theft of weapons reported in Army CID Report."*

Fig 2: surveillance reports example.

It would be most advantageous to have a tool that would extract the semantic information from documents such as the ones in the example above, automatically reason with that information and point us to relevant pieces of information and possible consequences that can be derived from that information. By using this, the analyst could focus on relevant information, not having to deal with all the noisy information contained in the documents.

Typically, the kind of situations described above is very task oriented and dwells in a specific domain. Therefore one could establish a domain description, the "physics of the system", that would supply the world knowledge that is pertinent to our domain.

The problem is mainly divided in two distinct parts:

a) Achieving the Semantic Representation.

b) Reasoning Towards Conclusions.

1.2.1 Achieving the Semantic Representation

The first part of the process involves the creation of a semantic representation from raw text. Not only we want to be able to parse any kind of text, but also we want to produce a semantic interpretation that is usable in a reasoning process. We cannot end up with a bunch of sparse and unconnected facts but we also cannot expect that the information available will always lead to a conclusion, since most of the times it will not.

This process is complicated by many factors. The first of these factors is the variability of the inner structure of information when is represented in text. That is, there are many ways to say the same thing and we need some way of unifying different expressions pertaining to the same concept.

Another factor is the uncertainty that exists in human language. This uncertainty is expressed in many ways, amongst which are the usage of different modal verbs to establish different level or certainty between relations *("**must** have bought"* vs. *"**could** have bought"),* the usage of adjectives to modify properties of expressions (*"**long** meeting"* vs. *"**brief** meeting")* and the different degree of certainty that context attributes to a sentence. We cannot, therefore, operate in a black and white world, for it does not correspond to reality. This uncertainty must be modeled in the semantics of our representation.

The third factor resides on the fact that the information will be spread across a range of documents over a period of time. This means that most of the time we will extract, at most, partial information but we must be constantly looking at the state our world is in and what kind of new information we can derive from our knowledge.

1.2.2 Reasoning towards Conclusions

Assuming the existence of a coherent representation that suffices, we must derive conclusions from this representation.

The goal of the HOUND system is, given an initial domain representation and the semantic representation of text, to relate the information we have and to derive conclusions. We must also keep in mind that, since the world is an extremely dynamic environment, there must be an easy way to change the system, allowing a user to reshape the domain knowledge, redistributing the uncertainty measures and adding new forms of relation.

1.3 Thesis Statement

In a close domain environment, given a defined set of possible interactions, it's possible to automate the processing of the input texts and the reasoning involved in bringing to the surface the interesting facts and deriving hypothesis.

We can subdivide this in several sub-tasks:

a) Extract the facts from the input texts.
b) Extract the predicted events from the current events.
c) Trace the predicted events to the supporting evidence and output it to the user.

The automatic process should lead to an output of derived hypothesis, substantiated by documents in the corpus. These documents should amount to a conclusion achieved only by the combination of the information of various documents.

I believe it's possible to detect these relations between the documents and expose those relations in such a way that we can derive conclusions automatically.

1.4 Scope

Although the complete process involves the extraction of semantic information and the reasoning with that information, it is not the scope of the current work to implement both questions, especially when the extraction of semantic information is already being addressed [VHKN03], [FR94]. Rather, this thesis will focus on the implementation part of the task, reasoning with semantic information and assume that the first part is taken care by 3rd party software.

Furthermore, some simplification must be done for implementation purposes. The current implementation of the system must be understood as a first attempt at a large problem and the benefits will be, at most, the clarification of the issues to address in a full implementation. Therefore, some assumptions are made for simplification purposes:

a) Time information is discarded. I acknowledge the extreme importance of this information but the treatment of temporal information is an area by itself and is not part of the scope of this thesis.
b) All information regarding uncertainty is discarded. It is the intent of the author to address this issue at a later stage but it is imperative that the bases for semantic reasoning and server architecture are tested before the problem is complicated further.

Given the constraints described above, the scope of this thesis consists in two main tasks.

1.4.1 Extract the predicted events from the current events.

Predicting an event is a very vague term. One could say that

"if I drop an apple I can predict that it will fall. "

This is true, as long as the law of gravity holds, but is not something that most people would call relevant since it is taken as a natural consequence of the action of dropping the apple.

Predicting actions becomes interesting when the predictions are not necessarily trivial events but alerts to unforeseen possibilities. One important rule to keep in mind is that most of the events that we are looking for do not derive of simple rules, but from sets of situations that make their outcome possible. Even when all the necessary conditions are in place the events might not happen. There are factors, like the human will, that cannot be modeled with a good degree of confidence and therefore represent random factors in our world. We can only predict up to a point.

I propose to develop a system where through the interaction of different rules at different levels of the domain hierarchy, situations are triggered. These rules will operate at different levels of abstraction and have to have the flexibility of changing either due to direct intervention of a user, negative feedback or negative examples. One simplistic example would be:

"terrorist(X),has_bombs(X),has_money(X) - > Terrorist Attack will occur"

But we must also be flexible enough to reach "*has_bombs(X)*", for example, without having it explicitly written.

Sometimes relevant predictions will be made of subtle webs of relations, like information about an individual or an association between organizationz based on scarce evidence. Any hypothetically valid conclusion must be shown to the user, for it could be relevant for his purposes.

1.4.2 Trace the predicted actions to the supporting evidence

Tracing the information back to its origins is vital in justifying our predictions. The last step is to do just that. Whatever conclusion we reach must be traceable to the document or documents from which originated, even if the concept is expressed by a different verb and there is a fair amount of uncertainty.

This will allows us to understand the reasoning context and the steps taken by the system in manipulating the information. When we have for example a set of documents that form a timeline pointing to a certain event, the user should be able to

view that timeline in the representation of the documents, so that he may also consider other events occurring at that time.

1.5 Background and Literature Review

By no means is this section pretended to be a full account of previous work. There are many important works that will be overlooked in here and the aim is to give an overview of the field's history that is relevant for the proposed work.

In order to describe the previous work I think there are two areas we must consider

- Information Extraction.
- Semantic Reasoning.

1.5.1 Information Extraction

A survey of the Area of Information Extraction was first done by J. Cowie in 1996 [CL96]. Following I will give a brief account of the area.

Some of the first work in the area of template filling was done in the 60's and 70's by Roger Schank [S75] in language understanding. Schank pointed out the patterns occurring in stories referring to them as scripts that could be filled with information from texts. The first attempt to build a so called IE system using Schank ideas was made by one of his students, Gerald De Jong. The system was called FRUMP [D82]. This system used what De Jong called Sketchy scripts and it was used to process texts directly from a UPI news wire feed.

Amongst the first commercial systems developed in this area was JASPER [AH92]. This system was developed by the Carnegie Group and contained a "fact extraction" system for Reuters. It was designed to fill templates with information about companies and dividends based on company press releases. Candidate news stories were produced based on these templates saving journalists time in the story preparation.

In the mid 1980's the First MUC (Message Understanding Conference) was created in order to allow different projects being developed at that time to be compared. The series of MUC conferences was sponsored by ARDA. The evaluations became more complex and focused on different tasks from one year to the next and each of them increased the level of difficult that the projects had to achieve.

The MUC evaluations provided the IE community resources, evaluation tools and perhaps above all a sense of identity and a forum for exchange of ideas [GW97]. Some radical changes in the approaches for IE were seen in the years of the MUC conferences. Mainly the approaches started to change from a full linguistic approach to a shallow parsing approach, from a theoretically driven approach to an engineering driven approach. One example of this was the TACITUS system [H91], which changed to the FASTUS system [HA92] from the MUC-3 to the MUC-4. While the TACITUS system relied heavily on NLP and a full syntactic analysis and semantic interpretation as well as a first-order predicate calculus representation, the FASTUS system had a cascade of finite state transducer for tokenization of names, phrases and patterns. The changed allowed more speed, but the main reason was not that the first approach was at fault, but rather that it was found that, to the task at hands, a more specific approach would be better.

Current work being done in the area involves building a discourse model lexical processing, parsing and semantic representation, from which the result can be read. An example of such system is LaSie [GH00]. LaSie was designed as a general purpose IE research system and was initially developed for the MUC-6. Version 2 is currently being implemented in JAVA.

In creating a mental path of development of this area, one cannot avoid noticing a changing back and forth between two different approaches as more advances were made available, Rule Based Systems and Statistical Based Systems.

This change seems to be cyclical and I believe the trend is to combine the two approaches on different parts of the systems. The robustness of statistical approaches is certainly more usable in certain situations, but when we want to treat situations that require a fine grain approach, the preferred solution seems to involve Rule Based Systems that try to capture the different subtleties and exceptions.

1.5.2 Semantic Reasoning

Semantic Reasoning is a field that originated in AI problems dating back to the 60's. Semantic Reasoning deals with inference, deduction, induction and abduction from a set of semantic concepts or representations, independent of the form of the representation.

In 1966 Ross Quillian demonstrates semantic nets in his PhD thesis in Carnegie Institute of Technology (now CMU). The goal behind it was to model the structure and storage of human knowledge in the space of a graph [Q68]. This allowed the exploration of the meaning of English words trough their relationship in a computational fashion.

Using the concepts of Semantic Nets, Jaime Carbonell (Sr.) developed SCHOLAR [CA70], a system for computer-aided instruction. This was a fact-oriented system that needed a database with a structured set of connections in many dimensions between concepts and facts.

In 1976 Doug Lenat's develops the first system that, based on a discovery model, is able to derive interesting conjectures [L82]. One of the goals was to instill a machine with common sense and common knowledge and learn from it. Although it was applied to mathematics, this system was able to predict important discoveries and it was thought as the first piece of original mathematical work produced by a computer.

Many other systems have been built since then, many of them with full logic capabilities, but very rarely the attempt is made to reason with information directly from unstructured text, based on semantic information.

One of such attempts came from a framework called Description Logics [DG98], that deals with information integration from a constraint-based perspective. Mozart [M99] is system platform designed for distributed programming, symbolic computation and constraint-based inference.

Perhaps the most relevant work for the task at hand comes not for the Computational Linguistic field, but rather from the Computer Science field. In 1984, Charles L. Forgy publishes *Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem*, which describes an efficient algorithm for the pattern matching problem based on rooted acyclic direct graphs [F82]. This algorithm is currently regarded as the most efficient in rule matching and although a new version of the algorithm has been released, the original idea is still extremely compelling in its scalability and adaptability.

As more tools and systems become available, the array of applications and experiments one can conduct becomes more complex and interesting and although reasoning is now a mature area, much is yet to be known on what process could lead us to a reliable source of semantic based prediction.

1.6 Challenges

There is a gap between the two fields described above. The vast majority of the systems have focused either on information extraction or in reasoning, but not the combination of both. Many are the challenges in this research. Given the scope of this thesis the pertinent problems are directly related with semantic reasoning and integration of the two distinct components.

