Natural Language Fact Extraction and Domain Reasoning using Controlled English

David Mott
Emerging Technology Services, IBM United Kingdom Ltd
Hursley Park, Winchester, UK

Stephen Poteet, Ping Xue
Boeing Research & Technology
Seattle, WA, US

Ann Copestake
University of Cambridge

Abstract

This paper describes the research undertaken under the International Technology Alliance (ITA) [1] to explore the ELICIT identification task [2] as an example of a complex problem that simultaneously involves many cognitive tasks which would have to be modelled in order to be achieved by a computer agent. This task requires the identification of the participants in a possible attack, using information contained in short Natural Language (NL) sentences. Successful problem solving in this task requires the analysis of NL sentences based on common sense knowledge, the representation of the domain, the representation of a problem solving strategy, the construction of suitable reasoning rules, the running of the rules, the making of assumptions, and detection of inconsistencies and the presentation of rationale. All of these problems are those faced by an analyst in their task, and an understanding of how these problems could be addressed could allow the development of support mechanisms to facilitate their performance. The paper describes research into how the use of a Controlled Natural Language (ITA Controlled English, or CE) [3] can contribute to the solution of these problems. Knowledge of the domain was represented in a CE conceptual model, and this model defined the concepts and the logical rules to perform reasoning. The full task of interpreting the NL sentences was considered at this stage to be too complex, since much common sense knowledge would have to be extracted and represented. It was therefore decided initially to have a human convert the original sentences into other more simple English sentences thus temporarily avoiding the more complex problems of common-sense interpretation, and it was also decided to focus on one aspect of the identification task (WHO were the responsible agents). The sentences were analysed by the ERG system [4] into an MRS [5] linguistic semantic representation, which was further analysed by a CE-based linguistic-to-domain semantic mapping system, resulting in the extraction of domain CE facts from the sentences. However some of the NL sentences expressed rules rather than facts, and research was undertaken to determine how the CE rules for processing the sentence could themselves generate other CE rules that could be added to the domain model in order to solve the ELICIT identification problem. The paper describes an extension of the CE meta-model and CE system to allow rules to be objects generated by other rules. The domain model and extracted rules were run to identify the "WHO", and this provided a set of CE facts expressing the rationale graph of the reasoning leading to the conclusion. This reasoning could be seen to be based upon two assumptions in the interpretation of the sentences, and these assumptions could be explored in the rationale graph, allowing the analyst to understand the reasoning and the sources of uncertainty in the conclusions.

0 Synopsis for busy readers

This paper covers much detail about the NL processing and reasoning, and we therefore provide a brief synopsis in order to guide the reader through the key points. Numbers in brackets (X) refer to the relevant sections of the paper.

The research aims to support analysts and other users in collaborative reasoning based on NL information (1) and ITA Controlled English (CE) to represent facts and reasoning (1.1).

The ELICIT identification task (2) is a complex problem that requires collaborative reasoning to identify key aspects of a possible attack, using English sentences as the source of intelligence information. This task provides a good focus for our research, as it requires the combination of NL processing, reasoning and rationale. However the sentences contain ambiguities requiring knowledge beyond basic domain reasoning to resolve (2.1), and it was decided initially to simplify the sentences into English that avoided these complex ambiguities (2.2). A CE domain model was constructed (2.3), consisting of concepts and inference rules, to enable the expression of facts, and the reasoning over these facts. This contained such concepts as operatives, embassies, time intervals, and working relationships between groups. A problem solving strategy to drive the reasoning was also modelled (2.4) based upon the intuition that the ELICIT task was best solved by a process of elimination. These components were combined to solve "who" was responsible for the potential attack.

Our NL processing research (3) utilises external DELPH-IN linguistic resources, the English Resource Grammar (ERG) to perform a deep parse of the sentences and Minimal Recursion Semantics (MRS) to represent the extracted semantics. These resources were integrated into the CE system, by converting the MRS (3.1) into CE (3.2, 3.3), with a view to converting this into domain CE, expressed in terms of the domain model. This was achieved by following a number of transformations of the CE. The MRS still contained linguistic information (3.4) and it was desirable to convert it into a generic (linguistic-free) semantic representation, based upon the concept of "situation" (3.5), named entities involved (3.5.1), their types (3.5.2) and the roles that they play (3.5.3). From the generic semantics, it was possible to construct domain semantics, leading to CE facts extracted from the sentence that are expressed in terms the user can understand (3.5.4). In these transformations, significant use was made of the ability for CE to undertake meta-reasoning, such as linking "words" to "concepts".

Sentences specific to solving the "who" aspect of the ELICIT task were simplified into English that still required full NL processing to interpret (4). Even though simplified, there were linguistic challenges to be solved, including the handling of negated and modal situations (those that "are not" or "may be") (4.1.2), and handling of generic entities such as "daylight" (4.1.3) and "locals" (4.1.4). These challenges correspond to known linguistic phenomena, some of which are still subject to research; an advantage of using CE is that there is a computational but readable specification of the linguistic theory. A significant challenge was that some sentences express rules rather than facts (e.g. "the Lion only works with the Bluegroup and the Azuregroup"); in order to use CE to analyse such sentences it was necessary to extend CE syntax to allow meta-reasoning so that rules could themselves construct other rules (4.1.5). This extension proved to be of more general use (see 5.3 and 6.3). A particular concern was to ensure that all CE-based transformation rules were generic and not just specific to the ELICIT sentences. This was achieved by further use of meta-reasoning, utilising semantic relations between concepts, such as that "possible participant" is a modal version of "participant" (4.1.2, but see also 4.1.5 and 5.3), and an approach to generalising rules was proposed (4.1.5). Thus facts and rules were extracted from the NL sentences (8)

Once facts and rules had been extracted from sentences, they could be used to infer new information (5). This was achieved by the rules (domain and problem solving) in the CE domain model together with the extracted facts and rules, resulting in the automatic

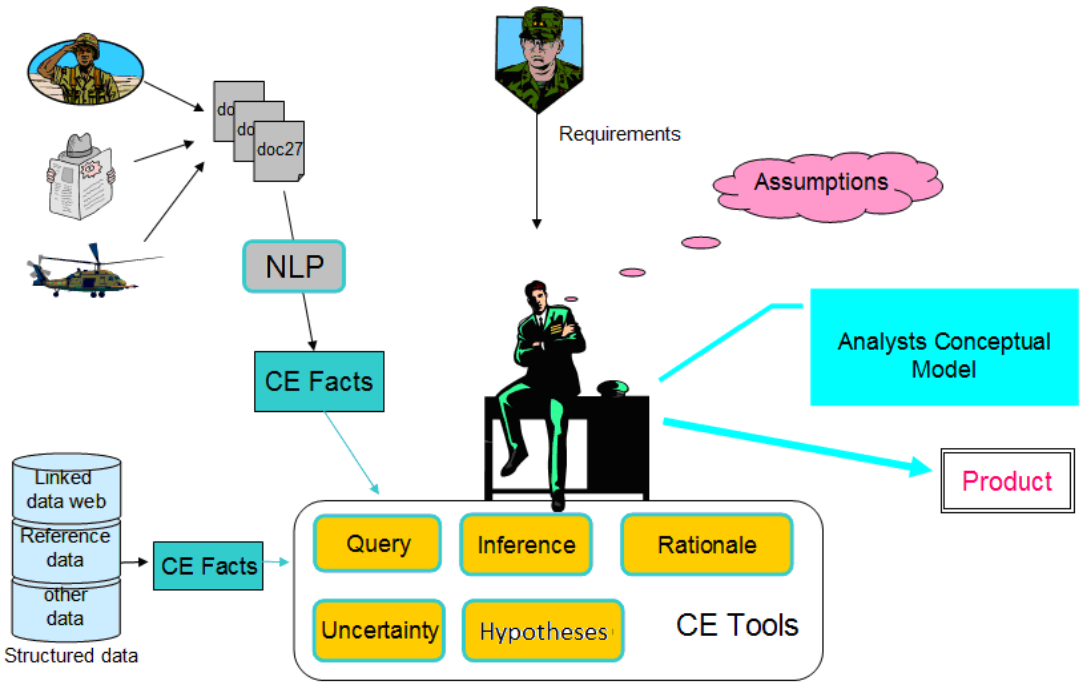
identification of the participants (the "who") in the attack (5.4). Of interest is the identification of "Violetgroup" as a participant, which was effected by the elimination of all other candidates, via a range of different reasons (5.1), followed by a rule that inferred that the only one remaining is the answer (5.2). This rule was dynamically generated from the knowledge of the complete set of possible candidates, using "rule-writing rules" (5.3).

A significant issue is the handling of ambiguities and uncertainties in the reasoning, which may arise from a number of different sources, including ambiguous interpretations of sentences and incomplete knowledge about the domain (such as the complete set of candidates) (6). We use assumptions and numerical certainty values (6.1) to represent CE facts that are uncertain, and apply assumption-based reasoning to infer new information, detect inconsistencies and label different sources of uncertainty (6.2) to the user in the rationale. An example is the assumption that the set of possible participants is known (i.e. the world is closed in respect of other possibilities), leading to the inference of the only possible participant, or to the detection of an error if the assumption is incorrect (6.3). Ambiguities also arise from interpretation of sentences, and we represent such ambiguities as assumptions which can be tracked through the rationale to allow the user to know the sources of ambiguity in any conclusion of high value information (6.4).

The NL analysis and reasoning to perform the ELICIT identification task has been achieved entirely via CE rules, and the steps performed by these rules have been recorded in a rationale graph. From this it is possible to determine the specific reasoning that led to conclusions together with any assumptions that might have been made on the way. This information can be shown to the user in order that they can understand the reasoning, and thereby evaluate the degree to which the conclusions can be trusted. Ways to visualise this rationale are shown throughout the paper.

1 Introduction

This paper reports work under the International Technology Alliance (ITA) program of research [1] on how military coalitions could be better supported in the use of networks of computers and in the use of information available to users in the performance of complex cognitive problem-solving tasks. One such task is shown in the diagram below:



Here intelligence analysts are required to provide information relevant to their commanding officers mission requirements. In order to provide information, the analysts must look at information from a variety of sources, including unstructured textual reports, as well as more structured sources such as databases and spreadsheets. For the analyst this is a complex cognitive task that requires the making of assumptions and reasoning based upon a "conceptual model" of the domain in which the analysis is taking place. Such an analyst requires support for querying, inference, handing of uncertainty, making hypotheses and understanding of the rationale for conclusions reached. In the ITA programme we are aiming to provide support for such an analyst, in the form of "CE Tools", based on a human-readable language (CE) and reasoning system.

1.1 ITA Controlled English

ITA Controlled English (or CE) [3] is a Controlled Natural Language, a subset of English, that is both human-readable and machine parseable, suitable for the expression of domain knowledge, concepts and reasoning. Being a derivative of ordinary English it is relatively easy for human analysts to use, but it also has a formal interpretation that is sufficiently unambiguous that a computer can interpret the input of the domain analysts and use it to perform inferencing. CE has been used in a number of different projects in the ITA, and we are now using it in NL processing, both as the semantic target language for the processing and for the configuration and guidance of the NL processing itself.

Central to the use of CE is a "domain model", a structure that holds all of the users' knowledge (concepts, relationships, logical inferences, constraints, and assumptions) of the domain in which the reasoning and problem solving is to be undertaken. In the example of the ELICIT task described in section 2, the domain is one of attacks, agents, groups, targets, embassies and dignitaries. In a different application, the domain might include medical diseases and patients, or plans, goals and activities. For any given problem solving task, there may be more than one domain involved. For example, the ELICIT task also includes reasoning about the NL sentences, and this requires a model of the linguistics domain. One domain may be an extension of another more abstract model; thus the ELICIT model is based upon a more general model of people, places, time, space etc. As a result, the CE conceptual models are often hierarchically defined. At the top of this hierarchy is the CE meta-model that contains concepts for describing and reasoning about concepts themselves. This meta-model turns out to be key for the linguistic processing, as described in this paper. At the bottom of the hierarchy is the specific domain model of interest to the analysts. An example of a hierarchy of models is given below:

Meta Model	Concept, Entity Concept, Relation Concept, Conceptual Model	belongs to, has as domain
Semiotics	Thing, Meaning, Symbol	stands for, expresses
General Semantics	Agent, Spatial Entity, Temporal Entity, Situation, Container	has as agent role, is contained in
Linguistic	Sentence, Phrase, Noun, Word, Word Sense, Predicate, Linguistic Frame	has as dependent, is parsed from, expresses
Analysts Domain Model	Place, Person, Village, Communication, IED, Facility, ...	is located in, monitors

These different levels will be touched upon in the various sections of this paper.

2 The ELICIT identification task

The ELICIT laboratory has devised the ELICIT framework [2] for researching into how differing organisational structures and communication patterns affect collaborative solving of problems requiring reasoning and interpretation of facts. Underpinning this framework is the "ELICIT task" that is set to be solved by different groups of participants operating under different communication structures. The task involves the identification of the key aspects of a (simulated) potential planned terrorist attack, and this identification must be achieved by interpretation of a set of simple facts (sentences or factoids) expressed in English. The key aspects are WHO is going to perform the attack, WHAT will be attacked, and WHEN and WHERE the attack will take place. The ELICIT task has been the subject of much research, and is therefore an interesting and representative problem to be addressed by our research.

2.1 Problems of Ambiguity and Interpretation

The set of ELICIT sentences were analysed in order to understand what concepts a suitable domain model might contain, and how the facts might be represented in this domain model [6]. During this analysis it was realised that there were significant potential ambiguities as to how these sentences might be interpreted. It should be noted that no specific instructions or background to the domain were given to the users to help disambiguation. Some examples of sentences together with their ambiguities are:

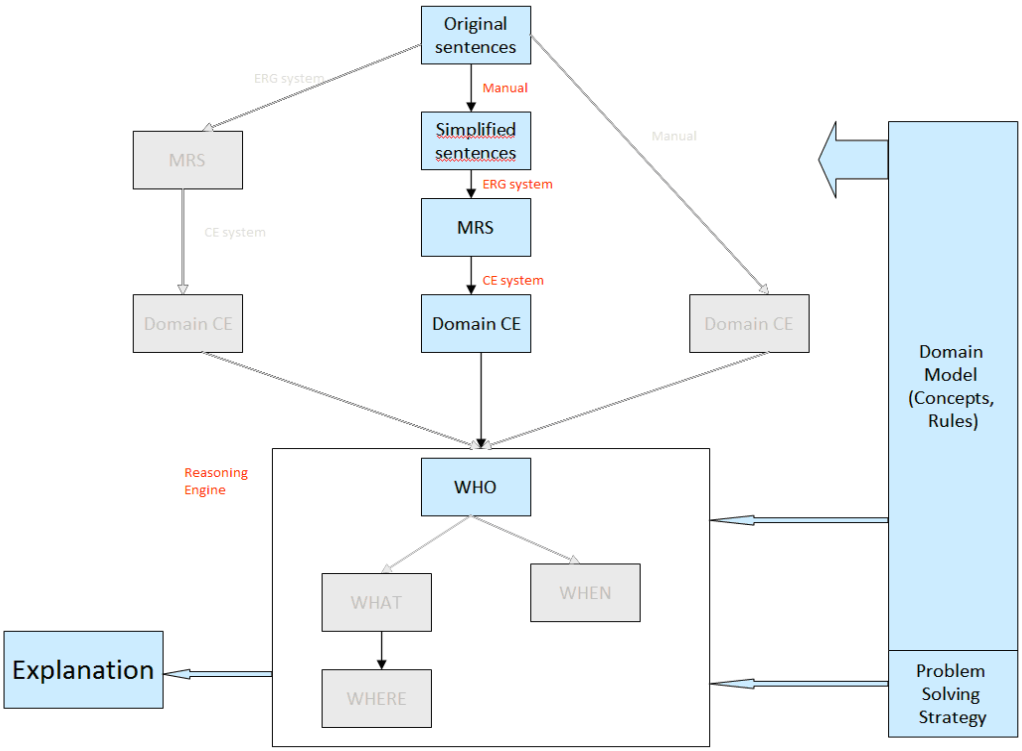
- *The Purple group or the Gold group may be involved*
  - What is the logical meaning of "or" in this context? Is it an exhaustive list of groups that may be involved? Is it inclusive or exclusive?
- *The attackers are focusing on a high visibility target*
  - What is the definition of high visibility?
- *The Lion will not risk working with locals*
  - Does this imply that the Lion actually does not work with locals?
- *Dignitaries in Epsilonland employ private guards*
  - Does "in" mean "belonging to" or "located in"?
- *Reports from Tauland, Chiland and Psiland indicate surveillance ongoing at coalition embassies*
  - Are the reports from the embassies or the host countries?
- *The Violet group prefers to operate in daylight*
  - Does this refer to lighting conditions or time periods? Is "daylight" opposite to "at night"? Does this mean they definitely don't operate at night?
- *The Azure and Violet groups use only their own operatives, never employing locals*
  - This seems to be a rule rather than a fact, but how is the statement to be generalised just enough to capture the intent of the rule?