There are two main challenges that must be addressed. The first is the creation of a modular and flexible architecture that combines semantic extraction and semantic reasoning in a server/client architecture that maintains information status independently of the number or clients that supply information and receive the produced information. The second challenge is the implementation of a matching algorithm for the rules and facts. The algorithm must be scalable and efficient, capable of handling different kind of rules and a complete set of operations.

2 Methodology

This thesis was done with an object oriented approach in mind. Object oriented approaches are not only suited to describe the concepts that we represent with semantic information, but also is generally used to create modular architectures.

2.1 Definitions

2.1.1 Fact

A fact is a piece of information about circumstances that exist or events that have occurred. In the context of this thesis a facto is also the semantic representation of a piece of information.

A fact is composed of:

- A predicate that indicates the type of fact.
- A list of arguments, or parameter/value pairs.

Although there is no constraint as to the number of types of facts, for the purpose of all the examples that relate to text-extracted semantic information I used only 4. I believe that it's possible to describe the vast majority of situations with these four fact types and the usage of a limited set improves the easiness of the text extraction process.

2.1.1.1 Object

An object is a reference to a world entity and its type, if known. Every time there is a reference in the text to a new entity we create an object entry.

Example 1: *object(name=al_qaeda,type=organization)*

predicate type	*object*
object name	*name= al_qaeda*
object type	*type=organization*

Table of contents: example 1

2.1.1.2 Property

A property refers to an attribute of a particular object. It can be the type of organization or the color of a car. There is no restriction on the number of arguments a property can have.

Example 2: *property(name=al_qaeda,org_type=terrorist)*

predicate type	*property*
organization name	*name= al_qaeda*
organization type	*type= terrorist*

Table of contents: example 2

2.1.1.3 Relation

A relation is a way in which two or more objects are connected, associated, or related. It can be an is_a type of relation, but can also define relations of possession, location, etc.

Besides the predicate specifying the type of fact, the first argument defines the type of relation.

Example 3: *relation(type=is_in,object1=denver,object2=colorado)*

predicate type	*relation*
relation type	*type=is_in*
object 1	*object1=denver*
object 2	*object2=colorado*

Table of contents: example 3

2.1.1.4 Action

An action fact type refers to an action performed by one or more objects in the system. Two people meet, a person rents a car, someone enters the country, etc.

Besides the predicate specifying the type of fact, the first argument defines the name of the action.

Example 4: *action(name=meeting,object1=ralph_t,object2=george_w)*

predicate type	*action*
action name	*name=meeting*
object 1	*object1= ralph_t*
object 2	*object2= george_w*

Table of contents: example 4

2.1.2 Pattern

A pattern has the same form of a fact, with the exception that not all the values are necessarily defined, that is, it contains variables. Variables are denoted by upper case Letters or words (*X*, *PERSON*, *Country* are some examples).

They matched against the existing facts of the system and when we find a coherent unification we have a partially fulfilled rule.

Example 5: *object(name=NAME,type=organization)*

template type	*object*
object name	*name= NAME*
object type	*type=organization*

Table of contents: example 5

Example 5 shows a pattern that will match with all objects whose type is *organization*.

2.1.3 Rule

A rule is a principle or condition that customarily governs behavior within the specific domain. It capitalizes on the intuitive knowledge of the domain knowledge builder. It reflects common sense in some cases and could be the result of

intuitive empirical notions. The form of a rule is the form of IF-THEN clause. If the antecedents are true then the consequences are also true.

More specifically:

- Antecedents – a set of one or more patterns with a semi-colon in between.
- The symbol #> separates antecedents from consequences.
- Consequences – a set of one or more patterns with a semi-colon in between.

Example 6

action(name=purchase,object=WEAPONS,buyer=BUYER,seller=SELLER);
object(name=BUYER,type=person);
object(name=SELLER,type=person);
object(name=WEAPONS,type=weapons)
#>
relation(type=possession,owner=BUYER,object=WEAPONS)

The previous rule reads as follows “If a person has bought weapons from another person then that person has possession of those weapons”

2.1.4 Programming Language

The programming language chosen for the implementation was C# with Microsoft Visual Studio.NET. There were two main reasons for that.

- C# is an object oriented language and therefore suited to the object oriented approach.
- C# has a EASYWIG programming environment and interface building that alleviates the task of building an interface as well as excellent debugging capabilities.

2.2 Inference Engine

The main efficiency and scalability bottleneck, in this type of application, is the inference engine. Typically as the number of rules and facts grows, it becomes increasingly harder to process information with efficiency. In order to address these issues and to guarantee optimal performance a modified version of the Rete algorithm was implemented as the base of the inference engine.

2.2.1 The Rete Algorithm

The Rete algorithm [F82] was created by Charles L. Forgy in the late 70's and it's considered one of the most efficient algorithms for rule pattern matching.

Rete [GI03] is intended to improve the speed of forward-chained rule systems by limiting the effort required to recompute the conflict set after a rule is fired. Its drawback is that it has high memory space requirements. It takes advantage of two empirical observations:

- *Temporal Redundancy*: The firing of a rule usually changes only a few facts, and only a few rules are affected by each of those changes.
- *Structural Similarity*: The same pattern often appears in the left-hand side of more than one rule.

The Rete algorithm uses a rooted acyclic directed graph where the nodes, with the exception of the root, represent patterns, and paths from the root to the leaves represent left-hand sides of rules. At each node is stored information about the facts satisfied by the patterns of the nodes in the paths from the root up to and including this node. This information is a relation representing the possible values of the variables occurring in the patterns in the path.

There are two main node types. The one input nodes, that primarily check for non variable information and the two input nodes that unifies different paths.

The Rete algorithm keeps up to date the information associated with the two input nodes in the graph. When a fact is added or removed from the working

memory, a token representing that fact and operation is entered at the root of the graph and propagated to its leaves modifying as appropriate the information associated with the nodes.

2.2.2 Conflict Set Resolution

Rete creates, as the output, a conflict set comprised of the set of rules that can be fired and the instantiations that satisfy all the patterns in the antecedent. The task of actually creating the Consequences of those rules is achieved *a posteriori* in the conflict set resolution step.

For each instantiation in the conflict set we must create the consequent of the rule that is fire. One or more facts are created and inserted again in Rete, so that propagation is achieved and the forward chaining mechanism proceeds. The inference engine stops when there are no more unprocessed facts to run trough Rete.

2.2.3 Modifications to the Rete algorithm

2.2.3.1 Practical Shortcomings

The original Rete algorithm was developed for the OPS5 languages and that presents some limitations.

The OPS5 language assumes that each working element has the same number or arguments. The ones that don't have an explicit value exist with null values. This forces the definition of all arguments a priori, which is not convenient if the goal is to have an open-domain algorithm. Although there is a domain definition, the system is prepared to deal with any domain definition using the same inference engine.

Similarly, OPS5 does not distinguish between predicate types; rather it has only working memory elements. For an object oriented approach we must be able to define objects and properties and benefit greatly from the possibility of defining explicit actions and relations, for there is evidence that there can be a relation between these facts and existing verbs in the analyzed text.

2.2.3.2 Algorithm Modifications

The working memory elements described in the original algorithm correspond to the facts described above. There are no restrictions on the number of arguments a fact can have, it is all taken care in the domain description rules.

The patterns in the original algorithm corresponded to parts of the working elements. A partial matching between the pattern and the working memory element was all that was necessary for an unification to succeed. In the modified version, the patterns are considered to be generalizations of the facts and therefore only match the complete fact.

In the original algorithm, the universal quantifier was not specifically adressed, that is, whenever we wanted to refer to “all X” it was not specified how this was handled at instantiation time. In the modified version the universal quantifier is treated as an exception of a two input node which takes one input only as the left input. Although it is still not the most elegant solution, enables the use of the universal quantifier in the form of a single variable.

For example the pattern *object(name=NAME,type=person)* will match all the objects of the type person in the working memory.

2.2.4 RETE Tree example

The following example shows a typical rule of the system and the rule internal representation in the inference engine.

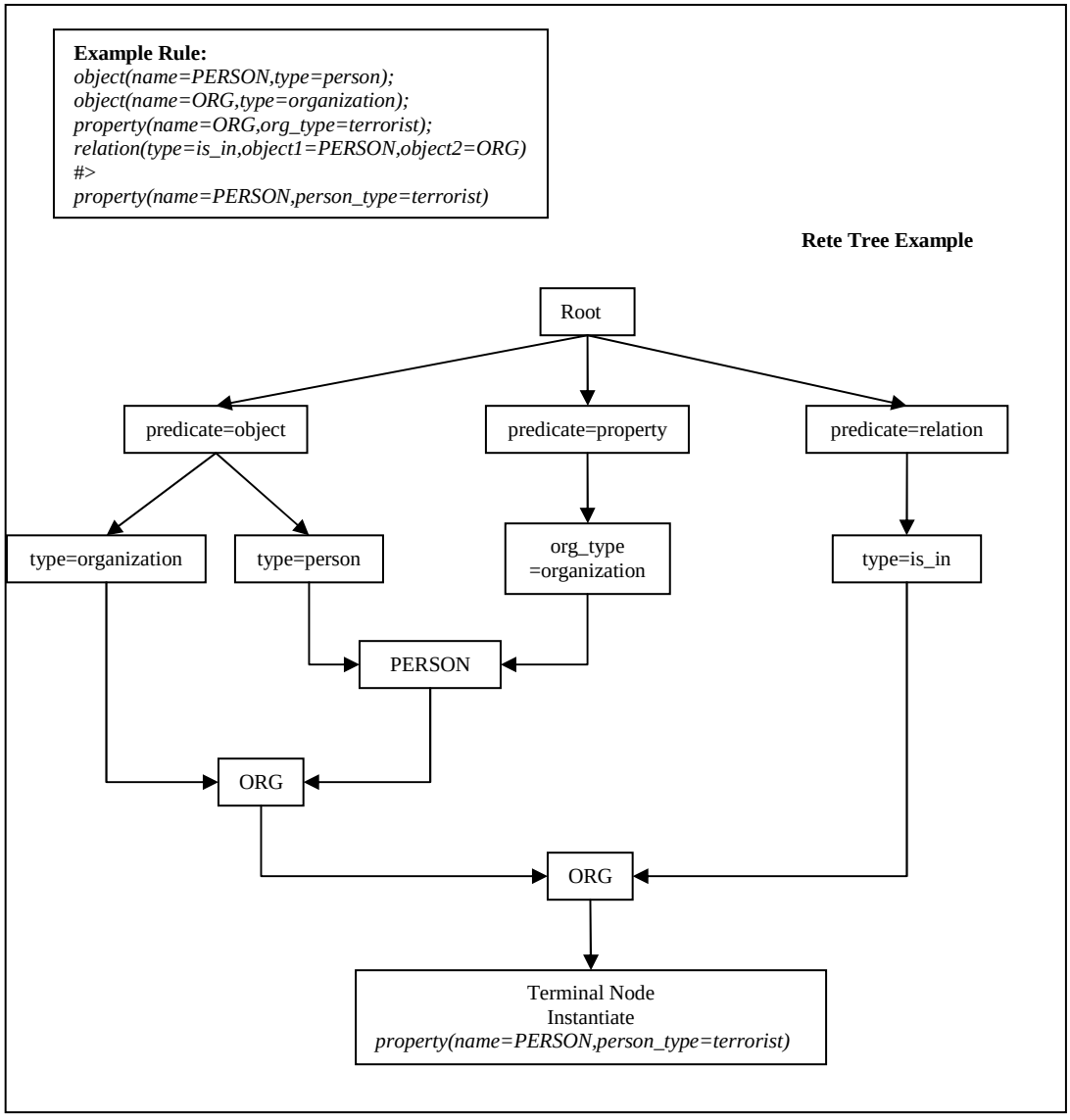

Fig 3: Example of Rete Tree

As we can see form the example above, the propagation of the tokens is done in parallel and when all the antecedents are fulfilled the rule is fired with an instantiation of the consequent.

2.3 Architecture and Implementation

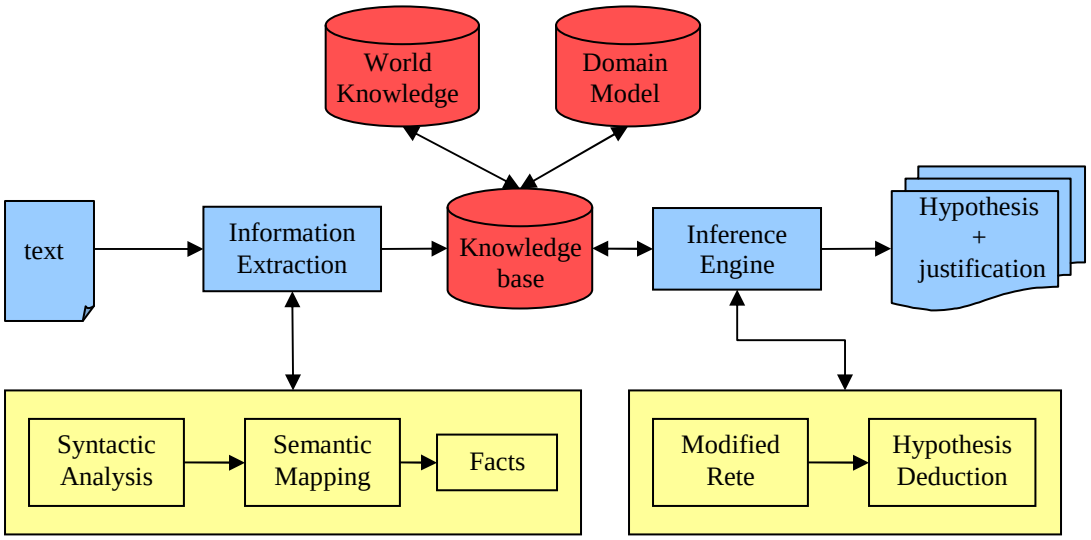

Fig 4: Architecture Implementation

2.3.1 Extracted Facts

This implementation presumes the existence of an existing tool for semantic extraction. Since at this time there is none that is ready from integration I bypassed that process and for all the examples created the simulated output of that hypothetical tool by hand. This consists mainly in the creation of two files that server the purpose.

- Facts file
  - List of the extracted facts for a given example in the following form : [document_number]#fact.

    Ex: 1#object(name=john,type=person)
- Documents File
  - List of the documents pertinent to the example in the following form: [document_number]#[document_name]#[document_date].

Ex: 1#doc1.txt#04/12/2003

These files must be supplied at the time of the server activation.

2.3.2 Server-Client Architecture

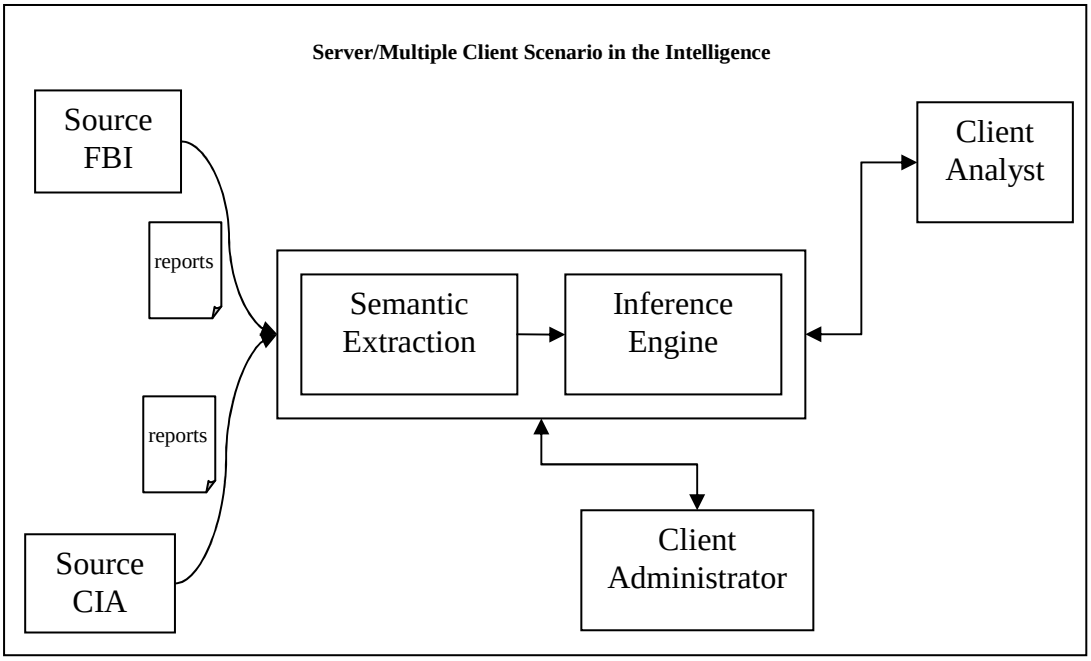

Fig 5: Server-Client Architecture

The main advantage of the server-client architecture is to enable the separation of module roles and the creation of a modular approach. In the example above we have multiple sources of information that keep updating the system with reports, the server that consumes that information, processes the reports and infers the appropriate the predictions, an analyst client that works with the information and the system predictions and the client administrator that assures the maintenance of the system.

The server maintains consistency of the information independently of the number of clients that are connected which assures that at anytime that the analyst requires information, it will be up to date and coherent.

The server-client architecture was implemented using the remoting procedures in C#.net. Remoting uses delegates to create instances of the server function in the clients, creating a transparent layer of communication using XML between the server and the client. From the client point of view the objects become local objects and can be used as so, yet the objects are being remotely drawn from the server in a pull fashion that instantiates the server version of the object in the client. The benefits of this approach include the automatic management of the communication protocol, concurrency issue handling, and multiple connection availability with state consistency between the connections in a straightforward way. For more information on remoting, please consult [IR02].

2.3.3 Server Implementation

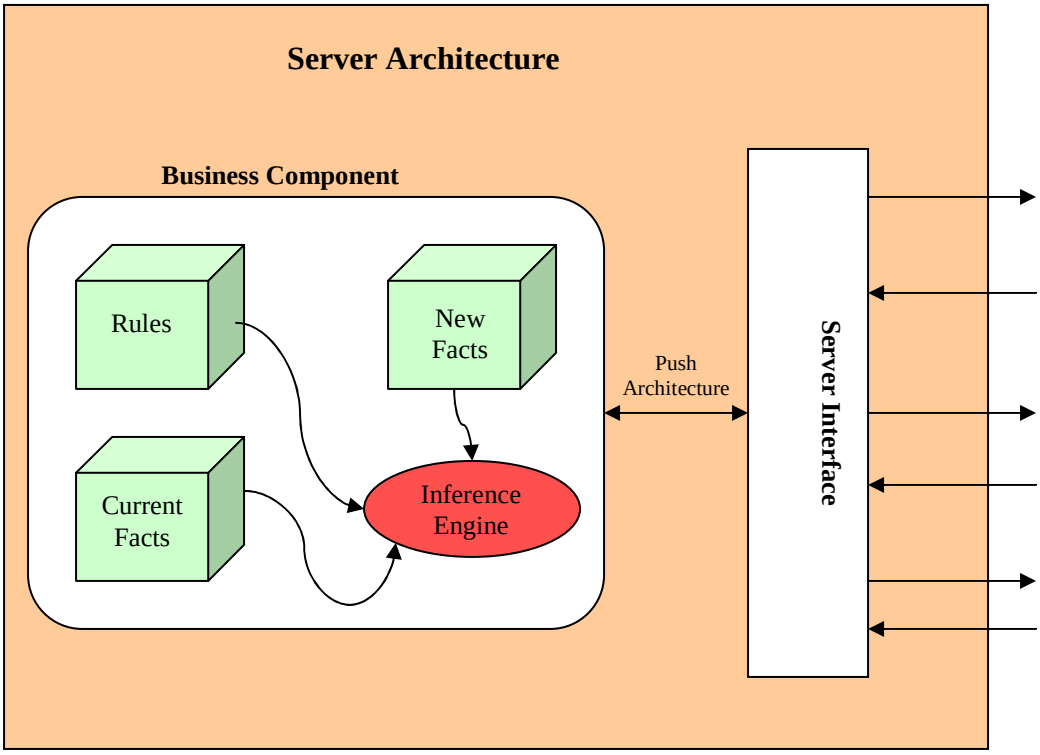

Fig. 6: Server diagram

Consisting of three primary classes, the server implementation includes the connection server, the inference engine and the modified Rete algorithm.

- Server Class
  - The server class implements the server functionally. Accessible we can find several commands for interaction with the server. It initiates an inferences engine and maintains the server status. Each new connection gets the same fact database and rule database.
- Brain Class
  - The brain class is the inference engine. In functions as a wrapper around the Rete algorithm, building the Rete tree, processing the new facts and solving the conflict set produced by Rete.

- Rete Class
    - The actual matching algorithm. Processes the new facts against the Rete tree and find the rules whose antecedents are fulfilled. Implements the function that builds the tree and processes the nodes in a recursive fashion. It may consume a big chunk of memory.