These ambiguities are derived from several different sources: logical, lexical and domain-specific. Furthermore some of these sources cannot be disambiguated without detailed domain knowledge and common sense, something that is notoriously difficult to capture in sufficient detail to cover all possible circumstances.

2.2 Three approaches

In general we have considered three approaches to handing the complexity of the ELICIT sentences: to hand-code all of the sentences directly into CE, thus avoiding the problems of Natural Language (NL) interpretation completely; to simplify the sentences into a less ambiguous, but still free-form Natural Language, which would then be analysed automatically by NL processing; to press ahead on building common sense into the system, and attempting to analyse the sentences as they are. It was decided at this stage in the research to follow the second path, by having a human simplify the sentences and run the NL processing on these. Even if this does require some form of human intervention, nevertheless it is still of benefit as

the subsequent problem solving on the extracted facts is difficult to achieve manually. In addition, it was judged to be a relatively easy and natural task for a human to do, since there were no real constraints on the syntax of the simplified sentences. As well as using the simplified approach, it was decided to start with just the WHO component of the ELICIT task, that is to determine the agent(s) responsible for the future attack. The following diagram shows the information flow in this second approach highlighted in blue.



This diagram shows the flow of the NL processing through the ERG system that parses the sentences and the CE system that generates extracted facts based on the domain model. These facts are passed to the Reasoning Engine (an interpreter of CE rules to generate inferences) which is tasked with determining the WHO of the ELICIT task. The Reasoning Engine is guided by the rules and concepts in the domain model together with a set of rules defining the problem-solving strategy. The resulting conclusion about the WHO is available for explanation to the users, based upon the rationale of reasoning steps leading from premises to conclusions. All of these components are described in more detail in the sections below.

2.3 ELICIT Domain Model

The ELICIT identification task requires reasoning about a number of different types of concepts:

- agents, operatives, groups of operatives
- targets, financial institutions, visiting dignitaries, embassies
- time intervals, daytime, nighttime
- attack situations, participants, non-participants
- working relationships, works with, cannot work with

A full description of the ELICIT domain conceptual model is beyond the scope of this paper, see [7] for the full model. Here we describe some examples that are involved in the reasoning described in the paper. These are defined in the CE model using "conceptualise" statements.

The ELICIT conceptual model extends from a general military model that includes the concept of "agent", something that causes events to happen and is volitional (i.e. has intent). Examples of agent include military forces, foreign fighters, individual soldiers, aid agencies, village elders, but not natural forces such as earthquakes. In the ELICIT domain there are individual hostile agents called operatives and terrorist group agents called groups. For example an operative is conceptualised as a subtype of agent:

```
conceptualise an ~ operative ~ O that is an agent.
```

In this way, every operative, such as "the operative Lion" and every group, such as "the group Azuregroup" inherit the attributes and relationships that define the concept of agent.

The general military model represents events and state of affairs that occur in the world by the concept of "situation", and more details on this concept are given below. For example a type of situation that represents an attack may be defined by:

```
conceptualise
  an ~ attack  situation ~ X that is a situation.
```

In an attack situation there are a number of agents (operatives and groups) performing the attack, and we have chosen to model this simply as "involvement" in a situation:

```
conceptualise
  the situation S ~ involves ~ the agent A.
```

This allows the construction of CE sentences such as "the attack situation Elicitattack involves the agent Lion" where Elicitattack refers to the current potential attack situation under analysis by the identification task and Lion is the name of an agent. When an agent is involved in the Elicitattack, that agent may also be described as a "participant", as in "the agent Lion is a participant". This is conceptualised via the following, which also ensures that all participants are also agents:

```
conceptualise
  a ~ participant ~ X that is a agent.
```

It is useful to create additional related concepts, including the negative "non-participant" and the modal "possible participant":

```
conceptualise
  a ~ non-participant ~ X that is a agent.

conceptualise
  a ~ possible participant ~ X that is a agent.
```

A key component of the conceptual model is the definition of rules, expressing logical inferences that may be made about the entities and relationships in the domain. For example there is a simple relationship between the concept of a participant and being involved in the Elicitattack, which may be expressed by the following rule, called "sem_participant":

```
[ sem_participant ]
if
  ( there is a situation named Elicitattack ) and
  ( there is a participant named P )
then
  ( the situation Elicitattack involves the participant P ).
```

In this rule there is a variable P that may be matched to any instance of the concept "participant", and the informal interpretation of the rule is all "participants" are involved in the Elicitattack.

Another concept that describes agents is that of being non-operational, i.e. they have some characteristic, such as regrouping, or being in custody, that prevents them from operating in the attack situation. We will not show the "conceptualise" statement for this, but instead will show a rule that captures the fact that a non-operational agent cannot be a participant:

```
[ non_operational ]
if
( there is an agent named A that is a non-operational agent )
then
( the agent A is a non-participant ).
```

An example of a domain relational concept is that of the working relationships between agents. Two basic relationships of "working with" and its negative "cannot work with" may be conceptualised by the following:

```
conceptualise
   the agent X ~ works with ~ the agent X1 and
       ~ cannot work with ~ the agent X2.
```

This allows the statement of such facts as "the agent Lion works with the group Azuregroup" and "the agent Lion cannot work with the group Coralgroup" (given that a group is a type of agent). The inference of these relationships is key to the reasoning described below, and there are several different reasons why agents can and cannot work with each other. One is that there is a third agent that is working with the second with whom the first cannot work; this entails that the first cannot work with the second either. This may be expressed as:

```
[ cannot_work_with ]
if
  ( the agent A cannot work with the agent B ) and
  ( the agent B works with the agent C )
then
  ( the agent A cannot work with the agent C ).
```

Another reason for not being able to work together is when the agents operate at different times of day, for example daytime and nighttime. This may be expressed by:

```
[ no_time_overlap ]
if
  ( the operative A operates in the time interval TA ) and
  ( the group B operates in the time interval TB ) and
  ( the time interval TA does not overlap the time interval TB )
then
  ( the operative A cannot work with the group B ).
```

The details of the conceptualisation of time intervals are not given here (they are inherited from the general military model). However two instances of "time interval" have been defined, daytime and nighttime, and the relevant overlap information is given by the CE fact "the time interval daytime does not overlap the time interval nighttime" where "does not overlap" is one of the Allen time relations [8] encoded in the general model. Such approximations to time intervals as daytime and nighttime is sufficient to solve the ELICIT task.

If the details of the agents "operating hours" are known as CE facts, then additional "cannot work with" relations between the groups can be inferred. Thus if it is known that:

the group Coralgroup operates in the time interval nighttime.
the operative Lion operates in the time interval daytime.

then the no_time_overlap rule will infer:

the operative Lion cannot work with the group Coralgroup.

When reasoning occurs, the rationale for the reasoning is recorded [22], and various ways to diagram this rationale are being explored. One approach is to show the reasoning as a form of "proof table", where the logic is expressed as a sequence of rows, each row being a single logical proposition, and the last row being the concluding proposition. The proof table also shows the rule as a column with light red indicating the preconditions of the rule and dark red representing the conclusion of the rule. For example, the following proof table shows the no_time_overlap rule applied to the information as described above. The rows have been simplified from the CE that expresses the proposition, for example the first row should really state "the thing daytime is a time interval", but "the thing" has been omitted to reduce space in the table. Further discussion and examples are given later on the use of such tables.

daytime is a time interval		
nighttime is a time interval		
Lion is a operative		
Coralgroup is a group		
Coralgroup operates in nighttime		
Lion operates in daytime		
daytime does not overlap nighttime		
Lion cannot work with Coralgroup	no_time_overlap	

2.4 Problem Solving Strategy

If the ELICIT sentences are analysed to extract the initial facts (such as "the operative Lion is a participant") then the basic rules and concepts in the domain model will infer new information about the facts of the attack situation and associated entities, such as who can and cannot work with each other. However this is insufficient on its own to solve the ELICIT identification task, since the task is essentially a problem solving task that requires the setting up of problem solving goals to be achieved. For example, to identify the WHO component of the attack, it is necessary to set up the goal of determining the possible participant, and this may not just "fall out" of the basic inference from the domain rules. In effect a strategy for solving the problem must be designed. From an informal analysis of the nature of the identification task and the type of information expressed in the sentences, it is natural to conclude that this task is a type of constraint-based problem, where the facts provide a possible set of participants, and constraints on who the participants can be (for example non-operational groups cannot be participants). The correct problem solving strategy seems to be a process of elimination; the constraints eliminate possible participants and the only ones remaining are the candidates for the WHO. If there is a single remaining candidate then this is the definite WHO; if there are no possible remaining candidates then there is an inconsistency in the domain logic (or the initial set of facts) and the problem has no solution; if there is more than one remaining candidate then either other information must be sought to eliminate

all but one or the solution must remain uncertain. Although it is "natural" to view this as a constraint-based problem, it is not easy to logically justify this statement. However it is the case that three of the ITA researchers when working on solving this problem, all (including the author) independently came to this conclusion, and it would be an interesting topic of research to determine why this is the case.

The outcome of this view of the problem solving strategy also has an implication on the design of some aspects of the domain model. Firstly the propositions expressed in the facts and inferences should be essentially negative since we are seeking to rule-out participants; we are interested in "non-participants", "non-operational" groups, and "non-working relationships", as reflected in the examples above. There are two approaches to representing such negative facts; the first is to design the concepts as directly expressing negative ideas in the terms used (such as "non-participant"); the second is to express the concepts positively but make the sentence negative, as in "it is false that the group Coralgroup is a participant". There are subtle differences in the logic of these two approaches, and this will be touched upon later; but we have chosen the first (negative concept) approach, rather than the second negative sentence approach.

The second implication of a constraint-based problem solving strategy is that it is necessary to set up additional problem solving logic, in the form of CE rules, to undertake the reasoning about the eliminated candidates and such logic is outside of the domain model itself (in effect the logic is meta-logic or meta-cognition). Specifically, the domain rules may be used to do the actual elimination of participants, but the meta-logic must handle the creation of the possible participants and the resulting inference of the actual participant. The key step is to have a rule of the general form:

if all candidates but one have been ruled out then the final candidate is the one!

This is not expressible directly in CE, and a less elegant approach is required. Suppose there are two possible participants, groupA and groupB. Then it is possible to construct rules:

```
if ( the group groupA is a possible participant ) and
   ( the group groupB is a non-participant )
then ( the group groupA is a participant ).

if ( the group groupB is a possible participant ) and
   ( the group groupA is a non-participant )
then ( the group groupB is a participant ).
```

to cover all the cases. If there are N candidates then N such rules must be written[1] , but this cannot be achieved unless the full set of possible candidates is known. The approach we have taken is to automatically generate the rules at the point where the set of candidates are known (or thought to be known), and this is described in section 5.3. For now it is useful to note that the generation of these rules requires the knowledge that the concept "participant", "non-participant" and "possible participant" are all semantically related in certain ways.

It is also useful to create a further rule that detects when all possible candidates are eliminated, and to generate a logical inconsistency when this occurs, indicating that there is something wrong with the logical reasoning. This extra rule will look something like:

```
if ( the group groupA is a non-participant ) and
   ( the group groupB is a non-participant )
```

[1] each with N premises, one of which includes “is a possible participant” and the other N-1 of which include “is a non-participant”

then (there is an inconsistency named IC).

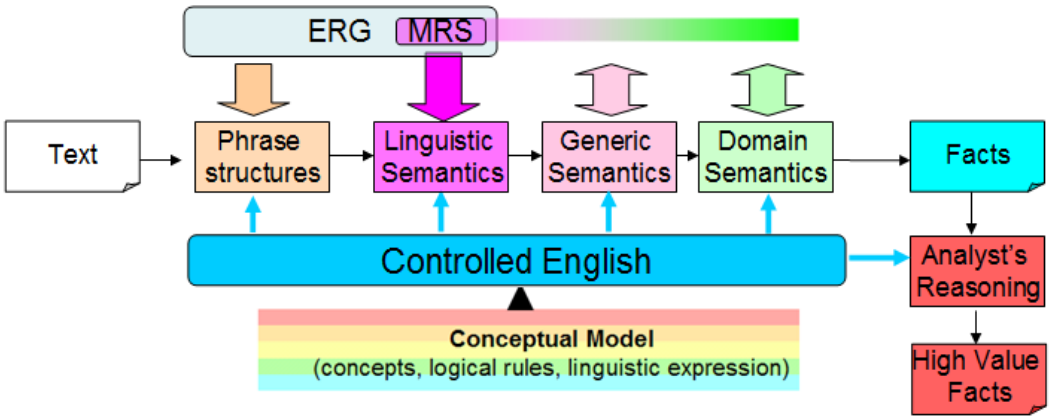
The generation of this rule is also achieved automatically, and will be described in section 5.3, and an example of its use when an incorrect assumption was made as to the total set of possible participants will be given in section 6.3

The set of domain rules and the set of problem solving rules, when put together, allow the solution of the WHO component of the ELICIT identification task.

3 The NL processing research system

The NL processing research being undertaken in the ITA, and described in this paper, is based upon linguistic resources from the DELPH-IN project [9]. The specific components used are the English Resource Grammar (ERG) [4] a high-precision grammar for English, the PET parser [10], and Minimal Recursion Semantics [5] to express sentence semantics as logical predicates on entities mentioned in the sentence. Our research focuses on transforming the linguistically-focussed semantics contained in the MRS into domain semantics based on the CE conceptual model, resulting in the extraction of CE facts.

The DELPH-IN components (ERG and MRS) are brought together with the CE reasoning system as shown in the following diagram:



The information flows through the system as follows:

- English text is sent to the ERG for parsing, and this results in the output of MRS predicates holding the linguistic semantics of this sentence. The MRS is converted into CE for further processing by the CE reasoning components.
- The linguistic semantics is transformed into a generic semantic form, including such concepts as situations and the roles that entities play in that situation
- The generic semantics is transformed into a specific domain semantics, which might include such concepts as people, places, agents, attacks, targets etc. This domain semantics is output in the form of CE facts, that conform to the conceptual model.
- The CE facts are used to perform domain specific reasoning, leading to inference of high valued information of use to the analysts, and based upon inference rules that may be written by the users.

The operation of the system will be illustrated on a simple sentence, "John chased the cat".

3.1 The raw form

When this sentence is parsed by the PET parser according to the ERG, a parse derivation and an MRS representation is generated. The resulting MRS from the topmost parse is given below, in the raw output from the PET parser:

```
[ LTOP: h1
  INDEX: e3 [ e SF: PROP TENSE: PRES MOOD: INDICATIVE PROG: - PERF: - ]
  RELS: <
      [ proper_q_rel<0:4>
        LBL: h4
        ARG0: x7 [ x PERS: 3 NUM: SG GEND: M IND: + ]
        RSTR: h6
        BODY: h5 ]
      [ named_rel<0:4>
        LBL: h8
        ARG0: x7
        CARG: "John" ]
      [ "_chase_v_1_rel"<5:11>
        LBL: h2
        ARG0: e3
        ARG1: x7
        ARG2: x9 [ x PERS: 3 NUM: SG IND: + ] ]
      [ _the_q_rel<12:15>
        LBL: h10
        ARG0: x9
        RSTR: h12
        BODY: h11 ]
      [ "_cat_n_1_rel"<16:19>
        LBL: h13
        ARG0: x9 ] >
  HCONS: < h1 qeq h2 h6 qeq h8 h12 qeq h13 > ]
```

A full description of this output is beyond the scope of this paper, but it may be seen that there are several components representing the semantic content of the sentence as derived by the ERG. Here we give a brief description of these components with further details as to their meaning being provided in the next section.

The first component, labelled by INDEX, is the top level "situation" or "event" expressed by the sentence. The second component, labelled by RELS, is a set of predicates, or elementary predications, each holding over a number of arguments, containing the main semantic content. For example there is a predicate "_cat_n_1_rel" that holds over the "ARG0" x9. The third component, labelled HCONS, lists constraints on possible scoping relationships between quantifier-like elementary predications and other elementary predications that are to be quantified, see [5] for further details.

3.2 The CE form

The raw MRS is converted into a CE version, so that it is available for further automated analysis by the CE-based NL processing mechanisms described below. In addition, for some users at least, the CE form may be more readable than the original raw form, but it should be noted that the CE form is also of a low level, in that it uses terms and relationships that describe the raw MRS, which require some specialist knowledge, these terms being defined by the conceptual model of MRS structures. The conversion is performed by a Prolog program that runs the PET parser and transforms the raw output into CE.

The CE that corresponds to the raw MRS is shown below, although not all of the raw data is yet used. The set of elementary predications is as follows:

the mrs elementary predication #ep0 is an instance of the mrs predicate 'proper_q_rel'
and has the thing x7 as zeroth argument.

the mrs elementary predication #ep1 is an instance of the mrs predicate 'named_rel'
and has the thing x7 as zeroth argument
and has 'John' as c argument.

the mrs elementary predication #ep2 is an instance of the mrs predicate '_chase_v_1_rel'
and has the situation e3 as zeroth argument
and has the thing x7 as first argument
and has the thing x9 as second argument.

the mrs elementary predication #ep3 is an instance of the mrs predicate '_the_q_rel'
and has the thing x9 as zeroth argument.

the mrs elementary predication #ep4 is an instance of the mrs predicate '_cat_n_1_rel'
and has the thing x9 as zeroth argument.

These CE statements are essentially the same as the set of predicates from the raw form. Each mrs elementary predication is an **instance** of a predicate and has one or more arguments, indicated as zeroth, first and second. The meaning of these arguments depends upon the type of the predicate. The values of these arguments are either situations, i.e. states of affairs, or things, i.e. individual entities, whose type is not known at this stage. The detailed nature of this set of predications will be described in the next section.

There are some additional features associated with the things and situations, as shown below:

the situation e3 has the category 'SF:PROP' as feature and has the tense category 'PRES' as feature and has the mood category indicative as feature and has the category 'PROG:-' as feature and has the category 'PERF:-' as feature.

the thing x7 has the person category third as feature and has the number category singular as feature and has the gender category male as feature and has the category 'IND:+' as feature.

the thing x9 has the person category third as feature and has the number category singular as feature and has the category 'IND:+' as feature.

These features specify linguistic categories on the things such as person, number, gender, and categories on the situations such as tense and aspect. The features are not currently used to any significant degree in the NL processing, and will not be considered further in this paper.

The scoping relationships between the elementary predications are shown below:

the mrs elementary predication #ep0_563 equals modulo quantifiers the mrs elementary predication #ep1_563.

the mrs elementary predication #ep3_563 equals modulo quantifiers the mrs elementary predication #ep4_563.

These will not be discussed further.

3.3 The tabular form

It is useful to see the complete set of CE sentences as a whole, representing the semantics in a more graphical manner. This may be seen in the following table, generated automatically:

the ep #ep0 |proper_q_rel					
	has as zeroth argument			x 7	
the ep #ep1 |named_rel					
	has as zeroth argument			x 7	
	has as c argument	John			
the ep #ep2 |_chase_v_1_rel					
	has as zeroth argument		the situation e3		
	has as first argument			x 7	
	has as second argument				x9
the ep #ep3 |_the_q_rel					
	has as zeroth argument				x9
the ep #ep4 |_cat_n_1_rel					
	has as zeroth argument				x9

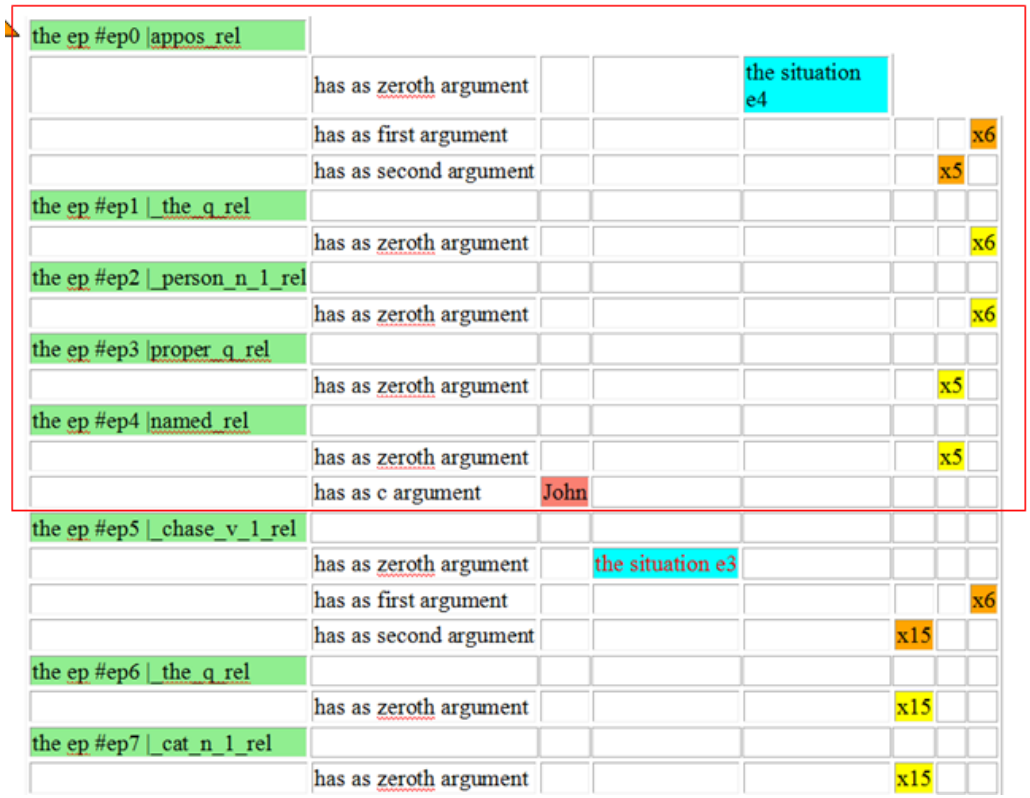
The table displays the CE in a reduced form. The rows with an entry in the first column represent the elementary predications, with a name generated from the original elementary predication name concatenated with the name of the predicate. The arguments for each predication are shown in the subsequent rows. The values of each argument are given in the columns to the right. Each column holds a single thing or situation, with entries in the rows where the thing is an argument of a predication. Thus the last column represents the thing x9 and shows that it occurs as the second argument of "_chase_v_1_rel", the zeroth argument of "_the_rel" and the zeroth argument of "_cat_n_1_rel".

A brief description of the information in this table (and hence of the CE sentences and the original raw MRS) may now be given. There is an overall situation named e3 (shown as red text) that is an instance of the predicate "_chase_v_1_rel" with two arguments. The first is the thing x7 which is predicated by "_proper_q_rel" and "named_rel" whose arguments include the string "John"; this combination represents a thing that is expressed by a proper noun, John. The second argument of the chase situation is the thing x9 that has the predicates noted above.

This structure also has a richer semantic interpretation, based on the notion of situations, which represent "states of affairs" together with entities that play roles in this state of affairs. The situation e3 represents a "chase situation" which has two roles. The first role is taken by the thing x7 which represents an entity called John. The second role is taken by x9 which represents a cat, although this cat is not identified. The first role expresses the thing doing the chasing, and the second role expresses the thing being chased. It should be noted that in the design of the ERG, no explicit commitments are made as to the nature of each role, so traditional linguistic roles such as agent and patient are not explicitly used.

3.4 Linguistic semantics and intermediate forms of MRS

It may be seen in the raw forms of the MRS shown above, there is a semantic content, but the predicates expressing this content are linguistically nuanced. For example there is a "proper_q" predicate that uses the linguistic concept of proper noun. In this simple example the degree of linguistic colouring is not much in evidence, but it can be seen in a more complex example. The tabular MRS output for the sentence "the person John chased the cat" is as follows (the red box has been added by hand):



the ep #ep0 |appos_rel							
	has as zeroth argument			the situation e4			
	has as first argument						x6
	has as second argument					x5	
the ep #ep1 |_the_q_rel							
	has as zeroth argument						x6
the ep #ep2 |_person_n_1_rel							
	has as zeroth argument						x6
the ep #ep3 |proper_q_rel							
	has as zeroth argument					x5	
the ep #ep4 |named_rel							
	has as zeroth argument					x5	
	has as c argument	John					
the ep #ep5 |_chase_v_1_rel							
	has as zeroth argument		the situation e3				
	has as first argument						x6
	has as second argument				x15		
the ep #ep6 |_the_q_rel							
	has as zeroth argument				x15		
the ep #ep7 |_cat_n_1_rel							
	has as zeroth argument				x15		

Here the table includes an "appos_rel" predicate with the first and second arguments capturing the relationship between the person and the proper noun John, and with the first argument functioning as the entity used in the "doing the chasing" role. However this predicate has a linguistic name, encoding the linguistic concept of "apposition", whereas a pure logical representation of the propositions underlying this sentence would probably not use the concept of apposition. The inclusion of linguistic terms in the semantics is caused by the need to maintain a somewhat one-to-one relationship between grammatical structure and semantic output arising from the use of the same ERG structures to hold both types of information. [21] Thus the output MRS is not entirely clear of linguistic concepts, and it is reasonable to describe the raw MRS output as "linguistic semantics".

In order to start the transformation from linguistic semantics to domain semantics it is useful to convert some of the raw linguistic structures into an intermediate form, called "intermediate MRS", that abstracts away some of the linguistic concepts. An example of intermediate MRS is that of a "definite quantification" that represents the existence of a definite thing, one that is expected to be known to the listener. This may be expressed by a noun phrase such as "the cat" in the previous example, and is flagged in the MRS by the "the_q_rel" predicate. The following rule will detect this pattern and infer the presence of a "definite quantification":

```
[ quant_definite ]
if
  ( the mrs elementary predication P0
      is an instance of the mrs predicate '_the_q_rel' and
      has the thing T as zeroth domain argument and
      equals modulo quantifiers the mrs elementary predication P1 ) and

   ( the mrs elementary predication P1
      is an instance of the mrs predicate MRS and
      has the thing T as zeroth domain argument ) and
```

```
( it is false that there is an mrs elementary predication named C that
   is an instance of the mrs predicate 'card_rel' and
   has the thing T as first domain argument )
then
 ( there is a definite quantification named Q that
     is on the thing T and has the mrs predicate MRS as sense ).
```

The definite quantification will be "on" a specific thing (x9 in the above example), indicating that it is x9 that is the definite thing. In addition the quantification will hold the specific predicate that defines the type ("_cat_n_1_rel") of the definite thing. In the pattern above there is an additional premise, that there is no additional "card_rel" predicate on the thing x9; a justification of this additional premise will not be attempted here, but it can be seen that a degree of knowledge of the linguistic nature of the MRS is necessary, and it is the need for this knowledge that we are attempting to abstract out. It is the intention that the builder of rules that use this information downstream will not have to work at the raw MRS level, but instead use the intermediate level.

In this example, the application of the rule will lead to the following fact being inferred:

```
there is a definite quantification named q1 that is on the thing x9 and has the mrs predicate
'_cat_n_1_rel' as sense.
```

Currently we perform other transformations into intermediate MRS structures, including:

- an indefinite quantification, indicating an entity of a certain type, but where the specific entity is unknown, such as in "a cat"
- a group quantification, indicating a set of entities, possibly with a cardinality, such as in "three cats"
- a set quantification, also indicating a set of entities, where the specific entities are stated, such as in "the Azuregroup and the Browngroup". The set may be a conjunction, as above, or a disjunction, as in "the Azuregroup or the Browngroup". This transformation, in particular, is complex involving a recursive traversal of a tree of MRS elementary predications.

These transformations have been implemented by CE rules, but cover a relative small proportion of the possible patterns to be found in the MRS output. It is planned to extend the rules to cover the MRS test suite [11] to ensure that the majority of outputs are handled.

3.5 Generic Semantic Processing

After the intermediate MRS is constructed the resulting CE is passed to generic semantic processing, where situations are fleshed out with the entities performing the roles, and the general types of the entities involved are determined. In theory, tense information could be handled at this stage; this was achieved in previous research [12] but has not yet been undertaken in the current system.

3.5.1 Naming individuals

The most elementary generic processing is the construction of named entities from proper nouns. In the example above there is an entity named John, encoded in the MRS via the "named_rel" predicate with a "c argument" that holds the string encoding the actual name. In CE each entity has a unique id, necessary to reference the entity unambiguously. However the need for a unique entity causes issues in human readability, since identifiers like "p1234" are not easily readable or memorable. In English writing a number of devices are employed to avoid this problem whilst retaining uniqueness and most are based on the use of contextually based common names; thus a person named John may be introduced into the dialog and used

unambiguously for a while, until there is a need to introduce a different John, at which point a different strategy may be employed such as referring to "John Smith" Currently CE does not permit this use of contextual naming, although this is under consideration.

An intermediate position has however been implemented. Entities are still named uniquely but may also have a "common name" attribute, as in "the person p1234 has "John" as common name". This allows the possibility of displaying the information about John, based on the common name, but there is no guarantee that this common name is unique. Several approaches to the construction of the common name have been explored, and two will be given below. The simplest approach to the inference of the common name from the linguistic information is shown below, which matches the "named_rel" and c argument structure.

```
[ mrs_propernoun ]
if
  ( the mrs elementary predication P is an instance of the mrs predicate ‘named_rel’ and
  has the thing T as zeroth argument and
  has the value C as c argument )
then
  ( the thing T has the value C as common name ).
```

inferring such sentences as "the thing x7 has "John" as common name" in the above example.

This processing is sufficient to allow limited use of common names as well as the unique identifier. A more sophisticated analysis may be achieved by maintaining a set of reference entities, such as well known places, organisations and people. Each reference entity has a unique id for disambiguation and one or more common names, which may be used to match against the entities derived from NL sentences via the common name. Thus "JohnSmith123" might be such a reference entity, defined as a person using the following CE sentence:

```
there is a person named JohnSmith123 that has "John" as common name and is a reference
entity.
```

It may be noted that JohnSmith123 is categorised as a "reference entity" as well as being a person. Not all "person"s are reference entities, only those that are considered by the user to be significant, well-known and therefore identifiable in the text by proper names. These sentences specifying the reference entities are constructed in advance by the user as background knowledge. It is then possible to construct rules that notice the match of common name, and infer that "the thing x7 is the same as the person JohnSmith123", where "is the same as" is recognised by the CE system as meaning that two entities are identical and all attributes and relationships are to be shared across both entities. This would result in the thing x7 being typed as a person. However, the use of "same as" processing can lead to problems, and so for the ELICIT task, a different approach has been taken, which will be described later.

3.5.2 Categorising individuals

A definite quantification on a thing also contains the MRS predicate that gives the conceptual type of the thing; in the example this is "_cat_n_1_rel". The first step in transforming the linguistic semantics into domain semantics is to convert between the MRS predicate and the corresponding domain concept. These conversions must be established from the user's view of the meaning of the concepts, since the user is ultimately the author of the conceptual model, and hence the only authoritative source of the concepts. The relevant link is established by an "expresses" relation, such as:

```
the mrs predicate "_cat_n_1_rel" expresses the entity concept 'feline'
```

where feline is the relevant concept, established perhaps by the statement:

conceptualise a ~ feline ~ F that is an animal.

A rule may now be constructed to perform the mapping from the definite quantification to the concept. Such a rule must use meta-logic to apply the concept name to the thing in the definite quantification, in effect categorising it as an instance of that concept. A suitable rule is:

```
[ mrs_noun_def ]
if
   ( there is a definite quantification named Q that is on the thing T
         and has the mrs predicate MRS as sense ) and
   ( the mrs predicate MRS expresses the entity concept EC )
then
  ( the thing T realises the entity concept EC ).
```

where the key step is the relationship in the conclusion, indicating that the thing T is an instance of whatever concept was matched by the variable EC, which in turn was defined in the "expresses" statement. The result is the inference of the type of x9:

the thing x9 is a feline.