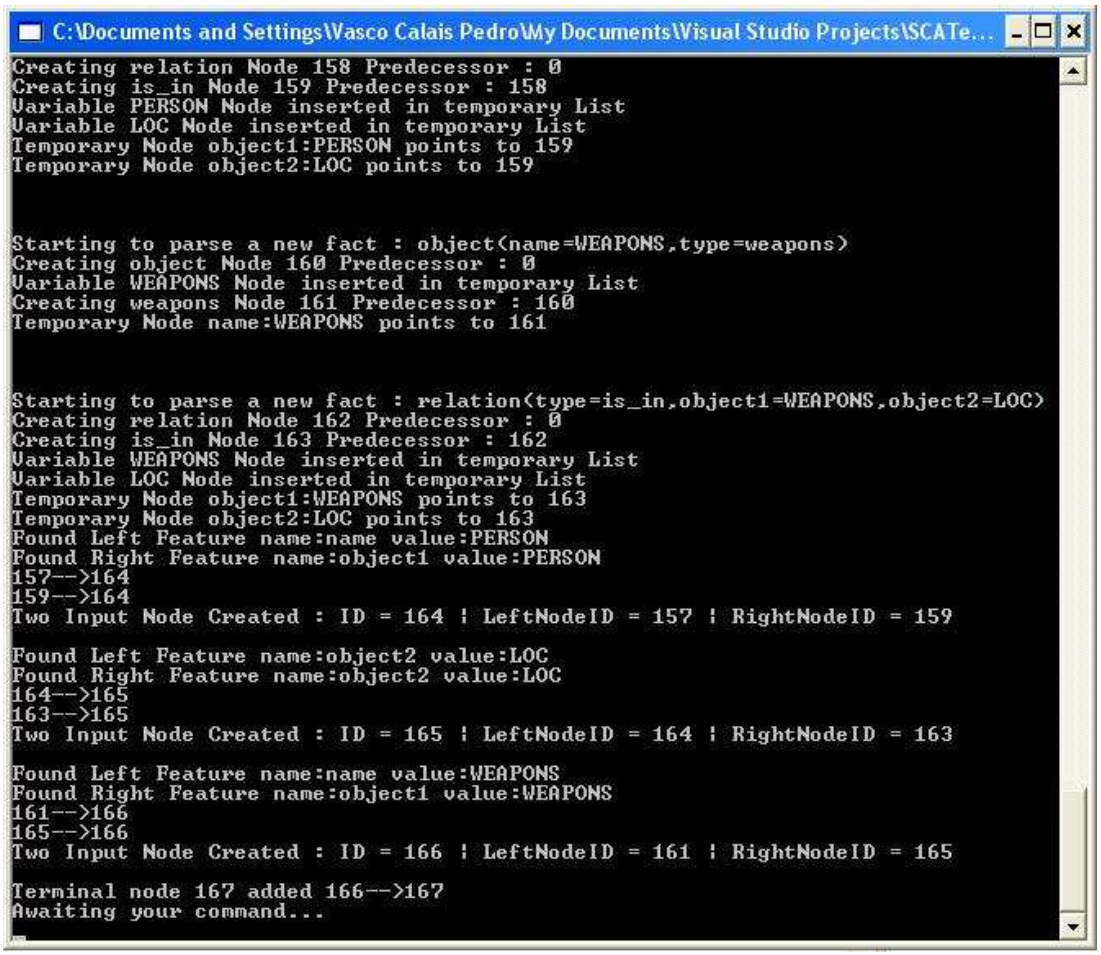

Fig 7: Server initialization after the Rete tree is built

2.3.4 Client Implementation

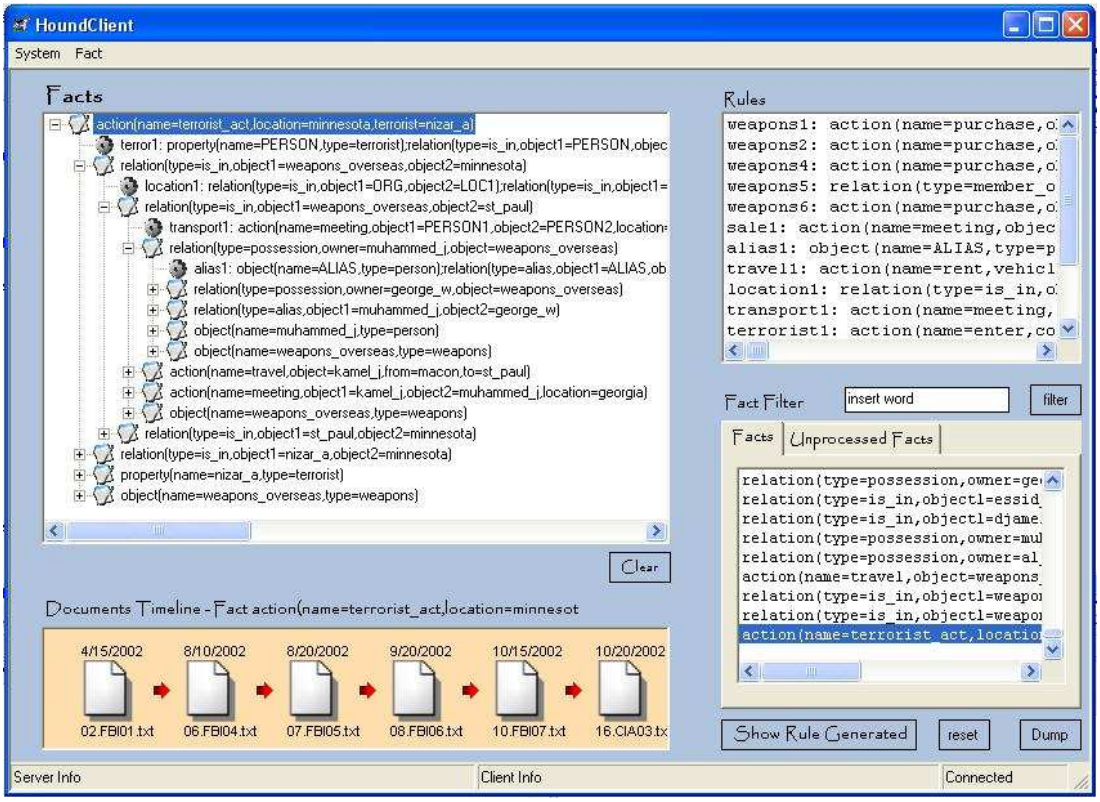

Fig 8: Client example - Trace of fact and document timeline.

The client was implemented using the MS Visual Studio.net and C# language. Goal wise, the concept behind the demonstration client was to convene as much information as possible in one screen, thus demonstrating the capabilities of the system.

We can obverse the information in Fig 7. On the top left we can see the tree that demonstrates the justification of the facts. This consists in the trace path on which rules were used to derive the conclusion and the facts that instantiated it.

On the bottom left corner we can see the timeline of the events. This timeline shows the ordered set of documents on which the facts are based.

The top right shows the rules existing in the system. Using the right click button and selecting “view rule” we can see a more detail view of the rule.

On the bottom right we can see the current facts of the system, filter the facts according to a certain word, and show only the facts that were generated by the system.

2.4 Application Example

The intelligence example constituted the most complex experiment using the implemented architecture.

2.4.1 Corpus

The document corpus is comprised of 23 reports from several intelligence agencies. The reports address events between February and October of 2002 and describe the transactions and movements of certain individuals. The agencies that supply those reports are FBI, CIA, NSA, INS and CID.

> *"CIA Report 01 [20 September, 2002]: the person identified as John H. in FBI Report 02 was seen in London, UK on 12 June, 2001 in company with Lofti Raissi who may have helped train the four hijacker pilots involved in the Sept. 11, 2001 terrorist incidents. It seems that the name John H. is an alias for Omar K, a person known to be associated with AL-Qaeda."*

Fig. 9 - report example

The reports were classified using the BBN Identifinder [BBN00], were manually analyzed and the semantic facts extracted. Extracted facts were kept to the minimum degree of simplicity possible. The Objects extracted from the reports were the ones tagged by the Identifinder.

A document database was created with the ID, name and date of each report. This database is to be used as reference for the facts.

2.4.2 Domain Description Set

The development of the domain description set consisted in the creation of 13 rules that intend to simulate the descriptive intuitive notions of the analyst regarding

the intelligence domain. They are in no way intended to be comprehensive, but rather specific and focused on the phenomena observable in the corpus. The purpose of the experiment is to demonstrate the construction and usability of domain specific models and thus exemplify the application created.

The rules were created with the following format:

- [rule_identifier]#Rule
- Example :

```
terrorist3#
action(name=enter,country=COUNTRY,location=LOC,person=PERSON,document=DOC)
#>relation(type=is_in,object1=PERSON,object2=LOC)
```

2.4.3 Results

Given the extracted facts and the domain description, Rete was able to predict some interesting facts. The examples use the reasoning justification provided by the client. Any of these examples also provide a timeline of the documents upon which these conclusions were based.

2.4.3.1 Possession of weapons

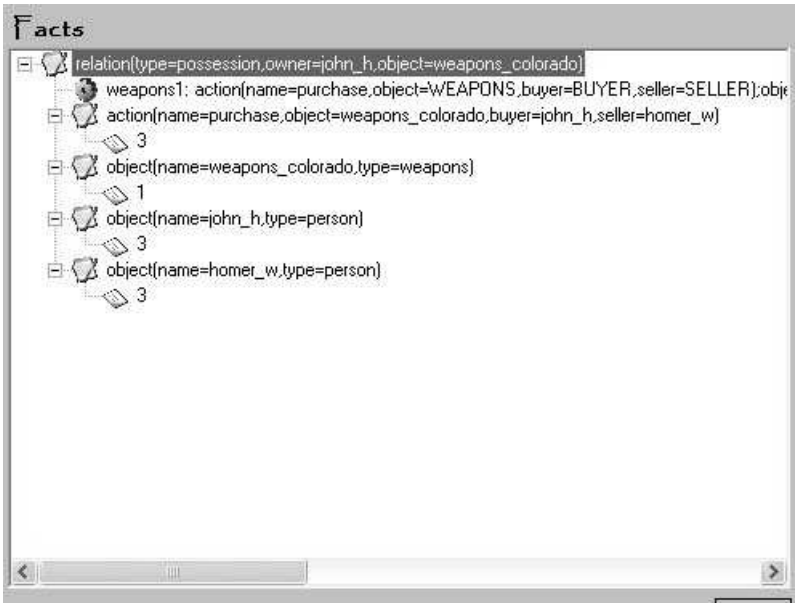

Fig. 10: Justification of weapons possession.

In the previous example we can observe a prediction by HOUND of the transaction between two people and the consequent possession of weapons by the buyer.