This inference step will also be applied to the situation e3, given an expresses link between the MRS predicate "_chase_v_1_rel" and the entity concept "chase situation", giving:

the situation e3 is a chase situation.

Note that the verb chase has been turned into an entity, the chase situation, thus the action of the verb has been reified as an object, i.e. a situation.

3.5.3 Adding roles to situations

A situation has a set of associated entities that perform roles, such as the thing "doing" and the thing "being done to". These may be recovered from the MRS extracted from the sentence, either directly from the first and second arguments of the top level situation predicate (as is the case for e3 in the example) or from predicates derived from prepositional "attachments" to the main verb. The sentence "John chases the cat" is an example of the first case, with John playing the first role and the cat playing the second role. The sentence "the Lion works with the Azuregroup" is an example of the second case, where the Lion plays the first role and the Azuregroup plays the second. Knowing which entity plays which role depends upon knowing how a particular verb uses linguistic structures such as direct objects or prepositional attachments, and this is a complex issue in linguistics [13]. The ERG does not provide specific markers to help determine this, so additional information must be provided to the CE system in order to determine the entities playing the roles. In previous ITA research [12] VerbNet, [14], was used for this purpose, as it provides a database of verb "frames" that assign roles for each verb according to the phrase structure of a sentence containing the verb.

In the current research VerbNet is not being used and knowledge about the roles is contained in simple CE sentences about the situations. The first role of a situation is always taken to be the first argument of the elementary predication whose zeroth argument is the situation itself. In the example, the first role will be x7 whose common name is John. The source of the second argument is determined from the "second role source" attribute of the corresponding mrs predicate; in the case of "_chase_v_1_rel" this has the value of "second argument" which indicates that the second role is the feline x9. There are other possible values for the second

role source, such as "with" in the case of the mrs predicate _work_v_1_rel'. The result is that the following roles of the chase situation are inferred (we do not give the rules here, see [15]):

the chase situation e3 has the thing x7 as first role and has the feline x9 as second role.

This approach is sufficient for processing the simplified ELICIT sentences, but it may be necessary to include the VerbNet information back in the CE processing, if the current approach is found to be insufficient in other cases.

3.5.4 Domain Semantics

The situation with roles described above describes the state of affairs correctly, and could be used in further inferencing, but is not an elegant way to express the information. It is possible to construct sentences that describe the relationship between the roles more directly. In CE we can conceptualise a relationship such as "chases" between two things, as follows:

conceptualise the agent T ~ chases ~ the thing T1.

given that an "agent" classifies something as having intent, or will, as noted above. This allows the construction of a CE sentence such "the agent x7 chases the feline x9". This requires two additional inferences: that there is a "chases" relationship between the two entities and that x7 is an agent (not just a thing).

The first inference is described in two stages, for ease of understanding. A situation can be viewed in two different ways, as an object with parts (a chase situation with roles) or as a relationship between two things (chases) and the first inference is effecting a transformation between the object and relationship views. As noted above the object version reifies the relational version, and each type of situation in the conceptual model has a definition of the equivalent relational concept that it reifies, such as:

the entity concept 'chase situation' reifies the relation concept 'chases'.

The following rule uses meta-logic to determine the relational view of each situation:

```
[ gen_situation_reify ]
if ( the situation S has the thing T1 as first role and
       has the thing T2 as second role and
       realises the entity concept EC ) and
   ( the entity concept EC reifies the relation concept RC )
then
   ( the situation S is viewed relationally as the relation concept RC ).
```

matching against situations that have two roles, and thus can be relational, and which are instances of an entity concept (EC) that reifies a relational concept RC. Application of this rule will generate the inference:

the situation e3 is viewed relationally as the relation concept 'chases'.

The next step is to turn such a relationally viewed situation into an instance of this relation concept, which is achieved by the rule:

```
[ gen_vr ]
if ( the normal situation S is viewed relationally as the relation concept RC and
   has the thing T1 as first role and
   has the thing T2 as second role )
```

then
(the relation concept RC has
the sequence (the thing T1 , and the thing T2) as relation realisation).

(where the meaning of "normal situation" is described in section 4.1.2). This rule matches against a relationally viewed situation with the two roles, and creates an instance of the relationship itself (defined by a pair of things held in the sequence construct). The result is:

the thing x7 chases the thing x9.

As noted above it is also possible to infer the type of the arguments of a relationship and this second inference is generated from the conceptual model itself, since the conceptualise statement for a relationship also defines the range and domain of the relationship. For example the "chases" relationship is defined above to have the entity concept "agent" as the domain (i.e. the type of the first argument). A rule to make the inference of the type of the first role of the situation is as follows:

[gen-relation-domain]
if (the situation S has the thing A as first role and
is viewed relationally as the relation concept RC) and
(the relation concept RC has the entity concept EC as domain)
then
(the thing A realises the entity concept EC).

where the domain concept EC of the relationship is that of its domain attribute. This results in

the agent x7 chases the thing x9.

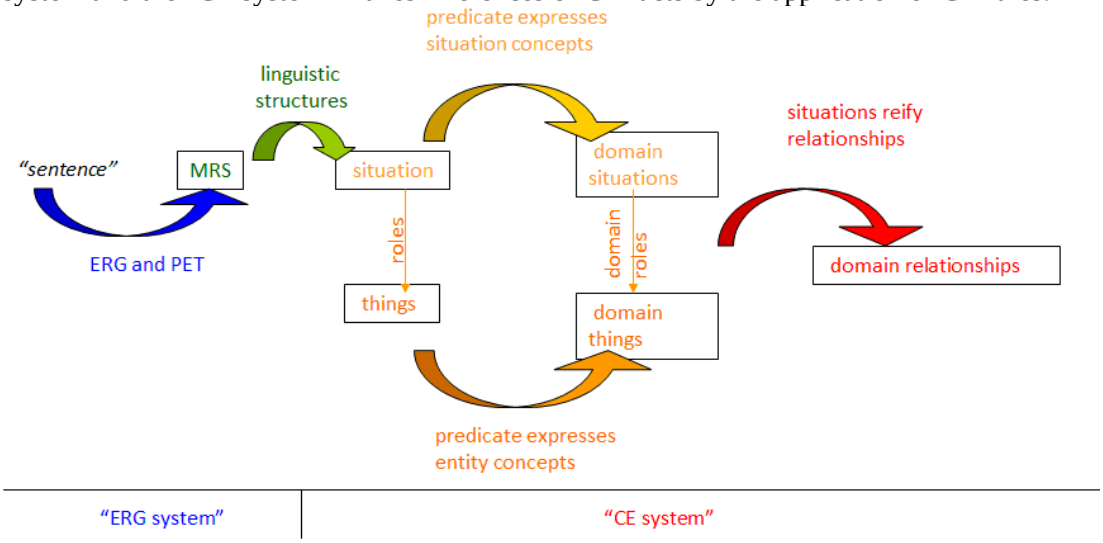
and given that we know the type and common name of x9, from above, the final result is:

the agent x7 has "John" as common name and chases the feline x9.

which is as far as it is possible to go without the use of references entities (that could indicate that the agent x7 is actually a person.)

3.5.5 Summary of transformations

The processing described above is summarised below, where the "ERG system" is the NL system and the "CE system" makes inferences of CE facts by the application of CE rules.



4 Simplified Sentences

As described above, we have chosen, for now, to have a human simplify the sentences to remove ambiguities that need common-sense reasoning to disambiguate. The result is a set of sentences that are still free-form English, which still need NL processing to turn them into CE facts. The complete set of simplified sentences that relate to solving the WHO component are given below, together with the original sentences:

Original ELICIT sentence	**Simplified sentence**
The Lion is involved	The Lion is a participant[2]
The Lion attacks in daylight	The Lion operates in the daytime.
The Azure, Brown, Coral, Violet, or Chartreuse groups may be planning an attack.	The Azuregroup may be a participant. The Browngroup may be a participant. The Coralgroup may be a participant. The Violetgroup may be a participant. The Chartreusegroup may be a participant.
The Chartreuse group is not involved	The Chartreusegroup is not a participant.
The Purple or Gold group may be involved	The Purplegroup may be a participant. The Goldgroup may be a participant.
All of the members of the Azure group are now in custody	The Azuregroup is not operational
Reports from the Coral group indicate a reorganisation	The Coralgroup is not operational
The Brown group is recruiting locals - intentions unknown	The Browngroup is recruiting locals.
The Lion will not risk working with locals	The Lion does not work with locals.
The Azure and Brown groups prefer to attack at night	The Azuregroup operates in the nighttime. The Browngroup operates in the nighttime.
The Violet group prefers to operate in daylight	The Violetgroup operates in the daytime.
The Coral group prefers to attack at night	The Coralgroup operates in the nighttime.
The Purple group prefers to attack in daylight	The Purplegroup operates in the daytime.
The Brown group needs time to regroup	The Browngroup is not operational
The Lion is known to work only with the Azure, Brown, or Violet groups	The Lion only works with the Azuregroup and Browngroup and Violetgroup.

4.1 Extraction of CE Facts from the sentences

The simplified sentences were passed through the processing chain described in the previous section, involving the ERG system to parse the sentences and generate the MRS, followed by the CE system to apply CE-based knowledge to transform the MRS into domain based CE facts expressed in the sentence.

Even with the simplifications above, and with the reduced syntactic complexity, there are still some interesting semantic challenges in processing the sentences. In fact there is no sentence that is as simple as the example above, although the extraction of a basic two argument relationship is central to the processing of many of them. A summary of the linguistic phenomena involved in the processing of the sentences is given below:

- "The Lion is a participant" involves a definition
- "The Browngroup is recruiting locals" involves a transitive verb and special treatment of the generic noun "locals"

[2] Actually it is not difficult to make the original sentence work as is, but we do not pursue this here

- "The Coralgroup operates in the nighttime" involves a prepositional attachment and special treatment of the noun phrase "the nighttime"
- "The Chartreusegroup is not a participant" involves a definition in negated form
- "The Purplegroup may be a participant" involves a definition in modal form
- "The Lion only works with the Azuregroup and Browngroup and Violetgroup." involves a prepositional attachment, a conjunction and the word "only" that indicates the sentence is expressing a general rule rather than a fact.

A sample of these linguistic phenomena and how they are processed will be given below. It is first necessary to explain that the method of linguistically processing named individuals in the ELICIT task is slightly different to that explained above, in order to avoid the complexities of the "same as" processing. Essentially the two steps of inferring the common name from the "named_rel" and the lookup against the reference entities are done in a single step. The result is placed into new attributes of the mrs elementary predication, called zeroth domain argument, first domain argument and second domain argument, corresponding to the first argument, second argument and third argument, respectively. Thus in the case of the example above, the elementary predications would now include:

the mrs elementary predication #ep2 is an instance of the mrs predicate '_chase_v_1_rel'
and has the situation e3 as zeroth domain argument
and has the thing JohnSmith123 as first domain argument
and has the thing x9 as second domain argument.

where the zeroth and second domain arguments are identical to the original zeroth and second arguments, but the first domain argument contains the result of looking up the common name "John" and finding the reference entity JohnSmith123.

All subsequent processing, including the identification of the roles, now works off the domain arguments rather than the original arguments. The result is effectively the same as before, but the attribute names in the rules and diagrams below use the domain argument format. This approach still allows the possibility of inferring the name of entities via other means.

For the ELICIT processing, a set of reference entities has been created for each of the types "group", "operative" and "country". An example of an "operative" reference entity is:

there is an operative named Lion that has "Lion" as common name and is a reference entity.

where the concept of "reference entity" is described in section 3.5.1. Two points should be noted. Firstly the definition of an operative such as Lion does not presuppose that it is involved in the attack. Secondly, for all the reference entities the identifier (as defined by "named XXX") is the same as the common name; strictly speaking this is not necessary, but it simplifies some of the later processing.

The result of extracting facts from the simplified sentences is summarised in Appendix A.

4.1.1 Handling a definition

In the sentence "The Lion is a participant", there is a definitional construction (is a), which is analysed by the ERG into the following MRS tabular form:

the ep #ep0 |_the_q_rel				
	has as zeroth argument			x 7
the ep #ep1 |named_rel				

	has as zeroth argument			x 7	
	has as c argument	Lion			
the ep #ep2 |_be_v_id_rel					
	has as zeroth argument		the situation e3		
	has as first argument			x 7	
	has as second argument				x9
the ep #ep3 |_a_q_rel					
	has as zeroth argument				x9
the ep #ep4 |_participant_n_1_rel					
	has as zeroth argument				x9

which shows the main situation as being a '_be_v_id_rel' predicate that links the structure for "the Lion" (x7) to the structure for "a participant" (x9). The "_a_q_rel" is a possibility for another type of intermediate mrs quantification, but this has not yet been implemented. Instead a rule specific to this pattern has been constructed:

```
[ mrs_is_defn ]
if
  ( the mrs elementary predication EP
      is an instance of the mrs predicate '_be_v_id_rel' and
      has the normal situation S as zeroth domain argument and
      has the thing SUBJ as first domain argument and
      has the thing PRED as second domain argument ) and
  ( the mrs elementary predication EP1
      is an instance of the mrs predicate '_a_q_rel' and
      has the thing PRED as zeroth domain argument ) and
  ( the mrs elementary predication EP2
      is an instance of the mrs predicate CATEGORY and
      has the thing PRED as zeroth domain argument ) and
  ( the mrs predicate CATEGORY expresses the entity concept EC )
then
  ( the thing SUBJ realises the entity concept EC ).
```

The key step in this rule is the picking up of the CATEGORY ("_participant_n_1_rel" in this example) and asserting that the SUBJ (x7) is an instance of the concept that is expressed by the category (in this domain the concept linked by the "expresses" relationship is 'participant'). Applying this rule leads to the inference of:

```
the thing Lion is a participant
```

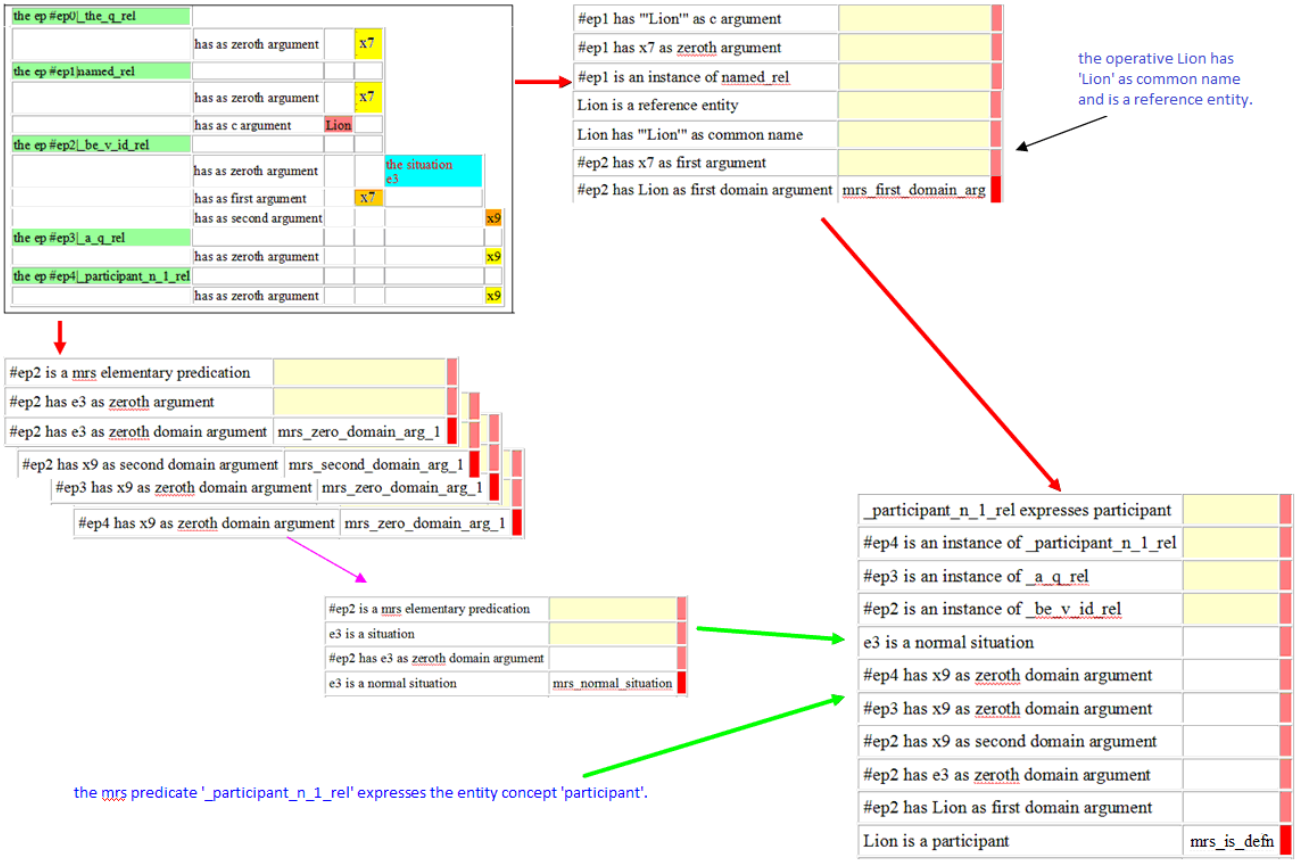
and given that the entity Lion is already defined (as the reference entity) as being an operative, then the final result is the extracted fact:

```
the operative Lion is a participant.
```

The "mrs_is_defn" rule includes a premise that requires the situation to be a "normal situation". The meaning of this concept is described in the next example, for now it may be taken that this situation is a normal one.

As noted previously, we are exploring the use of proof tables to visualise the reasoning. Proof tables can represent multiple reasoning steps in a single table, but it is also possible to "cut

up" the proof table into individual reasoning steps, and then to lay out the steps in a graphical form. This layout is not yet done automatically, and a manual version of the layout of the total reasoning in this example is given below:



This starts with the MRS tabular form at the top left and progresses through individual proof tables through to the conclusion (Lion is a participant) at the bottom right. The reasoning that constructs "the operative Lion" occurs in the top right, and the default mapping of argument to domain argument for the other entities occurs in the middle left. Finally all of the relevant information, together with the "expresses" link to the concept 'participant', are gathered up and applied by the "mrs_is_defn" rule to infer the conclusion.

4.1.2 Handling negation and modality

In the sentence "The Azuregroup may be a participant", the situation is possible but not definite. This is expressed in the MRS as an additional elementary predication MP with "_may_v_modal_rel" as predicate, and there is an "equal modulo quantifiers" link from MP to the elementary predication SP representing the top level situation. The ERG system that generates the MRS in CE takes account of this and adds a CE fact stating "the mrs elementary predication MP is modalised by the mrs elementary predication SP". This in turn is visualised in the MRS tabular form highlighted as a grey box, as shown below:

the ep #ep0 |_the_q_rel						
	has as zeroth argument					x7
the ep #ep1 |named_rel						
	has as zeroth argument					x7
	has as c argument	Azuregroup				

the ep #ep3 |_be_v_id_rel						
	has as zeroth argument			the situation e12		
	has as first argument					x7
	has as second argument				x11	
	is modalised by		#ep2 |_may_v_modal_rel			
the ep #ep4 |_a_q_rel						
	has as zeroth argument				x11	
the ep #ep5 |_participant_n_1_rel						
	has as zeroth argument				x11	

A similar effect occurs with a negated sentence such as "The Lion is not a participant", and a "is negated by" relationship is planted between the relevant elementary predications[3], leading to the inference that the associated situation is a "negated situation".

For both the modalised situation and the negated situation, and especially with the latter, there is a philosophical issue as to whether these situations actually exist or not. Clearly some form of the situation exists, since the sentence is talking about it, but it is counter-intuitive to consider that a "negated situation" actually exists in the real world. Our approach to this problem is that there are different "types" of situation. Most situations are "normal", and exist exactly as stated; the sentences "John chases the cat", and "the Lion is a participant" are examples of normal situations that "really exist". However situations that are modalised or negated (and constrained as will be described later) are considered to be abnormal situations; more specifically they are conceptualised as "modified situations". In the CE rules that map the linguistic semantics into domain semantics, this is taken into account and in some cases different processing pathways are taken for normal and modified situations.

Elementary predications that are linked to by the additional relations (eg "is negated by") are called "marked elementary predications", and situations that are the zeroth argument of such predications are the modified situations. The different types of modified situation may be recognised by the type of additional relations and rules have been constructed to infer the situation types from these relations. However normal situations are recognised only by virtue of being an argument of an elementary predication that is **not** a marked elementary predication and this requires a negated proposition in the premise of the rule:

```
[ mrs_normal_situation ]
if
  ( there is an mrs elementary predication named P that
      has the situation S as zeroth domain argument) and
  ( it is false that the mrs elementary predication P is a marked elementary predication )
then
  ( the situation S is a normal situation ).
```

Care must be taken to ensure that this rule is run **after** the rules inferring the presence of marked elementary predications, in effect a closed world assumption must be made that the

[3] Currently the ERG appears to construct the wrong predicate for a negated situation, this may be due to fact that we are using the older (1111) version of the ERG.

total set of the marked elementary predications is known. The ordering of the application of rules is ensured by grouping the rules into "rulesets" and then explicitly defining the order in which the rulesets are run. Currently this meta-information is held in the Prolog code, but we aim to allow the user to define the rulesets and rule ordering via CE statements in the future.

In the example sentence "The Azuregroup may be a participant", the processing follows a similar path to that for "The Azuregroup is a participant", as was diagrammed in the previous section, except that the processing to handle the modal auxillary "may" described in this section leads to the top level situation (e12) being a modalised situation rather than a normal situation. This leads to a slightly different rule to be applied, "mrs_is_defn_modal" rather than "mrs_is_defn":

```
[ mrs_is_defn_modal ]
if
  ( the mrs elementary predication EP
      is an instance of the mrs predicate '_be_v_id_rel' and
      has the modalised situation S as zeroth domain argument and
      has the thing SUBJ as first domain argument and
      has the thing PRED as second domain argument ) and
  ( the mrs elementary predication EP1
      is an instance of the mrs predicate '_a_q_rel' and
      has the thing PRED as zeroth domain argument ) and
  ( the mrs elementary predication EP2
      is an instance of the mrs predicate CATEGORY and
      has the thing PRED as zeroth domain argument ) and
  ( the mrs predicate CATEGORY expresses the entity concept EC ) and
  ( the entity concept EC is modal to the entity concept MODEC )
  then
  ( the thing SUBJ realises the entity concept MODEC ).
```

where the differences between non-modalised and modalised rules are underlined. Essentially the rule relies upon a statement of the relationship between the "basic concept" (participant) and its modal equivalent (possible participant). This relationship, although knowable from the linguistics, is not knowable from the concept terms in the conceptual model, since no implied semantic relations between concept names are to be inferred from the language used in the terms. Thus it is necessary for the user (or model builder) to explicit state such semantic relations, such as;

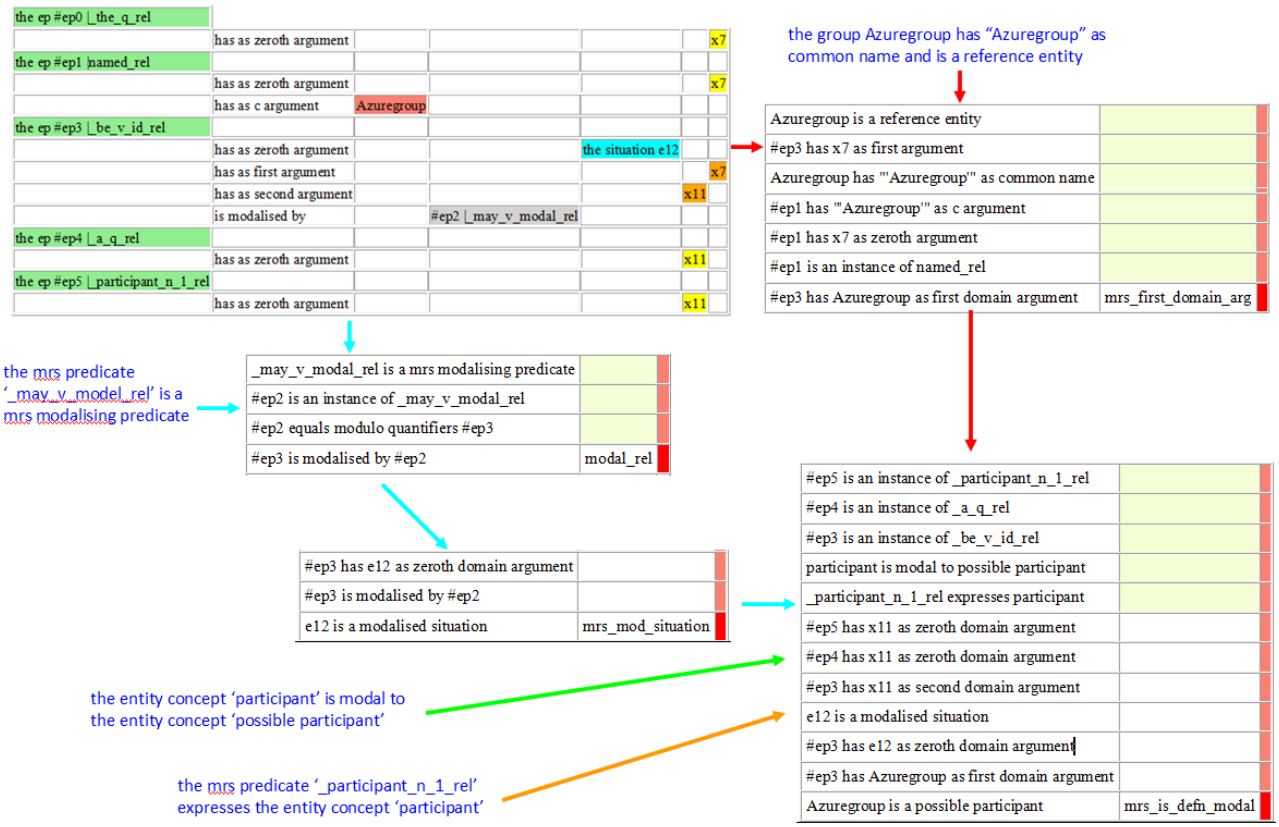
the entity concept ‘participant’ is modal to [4]the entity concept ‘possible participant’.

Thus the rule implements the intuition that if the situation being expressed is a modalised situation, then the "basic concept" should be expressed in a modal way. Its application results in the inference:

the group Azuregroup is a possible participant.

The total reasoning may be summarised in the following diagram:

[4] Intuitively this seems the wrong way round, and it may be better to represent it as "the entity concept ‘possible participant’ is modal to the entity concept ‘participant’".



where the MRS generated by the ERG system is on the top left and the conclusion (the group Azuregroup is a possible participant) is on the bottom right, generated by the "mrs_is_defn_modal" rule. This takes input from the analysis of Azuregroup as a reference entity (top right) and the inference that situation e12 is a modalised situation (middle). (This inference also involves the information that the key predicate "_may_v_modal_rel" is one that signals a modalising relationship). The "mrs_is_defn_modal" rule also takes input from the semantic "is modal to" relationship between 'participant' and 'possible participant'.

4.1.3 Handling "the daylight"

The sentence "The Lion operates in the daytime" is mostly handled in the same way as previous examples, but the prepositional attachment "in the daytime" causes several complexities. The MRS output is:

the ep #ep0 |_the_q_rel								
	has as zeroth argument							x7
the ep #ep1 |named_rel								
	has as zeroth argument							x7
	has as c argument	Lion						
the ep #ep2 |_operate_v_1_rel								
	has as zeroth argument				the situation e3			
	has as first argument							x7
	has as second argument					p9		
the ep #ep3 |_in_p_rel								
	has as zeroth argument			the situation				

				e10			
	has as first argument				the situation e3		
	has as second argument						x11
	has as second argument[5]		daytime				
the ep #ep4 |_the_q_rel							
	has as zeroth argument						x11
	has as zeroth argument		daytime				
the ep #ep5 |_daytime_n_1_rel							
	has as zeroth argument						x11
	has as zeroth argument		daytime				

The predicate "_operate_v_1_rel" is converted into a (normal) "operating situation" via an expresses link:

the mrs predicate "_operate_v_1_rel" expresses the entity concept 'operating situation'.

and "the Lion" is converted into a reference entity. Furthermore the Lion is assigned to the first role of the operating situation, thus giving:

the operating situation e3 has the operative Lion as first role.

However "in the daylight" is not converted into a second role, as it is better analysed as an "adjunct" indicating containment (or location), rather than a "complement" indicating a role. At the generic semantic level, the preposition "in" is transformed into a containment relationship, via the rule "mrs_in_prep", in this case between the operating situation and a "container" (x11), leading to the inference:

the operating situation e3 is contained in the container x11.

The expression "the daylight" is analysed as a definite quantification on x11 with "_daytime_n_1_rel" as sense. If we followed the processing above, this would be analysed as a instance (x11) of an entity concept such as 'daytime', but we do not wish to follow that route here. This is because we want to model daytime (and the corresponding nighttime) as instances of the more general concept of time interval, thus allowing the expression "the time interval daytime" and "the time interval nighttime". More importantly we wish to define temporal relationships between these instances, for example the key fact:

the time interval daytime does not overlap the time interval nighttime.

In CE it is not possible to say "the daytime" where "daytime" is a concept standing on its own, and an instance is always required, as in "the daytime x". But the latter expression cannot be involved in a simple "does not overlap" fact as above; instead a rule would be required informally stating that "all daytimes X and all nighttimes Y do not overlap", which is somewhat inelegant.

[5] There are two second arguments in the table because x11 has been asserted as "same as" daytime", so they are really the same thing.

Thus we wish to map the mrs predicate "_daytime_n_1_rel" onto an instance called "daytime" rather than a concept type:

the mrs predicate "_daytime_n_1_rel" has "daytime" as expressed instance.

This mapping is achieved via two reasoning steps. The first is:

```
[ mrs_noun_def_instance ]
if
   ( there is a definite quantification named Q that is on the thing T
         and has the mrs predicate MRS as sense ) and
   ( the mrs predicate MRS has the value ECI as expressed instance )
then
  ( the thing T has the value ECI as common name ).
```

which interprets the definite quantification (the daytime) on x11 as indicating a common name of "daytime" on the thing x11. We have also predefined the instance "daytime" as a reference entity of type "time interval":

the time interval daytime has "daytime" as common name and is a reference entity.

The second step implements "same as" processing:

```
[ nn_lookup ]
if ( the thing T has the value W as common name ) and
   ( it is false that the thing T is a reference entity ) and
   ( there is a reference entity named REF that
       has the value W as common name )
then
   ( the thing REF is the same as the thing T ).
```

which matches reference entities to things (that are not themselves predefined reference entities) by virtue of sharing the same common name, and inferring that they are actually the **same** thing. In this case x11 and the time interval daytime are inferred as being the same. As noted above the CE reasoning system automatically propagates properties between things that are the same as each other.[6] Recalling the containment relationship on x11 defined above, the following is therefore inferred:

the operating situation e3 is contained in the time interval daytime.

In the general semantics, being "contained in something that is a place" is inferred to be "located in" that place. The conceptualisation of a time interval is as a type of place (located in the time line), thus it is inferred that:

the operating situation e3 is located in the time interval daytime.

Putting all the information about e3 together, we have:

the operating situation e3 is located in the time interval daytime and has the operative Lion as first role.

[6] There is a philosophical question as to the status of things that are the same as each other: do the identifiers x11 etc actually correspond to unique individuals or are they descriptions of some underlying set of individuals?