2.4.3.2 Possession of weapons by a terrorist organization.

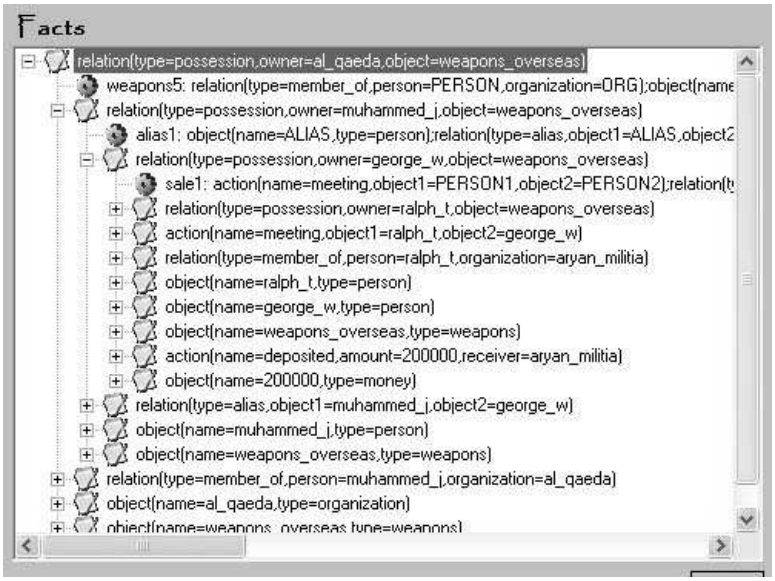

Fig. 11: Justification of possession of weapons by a terrorist organization

This example shows the possession of weapons by a terrorist organization based on the fact that one of its members has possession of those weapons.

2.4.3.3 Identification of possible terrorists.

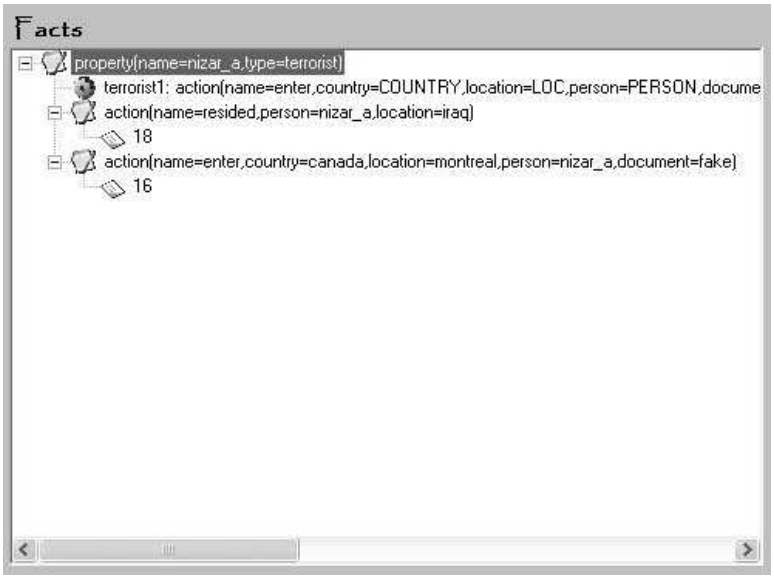

Fig. 12: Justification of the identification of terrorist nizar_a.

Here we can observe the possible identification of a terrorist by reasoning over the fact the person entered the United States with a fake document after having resided in Iraq.

2.4.3.4 Possible terrorist attempts.

Fig. 13: Justification of a possible terrorist attempt.

Finally, the prediction of a possible terrorist attempt appears based on the presence of weapons and a terrorist in the same place.

3 Evaluation

3.1 Performance

Although several aspects of performance can be mentioned when discussing HOUND, some are naturally more important than others. Moore's Law shows us that considerations regarding aspects that depend on the hardware are of less concern than those regarding algorithm limitations. Therefore I will focus the performance aspects on the Rete performance aspects, since those are the more relevant.

Rete was designed for systems in which the working memory changes relatively slowly. Given a scenario where number of fact input stream where enormous, the algorithm would slow down considerably. Furthermore, given that the working memory elements are stored at the node level and that the algorithm is implemented recursively, the memory constraints can become burdensome.

The following table represents the original performance of the algorithm, which still maintains [CF82].

Complexity Measure	**Best Case**	**Worst Case**
Effect of working memory size on number of tokens	O(1)	$O(W^C)$
Effect of production memory size on number of nodes	O(P)	O(P)
Effect of production memory size on number of tokens	O(1)	O(P)
Effect of working memory size on time for one firing	O(1)	$O(W^{2c-1})$
Effect of production memory size on time for one firing	$O(\log_2 P)$	O(P)
C is the number of patterns in a production **P** is the number of productions in production memory **W** is the number of elements in working memory		

Table 1: Space and Time complexity of Rete

3.2 Shortcomings

Regarding shortcomings of HOUND, there are two main areas that must be considered, the shortcomings in the algorithm itself and those that result from my implementation.

3.2.1 Rete Shortcomings

3.2.1.1 Assumption of stability of the production memory

The efficiency of the Inference engine is constrained by the assumption of relative stability in the production memory. If the state of the world changes too quickly, node level matching will become less efficient.

The two input nodes store the token information on the node itself.

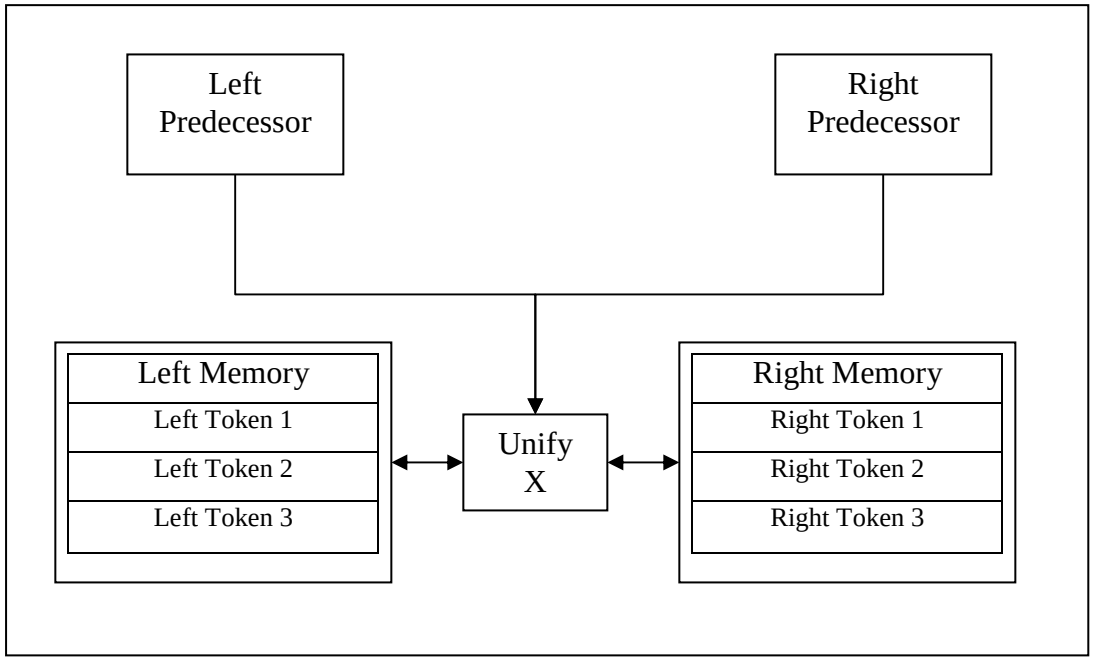

Fig. 13: Accumulation of tokens in the node memory

Each token that arrives at a two input node is stored either in the left or the right memories of the node, depending which predecessor it came from. The

matching process is achieved by individually matched with each of the other memory elements in order to find all unifications. If one such unification is found then the two tokens are merged and the resulting token sent to the next node.

The growth of the left or right memories poses a possible bottleneck of the algorithm, since most of the tokens to be matched typically will not unify with the new token, which generate useless work by the algorithm.

The algorithm would benefit greatly from a more efficient matching algorithm at the node level.

3.2.1.2 Double variables in facts

A problem that seems not to be specified in the original Rete implementation is derived from the fact that Rete uses an acyclic directed graph as a base.

The problem consists in the fact that, if two paths have more than one variable in common and the unification is made the two input node level, once they are united, there are no more two inputs.

The following figure illustrates the problem.

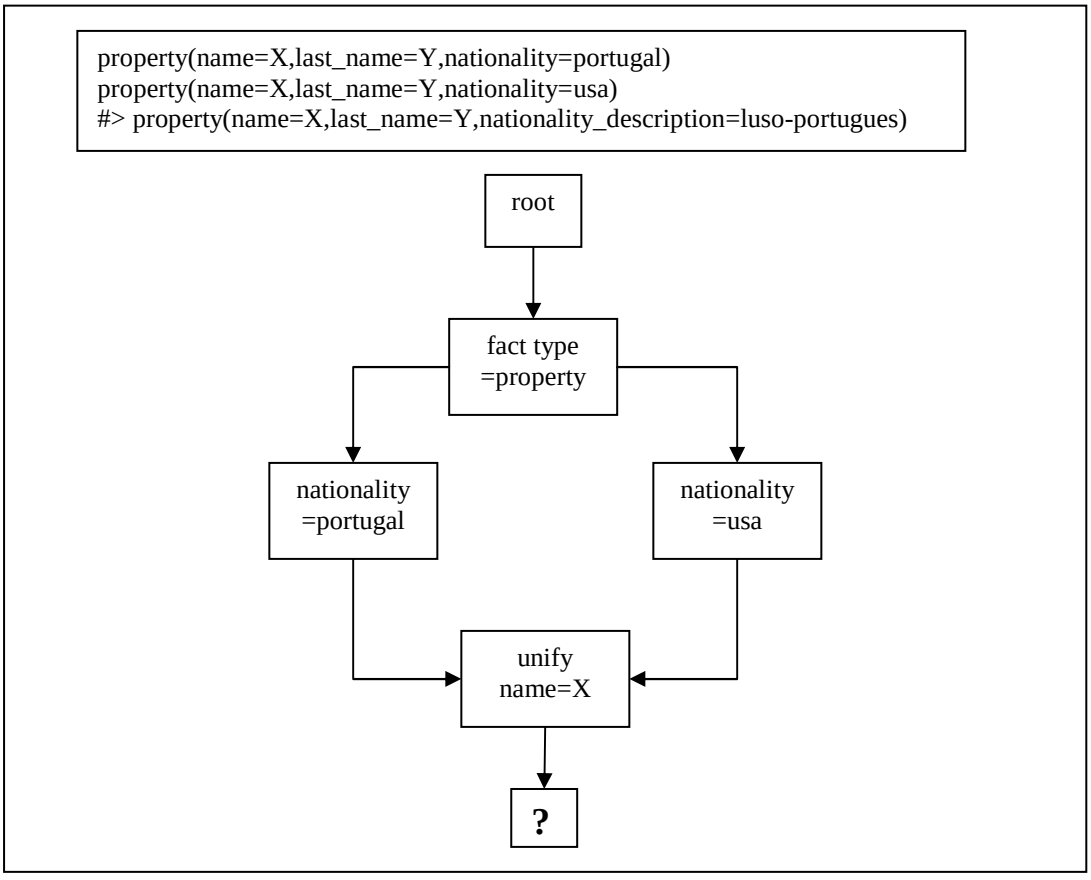

Fig. 14: Double Variable Problem

Once we unify the paths in the two input node that unifies the *name=X* we have a problem unifying the *last_name=Y*.