Finally, in the domain layer, there is a specific rule that transforms this reified (and normal) situation into a more readable domain relationship, "operates in":

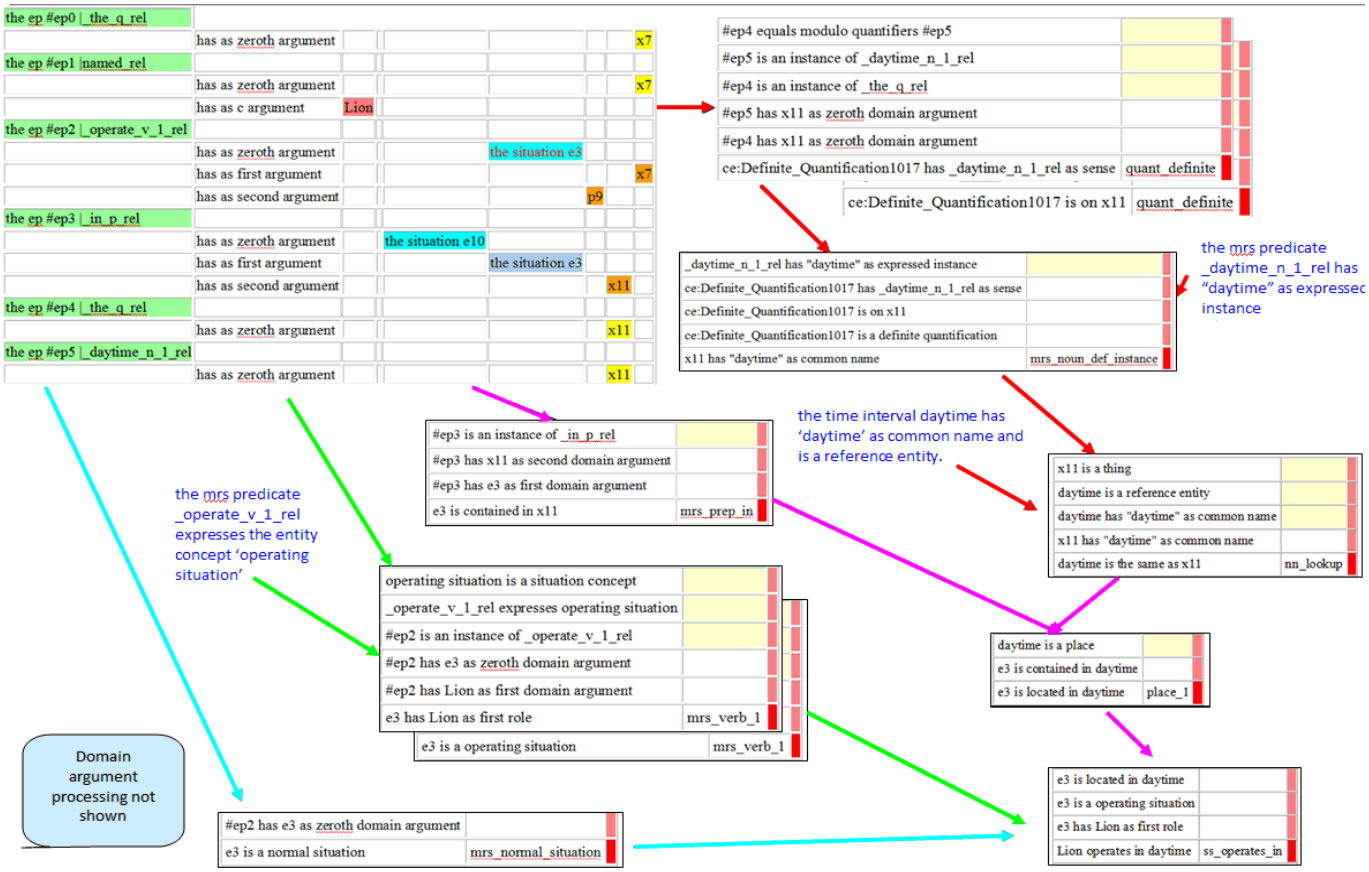
[ss_operates_in]
if
(the operating situation S has the agent T1 as first role and is located in the place L) and
(the situation S is a normal situation)
then
(the agent T1 operates in the place L).

resulting in the final sentence in the desired form:

the operative Lion operates in the time interval daytime

Similar processing occurs for sentences such as "the Azuregroup operates in the nighttime".

The following diagram summarises all of the above reasoning steps, from the MRS output in the top left to the conclusion in the bottom right.



4.1.4 Handling "locals"

In the sentence "The Browngroup is recruiting locals", there is a further semantic complexity. The MRS is shown below.

the ep #ep0 |_the_q_rel				
	has as zeroth argument			x7

the ep #ep1 |named_rel					
	has as zeroth argument			x 7	
	has as c argument	Browngroup			
the ep #ep2 |_recruit_v_1_rel					
	has as zeroth argument		the situation e3		
	has as first argument			x 7	
	has as second argument				x9
the ep #ep3 |udef_q_rel					
	has as zeroth argument				x9
the ep #ep4 |_local_n_1_rel					
	has as zeroth argument				x9

The sentence is mostly analysed as in previous examples, leading to a situation with two roles. The first role is straightforward and is inferred as:

the recruiting situation e3 has the group Browngroup as first role.

It is the second role that adds a linguistic and semantic issue. The term "locals" is represented in the MRS as an indefinite thing (via the quantifier predicate "udef_q_rel") and the key question is whether this indefinite "locals" is meant to be expressing a generic concept of "local people, as a whole" or a specific set of instances such as "the locals that were recruited by Browngroup". In the context of the set of ELICIT sentences, which includes "the Lion cannot work with locals" it seems clear that the first interpretation is correct, and that the "locals" referred to in both sentences relate to the same, unique, concept of "local people as a whole". In the problem solving step, we are going to infer that The Lion and the Browngroup cannot work together. For this interpretation to work, we must map "locals" onto the same entity in both sentences, such an entity representing the single unique concept of "some local agent", as will be described below.

However before that is done it is necessary to note that this first interpretation will not allow us to discriminate between different sets of locals. For example we may wish to say something specific about the locals that Browngroup recruited, e.g. the number of people, which we do not want to be said about the locals that some other group recruited. For this discrimination we need the second interpretation, where each occurrence of "locals" must be turned into a different entity. Of course this interpretation does not allow the matching of the locals recruited by Browngroup against those that the Lion will not work with.[7]

The first step in analysing the term "locals" is the recognition that it is an "indefinite quantification", using a rule that matches the combination of "udef_q_rel" and the predicate "_local_n_1_rel" on the thing x9. An indefinite thing will be represented as a "generic" instance of the type associated with the predicate. In this case the type associated with the predicate via an expresses relationship is 'local agent':

the mrs predicate "_local_n_1_rel" expresses the entity concept 'local agent'.

The name of the thing will be constructed to reflect a "generic entity" of that type, in this case "a local agent", leading to the term "the local agent named 'a local agent'". This processing

[7] An alternative, and possibly better, interpretation is in the form of a rule, such as "if x is a locals then the Lion will not work with x. This is similar to the approach in the next section, but has not been pursued here.

will transform **every** occurrence of the indefinite "locals" into the **same** instance as required by the first interpretation noted above. This new instance must be placed into the correct relationship with "recruit situation", in this case the second role. This is achieved first by the assignment of the new 'a local agent' to the second domain argument of the mrs elementary predication that stands for the recruiting situation, using the following rule:

```
[ indef_something_second_arg ]
if
  ( there is an mrs elementary predication named EP that
    has the thing T as second domain argument ) and
  ( there is an indefinite quantification Q that is a set quantification and
       is on the thing T and has the mrs predicate MRS as sense ) and
  ( the mrs predicate MRS expresses the entity concept EC ) and
  ( the value V = the constant 'a ' <> the entity concept EC )
then
  ( there is a thing named V that realises the entity concept EC ) and
  ( the mrs elementary predication EP has the thing V as second domain argument ).[8]
```

which both constructs the new entity V, called ' a local agent', and assigns it to the second domain argument. (The syntax X <> Y causes a concatenation of the values of X and Y into a single name). The result is:

```
the mrs elementary predication #ep2 has the local agent 'a local agent' as second domain
argument.
```

Then, via the rules that convert the domain arguments into roles, it is inferred:

```
the recruiting situation e3 has the local agent 'a local agent' as second role.
```

This combined with the group Browngroup being the first role, causes a domain specific rule to convert this reified (and normal) situation to a specific relationship between the roles:

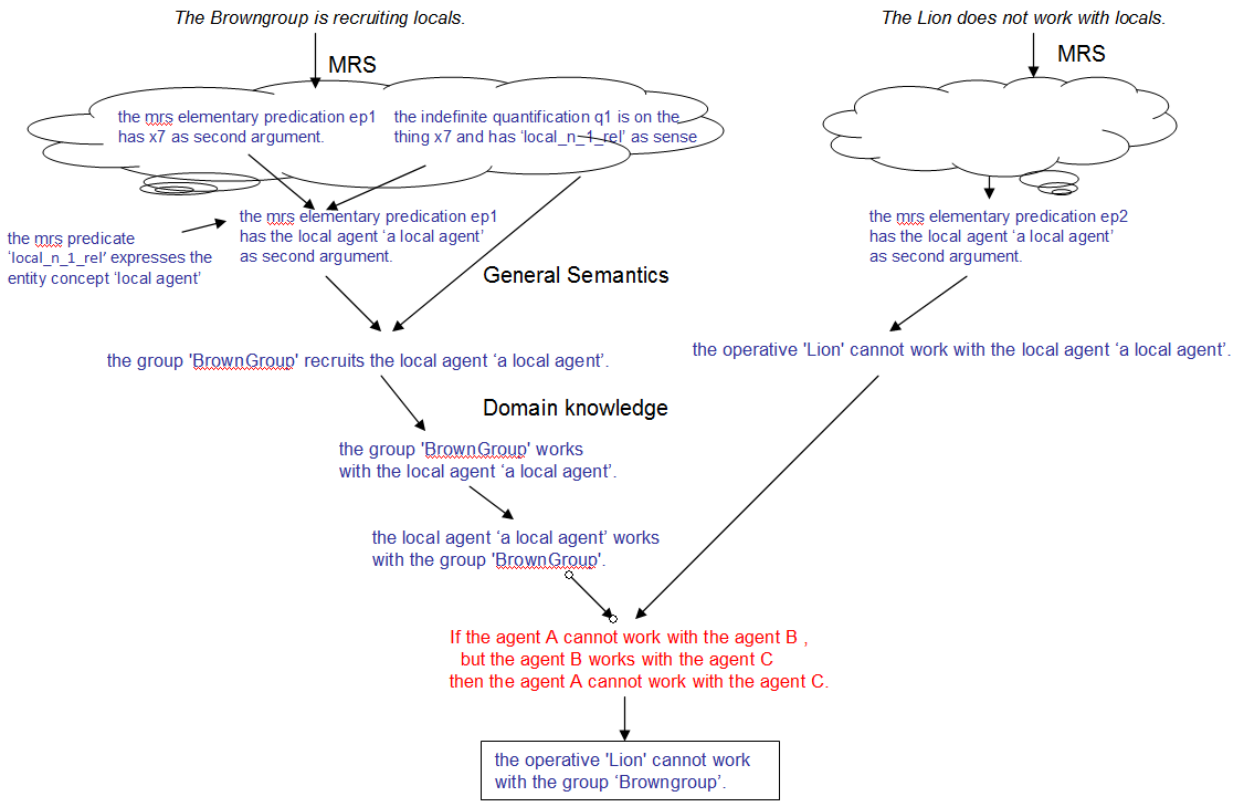
```
the group Browngroup recruits the local agent 'a local agent'.
```

The processing of the sentence "the Lion does not work with locals" leads via similar logic in respect of "locals" to the inference:

```
the operative Lion cannot work with the local agent 'a local agent'.
```

allowing domain inferencing to be made as to the fact that the Lion cannot work with the Browngroup, as will be described later. This processing is summarised as follows:

[8] The current CE Prolog does not permit two conclusions in a rule about different objects, although there is no logical reason not to do so, so for convenience this rule is shown as an amalgamation of two separate rules.



This diagram also shows the domain rule that leads to the inference of the non-working relationship between Lion and Browngroup.

4.1.5 Handling "only" and the generation of rules

The sentence "The Lion only works with the Azuregroup and the Browngroup and the Violetgroup." presents several significant linguistic and semantic issues, and is by far the most complex sentence considered so far. The two main issues are that there is a conjunctive phrase "the Azuregroup and the Browngroup and the Violetgroup" and that the use of "only" indicates that the sentence is indicating a rule rather than a simple fact. The handling of the conjunctive group is relatively simple and will be described first.

The MRS output for the conjunctive phrase is a recursive tree with its root at an elementary predication whose first argument that represents the group, and whose sub-nodes represent the elements of the group. The precise details of the MRS tree structure are not given here, but can be seen in [15]. Rules have been constructed to "walk" this tree and to generate a CE structure representing the group. This structure is based on a "conjunctive quantification" that is "on" the thing representing the group and "has as member" each thing that is in the group. In this example, assuming the reference entity processing has occurred, the result is:

```
there is a conjunctive quantification named q1 that
    is on the thing x7 and
    has the agent Azuregroup as member and
    has the agent Browngroup as member and
    has the agent Violetgroup as member.
```

For now this conjunctive quantification and its members will be put aside, and we focus on the rule-based component of the main sentence. For simplicity we will start with the simpler sentence "the Lion only works with the Azuregroup", and bring the conjunction in later. This

sentence presents a rule about the behaviour of the Lion, and seems to be re-expressable in at least two ways:

1. If there is a group G that is the Azuregroup then the Lion works with the group G[9]
2. If the group G is not the Azuregroup then the Lion does not work with the group G

However, the first seems to say more than is actually the case, i.e. that the Lion is actually working with the Azuregroup, whereas the sentence is really saying that the Lion could work with the Azuregroup. We will follow the second interpretation, which may be expressed as:

[only_rule]
if
(there is an agent named Lion) and
(the agent A is different to the agent Azuregroup)
then
(the agent Lion cannot work with the agent A).

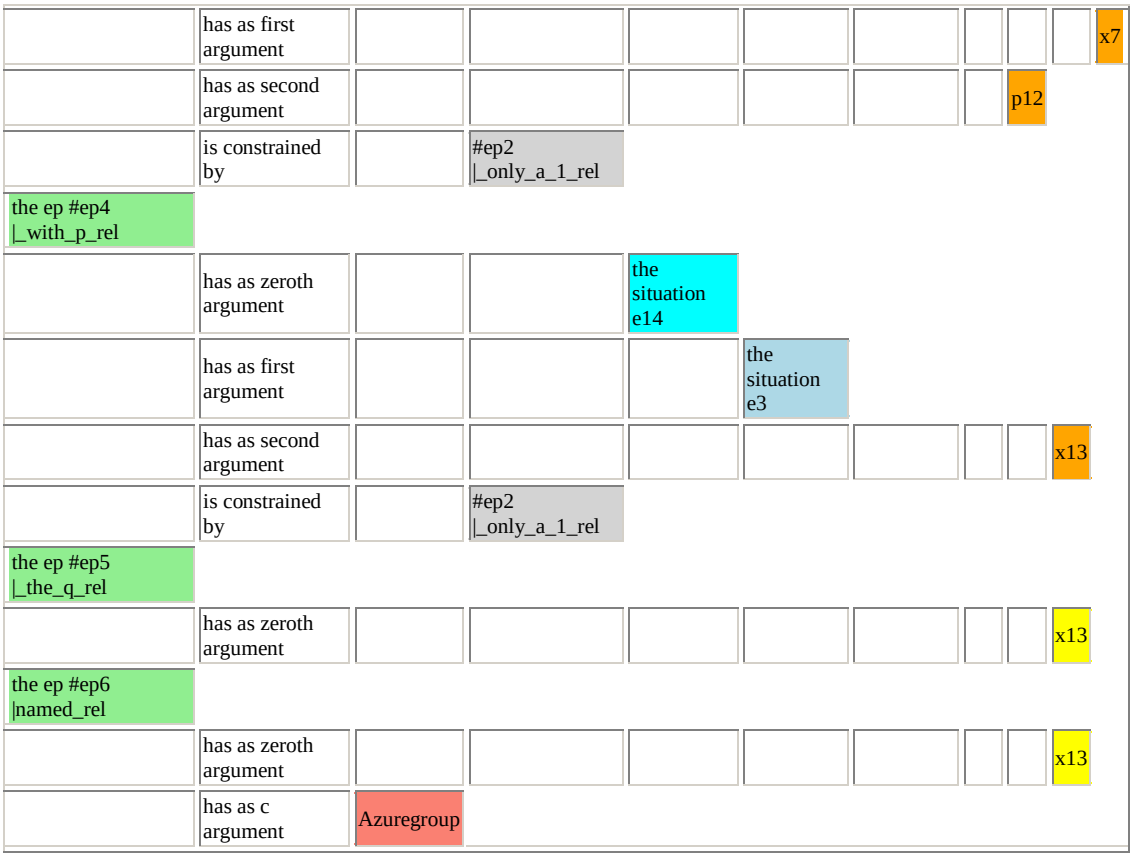
This rule has the additional premise that there is an agent named Lion. Whereas this seems obvious, it is added to ensure that the conclusion does not appear to infer a new piece of information (that there is an agent named Lion). If this extra premise is not present, the conclusion about who cannot work with the Lion will be unaffected, but there will be an additional justification for the fact that the Lion is an agent, based upon the premises of this rule (i.e. that the agent A is different to the agent Azuregroup), and clearly this justification is invalid. The problem arises, logically, because the conclusion would be presupposing the existence of the Lion as agent, and this not correct.

The relation "is different to" expresses the "is not" that occurred in the second interpretation above, with extra processing to handle the effects of the "same as" relation.

This rule, or something similar, is generated automatically by the CE system from the information from the ERG, as shown below:

the ep #ep0 |_the_q_rel										
	has as zeroth argument									x7
the ep #ep1 |named_rel										
	has as zeroth argument									x7
	has as c argument	Lion								
the ep #ep2 |_only_a_1_rel										
	has as zeroth argument						the situation e9			
	has as first argument							h10		
the ep #ep3 |_work_v_1_rel										
	has as zeroth argument					the situation e3				

[9] This formulation makes more sense when there is more than one group with which the Lion will not work (there is a group G that is the Azuregroup or the Browngroup …).



The rule-generation process occurs in two virtually parallel paths. The first path operates on the MRS for the underlying basic sentence equivalent to "the Lion works with the Azuregroup", using the techniques described earlier (reference entities, situation and role recognition), to end up with:

> the operating situation e3 has the operative Lion as first role and has the group Azuregroup as second role.

The second path uses the modification by the "_only_a_rel" predicate (as indicated by the fact the mrs elementary predication #ep3 is constrained by the mrs elementary predication #ep2[10]) to detect that the situation e3 is not a normal situation, but instead is a "constrained situation"[11]. The fact that the situation is constrained causes a different interpretation of the operating situation with its roles shown above, and it is this path that generates the desired rule. The path is itself implemented as a rule, and this will be shown, for ease of explanation in two versions, a specific and a generic version. To reduce confusion, the rule that is generated will be called the "only_rule" and the rule that generates this will be called the "meta-rule". The specific version of the meta-rule is shown below:

```
[ ss_works_with_only_neg ]
if
  ( the operating situation S is a constrained situation and
    has the agent T1 as first role and
    has the agent T2 as second role )
then
```

[10] The elementary predication #ep4 is also constrained in this way, but this is not used.
[11] This was originally called an "only situation", and it is not clear which is the best name from this type of situation.

```
( there is a logical inference named LI that
    has the statement that
        ( there is an agent named T1 ) and
        ( the agent "'A"' is different to the agent T2 )
    as premise and
    has the statement that
        ( the agent T1 cannot work with the agent  "'A"' )
    as conclusion
    ).
```

This matches to a constrained operating situation with its two roles and constructs an instance of a "logical inference" with a premise and a conclusion. A "logical inference" is a CE meta-concept that represents a CE rule, and there is a "rule-addition" command to the CE interpreter that causes instances of these concepts to be added as CE rules to the conceptual model, where they are then available for inferencing. The premise and the conclusion are both "statements", another meta-concept that contains one or more CE sentences, and may be used to build structures that contain CE sentences as values. Here they are being used to define the CE statements that are the premise and the conclusion of the "only_rule" that is being generated. Since these statements are in the scope of a rule (the meta-rule), they may contain variables, and these variables are bound to the values when the meta-rule is applied. In this example, the variables T1 and T2 are bound to the operative Lion and the group Azuregroup, since we are applying the meta-rule to the operating situation above.

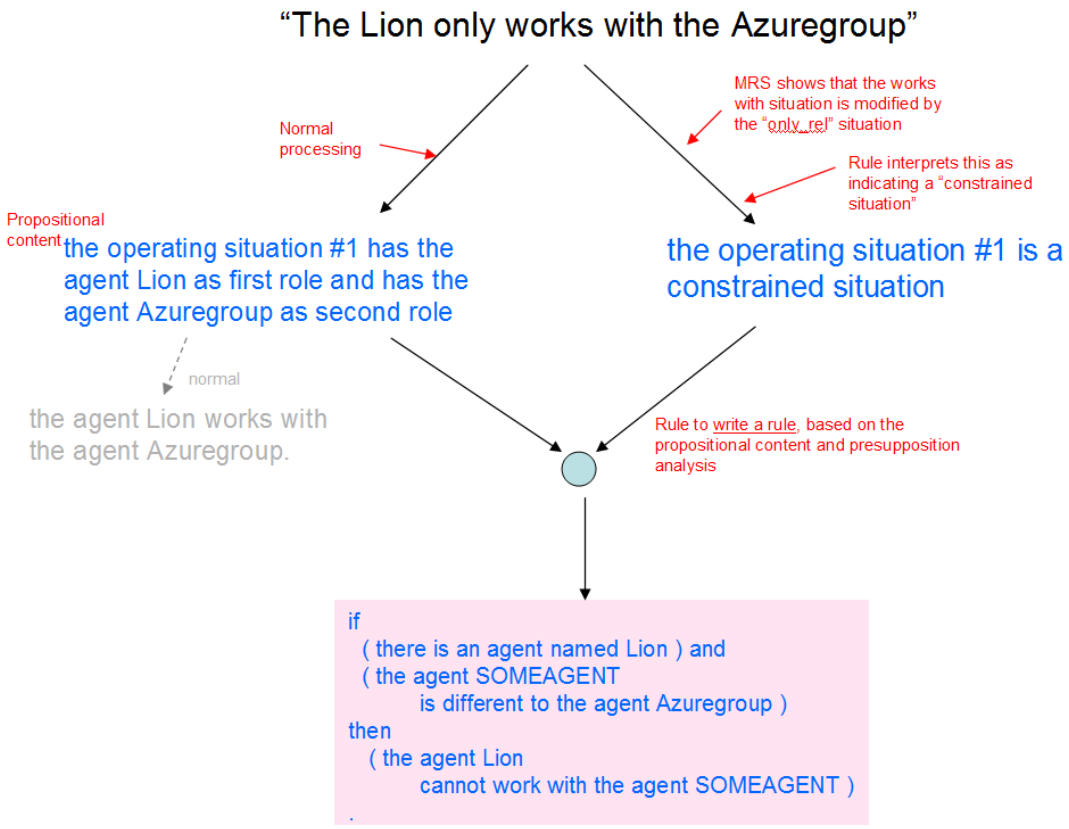
In the "only_rule", it can be seen that we need a variable A, which will be bound to the group when the "only_rule" itself is applied. However it is not possible to leave a variable A in the meta-rule, as this will lead to a conclusion with a floating unbound variable; instead we need to flag to the meta-rule that a constant is to be turned into a variable in the logical inference object. This is done by the use of the syntax "'XXX"' where the XXX is the name of the variable that is to be planted in the logical inference.

The result of applying the meta-rule is the construction of a logical inference structure:

```
there is a logical inference named l1 that
    has the statement that
        ( there is an agent named Lion ) and
        ( the agent "'A"' is different to the agent Azuregroup )
    as premise and
    has the statement that
        ( the agent Lion cannot work with the agent  "'A"' )
    as conclusion
```

which will turned into the "only_rule" when added to the conceptual model.

These processing paths to construct the "only_rule" are summarised in the diagram below.



The meta-rule above (ss_works_with_only_neg) is specific to the operating situation, and it is not reasonable to have to construct a meta-rule for each situation that might be involved in an "only" linguistic expression. However, using the meta-language of CE, it is possible to construct a generic version that will handle any situation. This meta-rule is more complex than the other rules that have been described. The simplest way to explain the rule is to focus on how the individual sentences in the specific version may be transformed into a meta-version that makes no reference to the "operating situation". This way to describe the meta-rule also has the benefit that it may lead to a more generic, and possibly automated, way to turn a specific rule into a generic rule.

The table below shows the correspondences in the sentences between the specific and generic. However a key piece of information necessary for the construction of the meta-rule should be noted; that the relationship in the conclusion ("cannot work with") is the opposite to the relationship that would have been used to represent the underlying proposition in the original sentence ("works with"). This is because we are turning the underlying sentence (the X only does Y with Z) into its opposite (if A is different to Z then the X does not do Y with Z). The semantic relationship of opposition is not automatically known from the terms in the conceptual model and must be given by the user, for example:

the relation concept 'works with' is opposite to the relation concept 'cannot work with'.

The correspondence table is as follows:

Specific meta-rule sentence	Generic meta-rule sentence	Comment
if	if	
the operating situation S is a constrained situation	the constrained situation S is viewed relationally as the relation concept RC	*The specific situation is turned into a pattern matching any constrained situation, and its relational view (e.g. 'works*

		with') must be picked up for use below
has the agent T1 as first role and has the agent T2 as second role	has the thing T1 as first role and has the thing T2 as second role	*The specific type (agent) is changed into the generic type (thing) since the type of the arguments of the situation is not known*
	the thing T1 has the value T1N as common name and the thing T2 has the value T2N as common name	*We actually need the common name part of the thing's description [12](Azuregroup etc) rather than the id, since it is the common name that is expressed in the "only" sentence. In many cases the common name and the id are the same, but in some cases, especially when the rule handling the conjunction is used (see below) they are not the same*
	the relation concept RC is opposite to the relation concept NOTRC	*The corresponding negative statement of the relationship (e.g. 'cannot work with') is determined from the user's specification*
	the relation concept NOTRC has the entity concept ECR as range and has the entity concept ECD as domain	*For use below, the concept type for the range and domain of the negative relationship ('cannot work with') is picked up*
	the value LI = the constant 'only_' <> the value S	*A unique name for the new rule is calculated as the concatenation of 'only_' and the id of the situation (e.g. e3).*
then	then	
there is a logical inference named LI that	there is a logical inference named LI that	*For the generic case, the new rule id is based on the name calculated above*
has the statement that (there is an agent named T1) as premise	has the statement that (the thing T1N realises the entity concept ECD) as premise	*The type of the first role T1 is not known in advance, so is dynamically calculated and is the same as the domain of the relational view of the situation. The premise in the specific rule has been broken out to simplify the comparison*
has the statement that (the agent "'A"' is different to the agent T2) as premise	has the statement that (the thing "'A"' realises the entity concept ECR) and (the thing T2N realises the entity concept ECR) and	*The "different to" relation is the same in both cases, but the type of the variable A and the second role T2 is dynamically calculated*

[12] This is not an elegant solution, and inelegance is a sign that there is something not right with the formulation; we will aim to address this in the future

	(the thing "'A"' is different to the thing T2N) as premise	
has the statement that (the agent T1 cannot work with the agent "'A"') as conclusion	has the statement that (the thing "'A"' realises the entity concept ECR) and (the thing T1N realises the entity concept ECD) and (the relation concept NOTRC has the sequence (the thing T1N , and the thing "'A"') as relation realisation) as conclusion	*A new instance of the opposite relation (' cannot work with') is created from A and T1, with the types defined dynamically*

For completeness the generic meta-rule is given below:

```
[ gen_vr_only_neg ]
if
  ( the constrained situation S is viewed relationally as the relation concept RC and
    has the thing T1 as first role and
    has the thing T2 as second role )  and
  ( the thing T1 has the value T1N as common name )  and
  ( the thing T2 has the value T2N as common name ) and
  ( the relation concept RC is opposite to the relation concept NOTRC ) and
  ( the relation concept NOTRC has the entity concept ECR as range and
        has the entity concept ECD as domain ) and
  ( the value LI = the constant 'only_' <> the value S )
 then
 ( there is a logical inference named LI that
      has the statement that
         ( the thing T1N realises the entity concept ECD )
      as premise and
      has the statement that
         ( the thing "'A"' realises the entity concept ECR ) and
         ( the thing T2N realises the entity concept ECR ) and
         ( the thing "'A"' is different to the thing T2N )
      as premise and
      has the statement that
         ( the thing "'A"' realises the entity concept ECR ) and
         ( the thing T1N realises the entity concept ECD ) and
         ( the relation concept NOTRC has
               the sequence ( the thing T1N , and the thing "'A"' ) as relation realisation )
      as conclusion
      ) .
```

It is now possible to explain how the basic meta-rule and group processing are combined, in order to analyse the sentence "the Lion only works with the Azuregroup and the Browngroup and the Violetgroup". The result is a rule of the form:

```
[ only_e3_491 ]
if
  ( there is a agent named Lion ) and
  ( the agent A is different to the agent Azuregroup ) and
  ( the agent A is different to the agent Browngroup ) and
  ( the agent A is different to the agent Violetgroup )
```

```
then
  ( the agent Lion cannot work with the agent A ).
```

which is similar to the first "only_rule" above, except that the premise "the agent A is different to the agent B" is iterated over each member of the conjunctive quantifier. This is achieved by a modification to the generic meta-rule so that it matches on the conjunctive quantifier as the second role of the operating situation rather than the single entity representing the Azuregroup, as in the simpler version of the "only_rule".

The key change is that the premise now includes the conjunctive quantification and its members M:

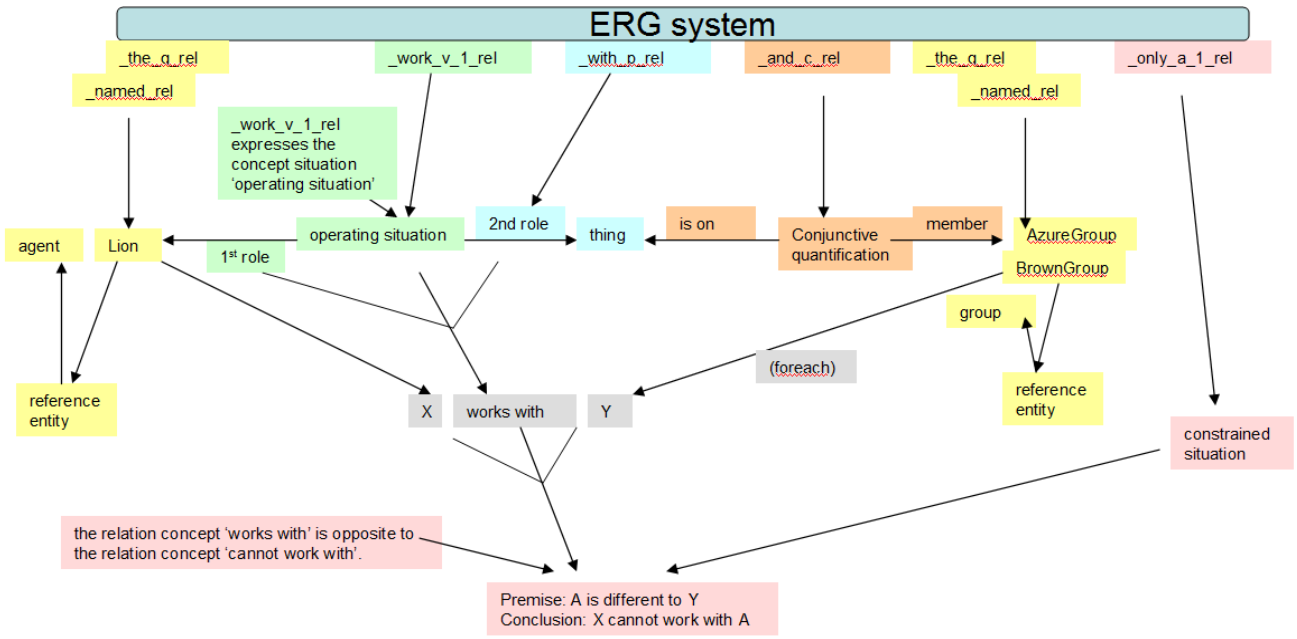
```
  ( the constrained situation S is viewed relationally as the relation concept RC and
    has the thing T1 as first role and
    has the thing T2 as second role )  and
  ( there is a conjunctive quantification named Q that is on the thing T2 and
      has the thing M as member ) and
  ( the thing T1 has the value T1N as common name )  and
  ( the thing M has the value MN as common name )  and

  …
```

This will match on **each** member M, causing the rule to generate a logical inference for each match, and hence for each member. However, the name of the logical inference will be the same each time (since it is generated from the same situation e3) so the information about the premise and conclusion sentences will be added to the **same** logical inference. Most of the sentences will contain identical information, and hence will be ignored. However the premise statement of the generated logical inference involves A being different to MN:

```
there is a logical inference named LI that has the statement that
  ( the thing "'A"' realises the entity concept ECR ) and
  ( the thing MN realises the entity concept ECR ) and
  ( the thing "'A"' is different to the thing MN )
as premise and
…
```

and this causes a **different** premise to be added to the "only_rule" for **each** member M (more strictly its common name MN), specifically that the variable A is different to this member M. It is the addition of "the thing A is different to the thing <M>" where <M> is replaced by the name of each member ("Azuregroup", "Browngroup" etc) that causes the creation of the set of correct premises for the "only_rule".