3.2.1.3 Single variable

The universal quantifier is expressed trough a single variable in the pattern. The problem comes once more from the fact that the unification of paths exists at the two input nodes. When we have a single variable the unification of paths becomes compromised.

Example:

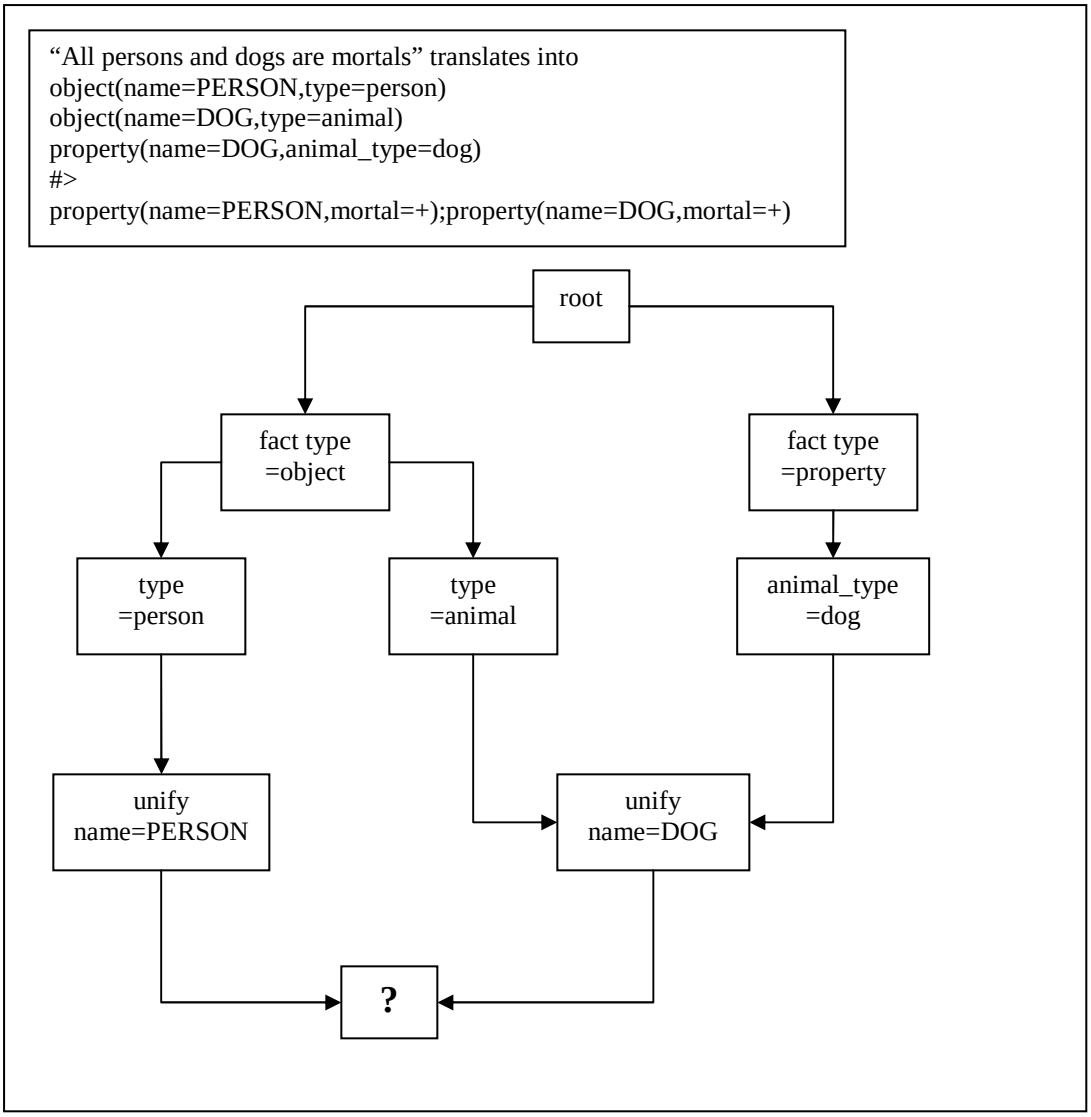

Fig. 15: Example of the universal quantifier problem

The algorithm does not supply the solution for the problem exposed above, the unification of the paths depends of the connection of the antecedents between themselves in the form of binding variables.

The inexistence of such variables between two patterns interrupts the path unification.

4 Future Work

The completion of this thesis is only the beginning of a deeper and more comprehensive work on probabilistic semantic reasoning. The work done lays the foundation for the development of such a system and should be considered in any way as definitive in this area. This was but an exercise of viability and demonstration of base ideas and the scope of the work does not permit the deep development of the ideas presented here. That scope is more suited for a PhD. thesis. There is much work to be done.

4.1 Iterative Forward Chaining

The current implementation incorporates the predictions into the fact set and executes the forward chaining reasoning until there are no unprocessed facts left.

Selective forward chaining will be implemented to allow the user to select which results seem valid and more promising, thus guiding the reasoning process. At least two reasoning modes are necessary to present a viable production system, the full forward chaining and the iterative forward chaining.

4.2 Push Architecture

The current architecture relies only on pull-type architecture to update the client fact database. Ideally the server would push the new facts onto the active listening client in a broadcast style.

This can be done by implementing delegates that act as agents for the broadcast of new messages. It was necessary for this thesis to implement such thing, but any production system should contain this in order to be efficient. There is no pre determined interval to update the facts and having multiple client request updates every 10 seconds is not a viable option.

Implementing a push architecture type will permit the system to work in real time, be more efficient and save network resources.

4.3 Node Set Completion

In the original Rete algorithm, two more node types were suggested on order to form a complete node set. The current node set does not permit two important operations, negated patterns and Intra-pattern variable binding.

4.3.1 Negated Patterns

A negated pattern is a pattern preceded by a minus sign. It is only true if no fact in the working memory unifies with that pattern.

Negated pattern example:

object(name=PERSON,type=person);

-property(name=PERSON, person_type=terrorist)

#>property(name=PERSON,person_type=good_citizen)

Gloss: All persons that are not terrorists are good citizens.

Negated patterns are essential for any production system. Although they were not necessary in the examples demonstrated, they are part of the essential node set.

The negated patterns would be implemented by creating a new node type that only allows the token to propagate if no fact unifies with the token in the node's internal memory.

4.3.2 Intra Pattern Variable Binding

The other node necessary to complete a fully functional node set addresses the unification within the patterns. For example the pattern

Property(name=NAME,last_name=NAME)

This pattern refers to persons with the same first and last name. The current two input node only unifies elements in different facts. Thus we need to create another node type that unifies variables in the same fact.

4.4 Probabilistic Rete

The probabilistic version of Rete will constitute the focus the future work.

The world is rarely described in terms of black and white and

The uncertainty present in real world data manifests itself, within the HOUND context in the following ways

- Semantic Extraction Reliability
    - Given that it will be always impossible to guarantee absolute accuracy in the semantic extraction, and that in many cases the semantic representation will convey a different meaning than the original text, there must be a measure of uncertainty regarding the semantic extraction process.
- Information Uncertainty
    - A measure of uncertainty referring to the uncertain that the actual information contains must be created. Information is conveyed using a different number of verbs and structures that refer to the certainty of the speaker regarding the events and this must be taken into account. A sentence like "*John was seen renting a truck*" carries more certainty that "*John might have rented a truck*". The reflection of that difference must exist in a probabilistic model.
- Source Credibility
    - The credibility of an information source is as important as the information it contains. We must take into account the quality of our sources when we predict based on them.

- Weighted Production System
    - Any real world situation has factors that weight more than others. When we start our car, normally the fact that the car has fuel weights more on probability that the car will start than the condition of the weather, but if it snowing a lot the weather must be taken into account. The patterns in the productions must have a weight associated with them in order to take this fact into account.
- Production Intuitive Probability
    - The production itself must have an uncertainty measure as to the probability of the consequence happening when the antecedent is fulfilled. Not always turning the key will make the engine start. The productions must include the intuitive knowledge of those that know the domain and harness that knowledge to achieve realistic results.

Thus far Rete has been used primarily as a pattern matching algorithm in a binary system where rules are fired when the antecedents are satisfied. For future work I intend to develop a probabilistic framework integrated with the Rete algorithm to capitalize on the scalability and efficiency of Rete in an uncertain environment.

In a probabilistic framework not all predictions are considered, but rather those that are above a threshold of certainty. This will allow filtering

5 References

[AH92] Andersen, P.M., Hayes, P.J. Huettner, A.K., Nirenburg, I.B., Shmandt, L.M., & Weinstein, S.P. *Automatic Extraction of Facts from Press Releases to Generate New Stories*. In: Proceedings of the Third Conference on Applied Natural Language Processing. 1992, 170-177.

[AV02] AVENTINUS : Advanced Information System for Multinational Drug Enforcement. http://www.dcs.shef.ac.uk/research/groups/nlp/funded/aventinus.html. Site visited 10/15/2002

[BBN00] BBN Technologies, 2000. Identifinder User Manual

[CA70] Carbonell, J.R. (1970). *AI in CAI: An Artificial Intelligence Approach to Computer-assisted Instruction*. IEEE Transactions on Man-Machine Systems. Vol. 11, pp.190-202.

[CL96] Cowie, J., & LEHNERT, W. *Information Extraction*. Communications of the ACM,39(1), 1996, 80-91.

[D82] Dejong, G. *An overview of the FRUMP System*. In: LEHNERT, W., & RINGLE, M.H. (eds), Strategies for Natural Language Processing. Lawrence Erlbaum, 1982, 149-176.

[DG98] D. Calvanese, G. De Giacomo, M. Lenzerini, D. Nardi, R. Rosati, Description Logic Framework for Information Integration, Proc. Of the 6th Int. Conf. On the Principles of Knowledge Representation and Reasoning (KR'98), pages 2-13, 1998.

[F82] Forgy, C.L.: Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem Artificial Intelligence, 19(1982) 17-37

[FIS90] Fisher, K.M., 1990. *Semantic Networking: The new kid on the block.* Journal of Research in Science Teaching, 27(10): 1001-1018

[FR94] Frank, A. and U. Reyle (1994), *Principle based semantics for HPSG*, Arbeitspapiere des Sonderforschungsbereichs 340, University of Stuttgart.

[GI03] Giorgio Ingargiola

http://www.cis.temple.edu/~ingargio/cis587/readings/rete.html

[GH97] R. Gaizauskas and K. Humphreys, *Using A Semantic Network for Information Extraction* - Technical Report CS-97-03 Department of Computer Science, University of Sheffield, 1997.

[GH00] K. Humphreys, R. Gaizauskas and H. Cunningham, *LaSIE Technical Specifications*, Department of Computer Science, University of Sheffield, Technical Report CS-00-09, 2000.

[IR02] Ingo Rammer, *Advanced .net Remoting*, Apress publications, ISBN 1-59059-025-2.

[VHKN03] Benjamin Van Durme, Yifen Huang, Anna Kupsc, Eric Nyberg, *Towards Light Semantic Processing for Question Answering*. HLT-NAACL Workshop 2003.

6 Appendix a – Intelligence Corpus

Army CID Report 01 [15 July, 2002]: Report of theft of weapons from a military installation in Georgia on or about 12 July 2002. Missing are about 500 rounds of M-14 ammunition and an undetermined number of obsolete Manpads. [This theft is still under investigation].

Army CID Report 02 [21 February, 2002]: report of theft of small arms ammunition, four M-16 rifles, six grenade launchers, and six Manpads. This theft occurred at a military installation in Colorado. [This theft is still under investigation].

CIA Report 01 [20 September, 2002]: the person identified as John H. in FBI Report 02 was seen in London, UK on 12 June, 2001 in company with Lofti Raissi who may have helped train the four hijacker pilots involved in the Sept. 11, 2001 terrorist incidents. It seems that the name John H. is an alias for Omar K, a person known to be associated with AL-Qaeda.

CIA Report 02 [15 October, 2002 from a source, said to be credible, in Jordan]. A man named Majed H., using a faked Jordanian passport and travel visa, traveled from Aman, Jordan to Chicago, Illinois on 12 July, 2002. Majed H. is now known to have resided in Afghanistan for two years [1996-1997] and has been identified as a member of Al-Qaeda.

CIA Report 03 [20 October, 2002 from a source in Egypt said to be very credible]. Essid D. and Djamel K, both Syrian nationals, entered the USA illegally in Los Angeles on 3 August 2002 using false Egyptian passports and travel visas. Both are said to have spent at least a year in Afghanistan.

CIA Report 04 [22 October, 2002] Abu H., who was released from custody after the Sept. 11 incidents and whose fingerprints were found in the U-Haul truck rented by Arnold C. [see FBI Report 07] holds an Egyptian passport. He is now known to have spent six months in Afghanistan in the summer of 1999.

CIA Report 05 [22 October, 2002 from a credible source in Paris, France]. Nizar A., an Algerian residing in Paris, entered Montreal, Canada on 16 July 2002 using a French passport. He is known to have traveled to Winnipeg, Canada. Nizar A. is now known to have spent six months and may have traveled to Iraq.

FBI Report 01 [15 April, 2002] based on a report from a source inside God's Aryan militia, Atlanta, Georgia. Militia group's purchase of surplus weapons from an overseas arms supplier on or about 10 April, 2002. Purchase believed to include automatic weapons of various sorts and an unknown number of Manpads.

FBI Report 02 [8 July 2002]: Recorded conversation in Denver, Colorado on 1 July, 2002 between Homer W. of the Aryan Brotherhood of Colorado and a person who identified himself as John H. This conversation involved the sale of weapons of an unspecified nature. Homer W. is an Army sergeant being investigated in connection with the theft of weapons reported in Army CID Report 02.

FBI Report 03 [16 July, 2002]: Report of sums of money totaling $150 000 deposited in Denver bank accounts thought to be associated with the Aryan Brotherhood of Colorado.

FBI Report 04 [10 August, 2002]: Video-taped report of a meeting in Atlanta Georgia on 8 August, 2002 involving Ralph T, a member of God's Aryan Militia and a person identified as George W. Ralph T has been under surveillance in connection with the arms purchase noted in FBI Report 01.

FBI Report 05 [20 August, 2002]: Report of sums of money, totaling over $200,000 deposited in Atlanta bank accounts thought to be associated with members of God's Aryan Militia in Atlanta.

FBI Report 06 [15 September, 2002]: The person identified as George W. in FBI 03 may be Muhammed J., a Saudi national who is believed to be a member of Al-Qaeda.

FBI Report 07 [15 October, 2002]. Based on information obtained on 12 October, 2002 concerning Ryder & U-Haul truck rentals by persons who could be identified as foreign nationals:

- On October 6, 2002 a small panel truck was rented at a U-Haul agency in Boulder, Colorado by a man who identified himself by mean of a California drivers' license. The name on the drivers' license was Arnold C. whose residence was listed in Los Angeles, California. The rental was for a one-way trip; the truck was returned to a U-Haul agency in Los Angeles on 9 October, 2002. Mr. C. paid for the truck in cash.
- On October 8, 2002 a small panel truck was rented at a Ryder agency in Macon, Georgia by a man who identified himself by a Minnesota driver's license. The name on the license was Kamel J, whose address was listed in St. Paul, Minnesota. This truck was

returned to a Ryder agency in St. Paul in St. Paul on 11 October, 2002.

- On October 15, 2002 a U-Haul panel truck was rented in Nashville Tennessee by a man who identified himself Harold W. whose address, given on his Tennessee drivers' license, is in St. Louis, Missouri. This truck was to be turned in to a U-Haul agency in St. Louis and was turned in on October, 2002.

FBI Report 08 [18 October, 2002]. the driver's licenses provided by Arnold C. and Kamel J. to the truck rental agencies in Boulder, Colorado and Macon, Georgia are not authentic. The addresses shown on these licenses do not exist.

FBI Report 09 [19 October, 2002]. Examination of the U-Haul truck, rented by Mohammed H, and returned to St. Louis, Missouri was found to have traces of semtex explosives in the cab of the truck.

FBI Report 10 [19 October, 2002]. a truck was hijacked outside of Fargo, North Dakota. the truck was carrying several tons of industrial explosives. The truck's location is presently unknown.

FBI Report 11 [20 October, 2002]. Fingerprints observed in the cab of the U-Haul truck rented by Arnold C. on October 6, 2002 match those of Abu H. who was taken into custody shortly after Sept. 11, 2002 and then released. The U-haul truck returned to Los Angeles was not subsequently rented after it was returned by Arnold C.

FBI Report 12 [26 October, 2002]. A computer in a public library in Atlanta, Georgia recorded several connections on 23 and 24 October, 2002 to the web site in NSA Report 01. Connections to this same web site were found in the library at Colorado State University on 23 and 24 October, 2002.

FBI Report 13 [27 October, 2002]. The person identified as Muhammed J. in the FBI Report 06 and the person identified as Kamel J. in FBI Report 07 were photographed together in a bank in Atlanta, Georgia on October 6, 2002.

FBI Report 14 [27 October, 2002]. On 14 July, 2002, a man identified as Majed H. rented a car at Hertz car rental agency in Chicago, Illinois and dropped off the rented car in Minneapolis, Minnesota on 17 July, 2002.

INS Report 01 [25 October, 2002]. A man named Nizar A., holding a valid French passport, was allowed entry into the USA on 25 July, 2002 at International Falls, Minnesota.

NSA Report 01 [25 October, 2002]. On 24 October, 2002, embedded in a figure on a web site, allegedly associated with an Islamic charitable organization, was a coded number believed to be: 021127. It is likely that this number is the date: 27 November, 2002.

7 Appendix b – Document List

1#01.CID02.txt#02/21/2002
2#02.FBI01.txt#04/15/2002
3#03.FBI02.txt#07/08/2002
4#04.CID01.txt#07/15/2002
5#05.FBI03.txt#07/16/2002
6#06.FBI04.txt#08/10/2002
7#07.FBI05.txt#08/20/2002
8#08.FBI06.txt#09/20/2002
9#09.CIA01.txt#09/20/2002
10#10.FBI07.txt#10/15/2002
11#11.CIA02.txt#10/15/2002
12#12.FBI08.txt#10/18/2002
13#13.FBI09.txt#10/19/2002
14#14.FBI10.txt#10/19/2002
15#15.FBI11.txt#10/20/2002
16#16.CIA03.txt#10/20/2002
17#17.CIA04.txt#10/22/2002
18#18.CIA05.txt#10/22/2002
19#19.INS01.txt#10/25/2002
20#20.NSA01.txt#10/25/2002
21#21.FBI12.txt#10/26/2002
22#22.FBI13.txt#10/27/2002
23#23.FBI14.txt#10/27/2002

8 Appendix c –Extracted Semantic Facts

```
//facts from the intelligence example
//CID02
1#object(name=colorado,type=location)
1#object(name=weapons_colorado,type=weapons)
1#property(name=weapons_colorado,location=colorado)
1#action(name=theft,object=weapons_colorado,location=colorado)
//FBI01
2#object(name=aryan_militia,type=organization)
2#object(name=atlanta,type=location)
2#object(name=georgia,type=location)
2#relation(type=is_in,object1=atlanta,object2=georgia)
2#relation(type=is_in,object1=aryan_militia,object2=atlanta)
2#object(name=weapons_overseas,type=weapons)
2#action(name=purchase,object=weapons_overseas,buyer=aryan_militia)
//FBI02
3#object(name=denver,type=location)
3#relation(type=is_in,object1=denver,object2=colorado)
3#object(name=homer_w,type=person)
3#object(name=aryan_brotherhood,type=organization)
3#relation(type=is_in,object1=aryan_brotherhood,object2=colorado)
3#relation(type=member_of,person=homer_w,organization=aryan_brotherh
ood)
3#object(name=john_h,type=person)
3#action(name=meeting,object1=john_h,object2=homer_w)
3#action(name=purchase,object=weapons_colorado,buyer=john_h,seller=h
omer_w)
3#object(name=army,type=organization)
3#relation(type=member_of,person=homer_w,organization=army)
//this would be a key aspect to show of the probability scenario
3#action(type=steal,object=weapons_colorado,thief=homer_w)
//CID01
4#object(name=weapons_georgia,type=weapons)
4#property(name=weapons_georgia,location=georgia)
4#action(name=theft,object=weapons_georgia,location=georgia)
//FBI03
5#object(name=150000,type=money)
5#action(name=deposited,amount=150000,receiver=aryan_brotherhood)
//FBI04
6#object(name=ralph_t,type=person)
6#relation(type=member_of,person=ralph_t,organization=aryan_militia)
6#object(name=george_w,type=person)
6#action(name=meeting,object1=ralph_t,object2=george_w)
6#action(name=purchase,object=weapons_overseas,buyer=ralph_t)
//FBI05
7#object(name=200000,type=money)
7#action(name=deposited,amount=200000,receiver=aryan_militia)
//FBI06
8#object(name=muhammed_j,type=person)
8#relation(type=alias,object1=muhammed_j,object2=george_w)
```

```
8#object(name=al_qaeda,type=organization)
8#property(name=al_qaeda,org_type=terrorist)
8#relation(type=member_of,person=muhammed_j,organization=al_qaeda)
//CIA01
9#object(name=omar_k,type=person)
9#relation(type=alias,object1=omar_k,object2=john_h)
9#relation(type=member_of,person=john_h,organization=al_qaeda)
//FBI07
10#object(name=truck_boulder,type=vehicle)
10#object(name=arnold_c,type=person)
10#object(name=boulder,type=location)
10#relation(type=is_in,object1=boulder,object2=colorado)
10#action(name=rent,vehicle=truck_boulder,person=arnold_c,location_f
rom=boulder)
10#action(name=return_vehicle,vehicle=truck_boulder,location=los_ang
eles)
10#object(name=los_angeles,type=location)
10#object(name=california,type=location)
10#relation(type=is_in,object1=los_angeles,object2=california)
//
10#object(name=macon,type=location)
10#relation(type=is_in,object1=macon,object2=georgia)
10#object(name=truck_macon,type=vehicle)
10#object(name=st_paul,type=location)
10#object(name=minnesota,type=location)
10#relation(type=is_in,object1=st_paul,object2=minnesota)
//
10#object(name=kamel_j,type=person)
10#action(name=rent,vehicle=truck_macon,person=kamel_j,location_from
=macon)
10#action(name=return_vehicle,vehicle=truck_macon,location=st_paul)
//
10#object(name=nashville,type=location)
10#object(name=tenessee,type=location)
10#relation(type=is_in,object1=nashville,object2=tenessee)
10#object(name=truck_nashville,type=vehicle)
10#object(name=harold_w,type=person)
10#object(name=st_louis,type=location)
10#object(name=missouri,type=location)
10#relation(type=is_in,object1=st_louis,object2=missouri)
10#action(name=rent,vehicle=truck_nashville,person=harold_w,location
_from=nashville)
10#action(name=return_vehicle,vehicle=truck_nashville,location=st_lo
uis)
//CIA02
11#object(name=majed_h,type=person)
11#relation(type=member_of,person=majed_h,organization=al_qaeda)
11#object(name=aman,type=location)
11#object(name=chicago,type=location)
11#action(name=travel,person=majed_h,from=aman,to=chicago)
//FBI08
12#property(name=arnold_c,type=drivers_license,status=fake)
```

```
12#property(name=kamel_j,type=drivers_license,status=fake)
//FBI09
13#object(name=mohammed_h,type=person)
13#object(name=unknown,type=location)
13#action(name=rent,vehicle=truck_3,person=mohammed_h,location_from=
unknown)
13#action(name=return_vehicle,vehicle=truck_3,location=st_louis)
13#property(name=truck_3,type=contains,value=explosives)
//FBI10
14#object(name=fargo,type=location)
14#object(name=north_dakota,type=location)
14#relation(type=is_in,object1=fargo,object2=north_dakota)
14#object(name=explosives_fargo,type=weapons)
14#action(name=theft,object=explosives_fargo,location=georgia)
//FBI11
15#object(name=abu_h,type=person)
15#action(name=fingerprints_found,location=truck_boulder,person=abu_
h)
//CIA03
16#object(name=essid_d,type=person)
16#object(name=djamel_k,type=person)
16#action(name=enter,country=us,location=los_angeles,person=essid_d,
document=fake)
16#action(name=enter,country=us,location=los_angeles,person=djamel_k
,document=fake)
16#object(name=afghanistan,type=location)
16#action(name=resided,person=essid_d,location=afghanistan)
16#action(name=resided,person=djamel_k,location=afghanistan)
//CIA04
17#action(name=resided,person=abu_h,location=afghanistan)
//CIA05
18#object(name=nizar_a,type=person)
18#object(name=canada,type=location)
18#object(name=iraq,type=location)
18#object(name=montreal,type=location)
16#action(name=enter,country=canada,location=montreal,person=nizar_a
,document=fake)
18#object(name=winnipeg,type=location)
18#action(name=travel,person=nizar_a,from=montreal,to=winnipeg)
18#action(name=resided,person=nizar_a,location=iraq)
//INS01
19#action(name=enter,country=us,location=minnesota,person=nizar_a,do
cument=valid)
//NSA01
20#object(name=website_islamic,type=website)
20#object(name=021127,type=date)
20#property(name=website_islamic,date=021127)
//FBI12
21#action(name=web_connection,from=atanta,to=website_islamic)
21#action(name=web_connection,from=colorado,to=website_islamic)
//FBI13
```

```
22#action(name=meeting,object1=kamel_j,object2=muhammed_j,location=g
eorgia)
//FBI14
23#object(name=car_chicago,type=vehicle)
23#action(name=rent,vehicle=car_chicago,person=majed_h,location_from
=chicago)
23#action(name=return_vehicle,vehicle=car_chicago,location=minneapol
is)
```

9 Appendix d – Intelligence Domain Description

```
//rules for weapons cenario


//If a person has bought weapons from another person then he/she has
            possession of those weapons

weapons1#action(name=purchase,object=WEAPONS,buyer=BUYER,seller=SELL
            ER);object(name=BUYER,type=person);object(name=SELLER,ty
            pe=person);object(name=WEAPONS,type=weapons)#>relation(t
            ype=possession,owner=BUYER,object=WEAPONS)



//if a person has bough weapons then he she has possession of those
            weapons

weapons2#action(name=purchase,object=WEAPONS,buyer=BUYER);object(nam
            e=BUYER,type=person);object(name=WEAPONS,type=weapons)#>
            relation(type=possession,owner=BUYER,object=WEAPONS)


//if an organization purchase weapons then that organization has
            possession of those weapons

weapons4#action(name=purchase,object=WEAPONS,buyer=ORG);object(name=
            WEAPONS,type=weapons);object(name=ORG,type=organization)
            #>relation(type=possession,owner=ORG,object=WEAPONS)


//If a person has possesion of weapons and that person belongs to an
            organization then that organization also has possesion
            of those weapons

weapons5#relation(type=member_of,person=PERSON,organization=ORG);obj
            ect(name=ORG,type=organization);relation(type=possession
            ,owner=PERSON,object=WEAPONS);object(name=WEAPONS,type=w
```

```
            eapons)#>relation(type=possession,owner=ORG,object=WEAPO
            NS)


//the weapons are where it's buyer are
weapons6#action(name=purchase,object=WEAPONS,buyer=ORG);object(name=
            WEAPONS,type=weapons);relation(type=is_in,object1=ORG,ob
            ject2=LOC);object(name=LOC,type=location)#>relation(type
            =is_in,object1=WEAPONS,object2=LOC)


//if two people meet, one of them has weapons and money was
            exchanged then the other person has bought them
sale1#action(name=meeting,object1=PERSON1,object2=PERSON2);relation(
            type=possession,owner=PERSON1,object=WEAPONS);action(nam
            e=deposited,amount=MONEY,receiver=ORG);relation(type=mem
            ber_of,person=PERSON1,organization=ORG);object(name=WEAP
            ONS,type=weapons);object(name=MONEY,type=money);object(n
            ame=PERSON2,type=person);object(name=PERSON1,type=person
            )#>relation(type=possession,owner=PERSON2,object=WEAPONS
            )


//if a person has an alias then a possession of weapons means that
            the alias also has possession
alias1#object(name=ALIAS,type=person);relation(type=alias,object1=AL
            IAS,object2=PERSON);relation(type=possession,owner=PERSO
            N,object=WEAPONS);object(name=WEAPONS,type=weapons)#>rel
            ation(type=possession,owner=ALIAS,object=WEAPONS)
```

```
//if a vehicle was rented in place X and delivered in place Y then
            that vehicle traveled from X to Y

travel1#action(name=rent,vehicle=OBJECT,person=PERSON,location_from=
            FROM);action(name=return_vehicle,vehicle=OBJECT,location
            =TO);object(name=FROM,type=location);object(name=TO,type
            =location);object(name=PERSON,type=person)#>action(name=
            travel,object=OBJECT,from=FROM,to=TO);action(name=travel
            ,object=PERSON,from=FROM,to=TO)


location1#relation(type=is_in,object1=ORG,object2=LOC1);relation(typ
            e=is_in,object1=LOC1,object2=LOC2)#>relation(type=is_in,
            object1=ORG,object2=LOC2)


//transport of weapons - if the person with the weapons meets with a
            person that rent's a truck and travels then the weapons
            moved

transport1#action(name=meeting,object1=PERSON1,object2=PERSON2,locat
            ion=LOC1);action(name=travel,object=PERSON1,from=LOC2,to
            =LOC3);relation(type=possession,owner=PERSON2,object=WEA
            PONS);object(name=WEAPONS,type=weapons)#>action(name=tra
            vel,object=WEAPONS,from=LOC2,to=LOC3);relation(type=is_i
            n,object1=WEAPONS,object2=LOC3)


//if someone enters a country with a fake document after beeing in
            IRAQ then they are terrorists

terrorist1#action(name=enter,country=COUNTRY,location=LOC,person=PER
            SON,document=fake);action(name=resided,person=PERSON,loc
            ation=iraq)#>property(name=PERSON,type=terrorist);relati
            on(type=member_of,person=PERSON,organization=al_qaeda)

terrorist1#action(name=enter,country=COUNTRY,location=LOC,person=PER
            SON,document=fake);action(name=resided,person=PERSON,loc
```

```
            ation=afghanistan)#>property(name=PERSON,type=terrorist)
            ;relation(type=member_of,person=PERSON,organization=al_q
            aeda)


//is someone enters the counrty in a certain place then he is in
            that place

terrorist3#action(name=enter,country=COUNTRY,location=LOC,person=PER
            SON,document=DOC)#>relation(type=is_in,object1=PERSON,ob
            ject2=LOC)


//terrorist act - general rule

//if we have terrorists and weapons in the same place and a date
            then warning..

terror1#property(name=PERSON,type=terrorist);relation(type=is_in,obj
            ect1=PERSON,object2=LOC);object(name=WEAPONS,type=weapon
            s);relation(type=is_in,object1=WEAPONS,object2=LOC)#>act
            ion(name=terrorist_act,location=LOC,terrorist=PERSON)
```