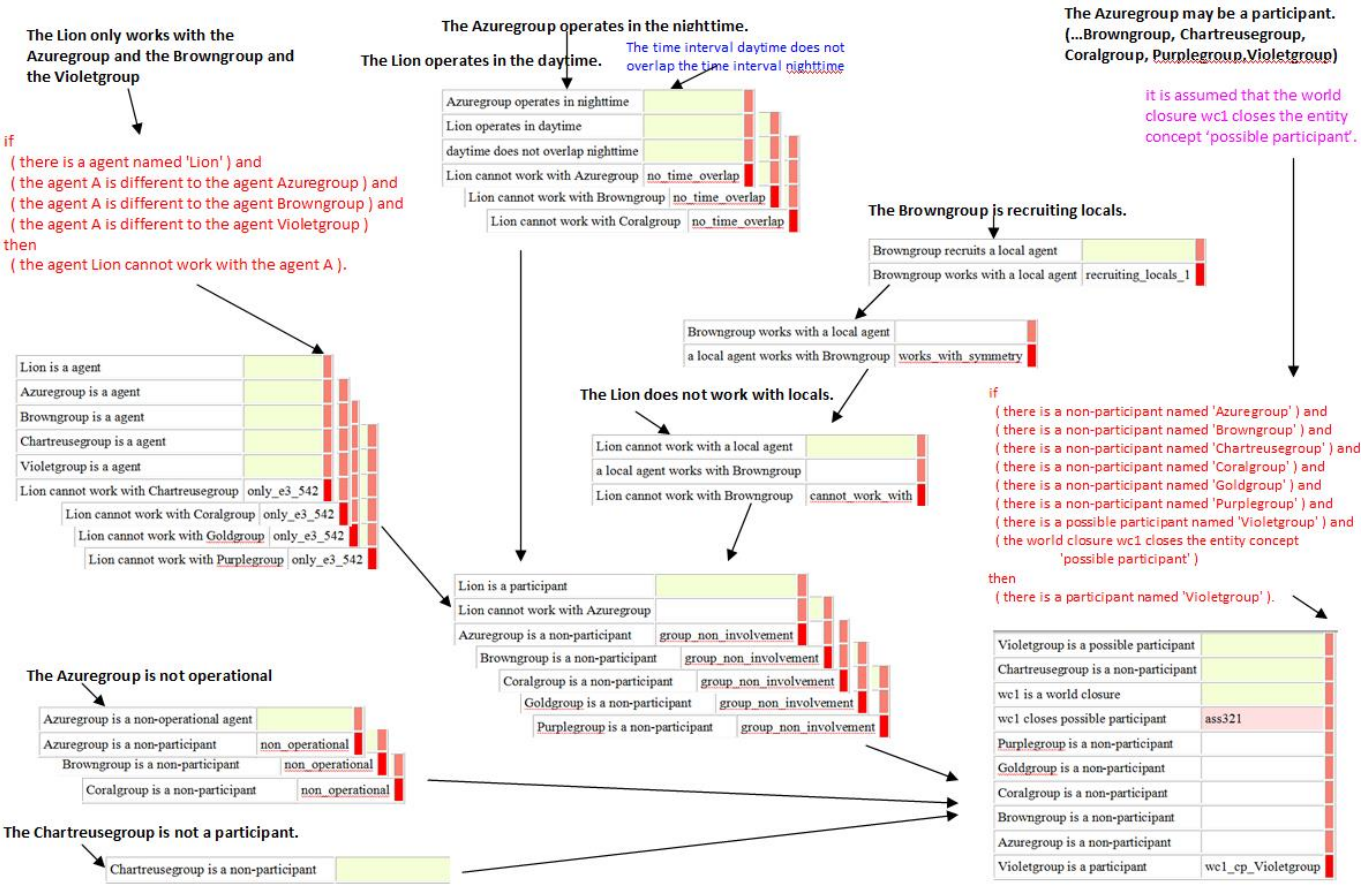
The full processing is summarised in the diagram below, where different colours are used to code the different processing steps; yellow for the construction of things matching the reference entities, green for the processing of the "operation situation" and its first role, blue for processing the 2nd role (as being the conjunction quantification), orange for the construction of the conjunctive quantification itself, pink for the recognition of the constrained situation and (at the bottom for the generation of the rule) and grey for the application of each member to the generic "works with" relationship.



5 Domain Processing to solve the ELICIT WHO task

The previous sections have explained how the simplified sentences are turned into CE facts and rules (only a single rule in this example, the "only_rule"), and the complete set of facts and rules so generated is given in appendix A. It is now possible to add these new facts and rules to those contained in the domain model and run the CE reasoning system to produce inferences about the attack situation. However it is not quite possible to solve the problem, as an additional step is required. The rules and facts we have generated from the sentences will allow the inference of one participant (the Lion) and a list of all of the groups that **cannot** be involved, but this is insufficient to solve the problem since it does not say who **was** involved. It is necessary to apply the final constraint-based reasoning step of inferring the group that is the participant on the grounds that all of the other possible participants have been ruled out, which in turn requires us to know the complete set of possible participants. It turns out that there is an issue in knowing the complete set, which will be taken up in section 6.3. We will consider the two steps (inferring the non-participants and inferring the participant) in turn. The application of these steps solves the ELICIT identification task for the WHO component.

A detailed account of all of the rules and reasoning steps involved will not be given (see [16,17] for more details), but a summary diagram based on proof tables is given below, and will be referred back to in the following sections.



5.1 Inferring the non-participants

For now it is assumed that the set of possible participants is defined by the original sentences: "The Azure, Brown, Coral, Violet, or Chartreuse groups may be planning an attack." and "The Purple or Gold group may be involved". As described above, this leads to CE sentences defining groups that are a "possible participant", where the groups are Azuregroup, Browngroup, Chartreusegroup, Coralgroup, Goldgroup, Purplegroup, and Violetgroup.

In addition, one of the extracted CE facts specifically states that the operative Lion is a participant. This causes other inferences to be made about the non-participation of other groups, to the extent that all but one of the possible participants is ruled out, as summarised below (note that there may be more than one reason to rule out a group):

Group	**Ruled out because**
Azuregroup	directly stated as being non-operational; operating time interval does not overlap with Lion's
Browngroup	directly stated as being non-operational; operating time interval does not overlap with Lion's; recruiting locals which Lion cannot work with
Chartreusegroup	directly stated as being non-participant; not one of the groups that Lion only works with
Coralgroup	directly stated as being non-operational; not one of the groups that Lion only works with; operating time interval does not overlap with Lion's
Goldgroup	not one of the groups that Lion only works with
Purplegroup	not one of the groups that Lion only works with
Violetgroup	

More details of the reasoning involved are given in the introductory sections above, but also in [16]. In the summary diagram above the reasoning to rule out the non-participants is shown as individual proof tables from left to right from a simplified sentence and converging on the final rule at the bottom right, which will be explained below. Some of the proof tables are "stacked" to reduce space, where the same reasoning is occurring on different entities.

5.2 Inferring the Group Participant

The table above shows the ruling out of the possible participants, leading to a set of facts of the form "the group X is a non-participant". It remains to infer the actual group participant, and in an earlier section on constraint-based reasoning it was noted that a set of "choice-point" rules could be constructed to infer the participant from the set of non-participants. As there are 7 group participants there are 7 rules in this set, and the relevant choice-point rule is:

```
[ wc1_cp_Violetgroup ]
if
  ( there is a non-participant named 'Azuregroup' ) and
  ( there is a non-participant named 'Browngroup' ) and
  ( there is a non-participant named 'Chartreusegroup' ) and
  ( there is a non-participant named 'Coralgroup' ) and
  ( there is a non-participant named 'Goldgroup' ) and
  ( there is a non-participant named 'Purplegroup' ) and
  ( there is a possible participant named 'Violetgroup' )
then
  ( there is a participant named 'Violetgroup' ).
```

Given the information about the non-participants, this rule will make the inference:

```
the group Violetgroup is a participant.
```

This inference is shown on the summary diagram on the right hand side. However there is an additional premise in the rule in the diagram, an assumption about the closed world, which will be described later.

This rule depends upon knowing the complete set of possible participants (and that there are no more possible participants) and this leads to several problems. Firstly, given that different versions of the ELICIT problem may have different participants, and that it is not reasonable to expect that the user will generate this by hand, some means of automatically generating it must be found. Secondly, when is the possible set of participants known, and specifically how is it known that there are no more participants of which we are unaware, and would thereby invalidate the reasoning? The first issue will be taken up in the next section, and the second will be deferred to section 6.3.

5.3 Generating the Constraint-based Rule

It is of benefit that the constraint-based "choice-point" rule above be generated automatically, given that we know we have a complete set of possible participants. In this section we show how this generation could be effected, without consideration of how we know the set of possible participants is closed. This consideration will be taken up in section 6.3.

Generation of the "choice-point" rule can occur using similar principles to the generation of the "only_rule" above, and specifically by constructing meta-rules that infer logical inferences with premises and conclusions, based upon matching the meta-rule premises to the CE facts. However this generation process are quite complex, and will only be sketched out here.

The key component is the conceptualisation of a type of logical inference specifically for representing choice-point rules:

```
conceptualise
  a ~ choice point inference ~ X that is a logical inference and
     has the entity concept C1 as ~ potential concept ~  and
     has the entity concept C2 as ~ definite concept ~  and
     has the entity concept C3 as ~ impossible concept ~  and
     has the value V as ~ chosen name ~.
```

which has three properties for holding the entity concepts that are needed to build the rule and which characterise the entities involved. In our example the "potential concept" is 'possible participant', the "definite concept" is 'participant' and the "impossible concept" is 'non-participant'. It can be seen from the choice-point rule that such concepts are necessary in the construction of the rule, and it is the way that entities are characterised by these concepts that underlie the choice point reasoning. In effect the template for a choice-point rule is:

```
if
  ( there is a IMPOSSIBLE CONCEPT named XXXX ) and
  ( there is a IMPOSSIBLE CONCEPT named YYYY ) and
   … and
  ( there is a POTENTIAL CONCEPT named BBBB )
then
  ( there is a DEFINITE CONCEPT named BBBB ).
```

and it is the job of the meta-rules to construct this template given the entity concept types. This is achieved in two stages:

- [cp_generaterule_1]: For each entity E that is of type POTENTIAL CONCEPT (i.e. is a possible participant) create a new "choice point inference" with the common name CN of E as its "chosen name" (thus specifying the rule to be about this entity) and with the relevant potential, definite and impossible concepts (based upon the user-defined "is modal to" and "is opposite" relations to the POTENTIAL CONCEPT, i.e. 'possible participant', 'participant' and 'non-participant'). This stage also plants the statements "there is a POTENTIAL CONCEPT named CN" as premise and "there is a DEFINITE CONCEPT named CN" as conclusion.
- [cp_generaterule_2]: For each **choice point inference** and for each entity E1 with common name CN1 that is of type POTENTIAL CONCEPT but does not have the same as the chosen name CN of the choice point inference, plant the statement "the thing CN1 realises the entity concept POTENTIAL CONCEPT and realises the entity concept IMPOSSIBLE CONCEPT" as premise.

In this way all of the choice point rules are generated, although a modification to this process is required in order to address the issue of when the complete set of possible participants are known. This is described in a later section.

A similar form of processing is used to generate the choice point inconsistency rule that notes an inconsistency when there are no possible participants (i.e. they have all been ruled out).

5.4 Solving the ELICIT WHO identification task

The result of all of the processing described above is that we have now inferred two participants, the Lion and the Violetgroup. A further rule in the domain takes this information and infers participation in the elicit attack:

the attack situation Elicitattack involves the operative Lion and involves the group Violetgroup.

This CE sentence constitutes the solution to the ELICIT identification task for the WHO component.

5.5 Hand Formulation of the complete ELICIT task

The description above shows the formulation of the ELICIT task to perform the WHO component, based upon the NL processing of simplified sentences. Work has also been done to formulate the entire identification problem, including WHAT, WHERE and WHEN. However this has not been done from the NL analysis of sentences, but, rather by the hand analysis of the sentences and the hand generation of the CE facts (and rules) [ELICITHANDFORMULATION]. The construction of this formulation has required the representation of the ambiguities in the NL sentences noted in [6], whereas this need was less evident in the formulation of the WHO, as there were no significant ambiguities.

The running of the hand formulated rules did lead to a conclusion for all of the required dimensions, but analysis of the rationale graph showed that this was significantly dependent upon one particular assumption, that an unprotected target was preferred, and therefore protected targets could be ruled out for the WHAT. Once this assumption was removed many possible options for the WHAT were evident, leading to a loss of confidence that the conclusion for the WHAT is actually correct.

The complete hand formulation exercise is therefore useful in raising another example of the need to handle uncertainty, and it is for this reason that it is mentioned in this paper. Further discussion of the ambiguity is deferred to a later section, after the mechanisms for handling ambiguities are introduced.

6 Assumptions for Ambiguity and Uncertainty

The processing described above used CE to express the logic of the facts, and rules, as propositions which define entities, properties and their relations, and it has been presumed that these propositions are either definitely true or not true (the latter being the case when a fact is expressed using the syntax "it is false that P"). However there are some forms of reasoning that require the expression of truth values other than definitely true or definitely false, when there is some uncertainty about the facts or ambiguities in the interpretation of information. Two examples have already arisen from the descriptions above, and now need to be addressed:

- the uncertainty as to whether the set of possible participants is complete
- the ambiguity of some sentence interpretations when formulating the complete solution to the WHAT component of the ELICIT task

The syntax and semantics of CE has been extended to allow expression of, and reasoning about, such uncertainties and ambiguities. The extensions involve two aspects, the use of **assumptions** and the use of numerical or verbal expression indicating a **degree of uncertainty**; these can also be combined.

An assumption represents the act of hypothesising a proposition (called the **assumed proposition**) that has no definite logical basis but that might be true, for the purpose of seeing what would follow if it were true. Once the assumed proposition is added to the fact base, it is treated as any other basic proposition and is subject to the application of rules leading to inferred facts that are dependent upon the assumption. If the reasoning leads to an inconsistency (i.e. the inference of two propositions that are contrary to each other) then an

analysis is performed to determine the assumptions that led to the inconsistency. It is possible for the user or the system at any time to retract (or defease) an assumption, leading to the loss of logical support for the assumed proposition and all propositions (CE sentences) that depended upon the assumption. If the result is that a proposition no longer has a logical support from given (or other assumed) facts via rules, then that proposition is also in effect defeased, and is not longer considered to be true.

This capability supports different reasoning strategies such as: reasoning by multiple interpretations, assuming each interpretation in turn and ruling out those that lead to an inconsistency; maintaining multiple hypotheses about the world in parallel; "reductio ad absurdum" where a proposition is proved true by assuming its opposite and showing that it leads to an inconsistency. Much work has been done in the ITA on the use of assumption-based reasoning, using ideas originated in [18], in the area of handling ambiguities in NL sentences [19] and uncertainties in NL sentences [20] and further work is being proposed in the use of argumentation based upon assumptions for more advanced reasoning.

6.1 Assumptions and numerical certainty values

A basic assumption may be expressed in CE as:

it is assumed by the thing A that AP

where AP is a basic CE sentence and is the assumed proposition, and A is the entity that is making the assumption. Thus the sentence:

it is assumed by the agent A that the group Coralgroup operates in the time interval nighttime.

would state that this assertion of the operating time interval was not a given truth but something that one might wish to assert in order to determine its consequences (in fact in the processing above this assertion was a given truth; the sentence given here is for illustrative purposes only).

A numerical value can be associated with the assumed proposition by a CE sentence such as:

it is certain to degree '0.7' that the group Coralgroup operates in the time interval nighttime.

An account of the handling of uncertainty values is not required here. It suffices to note that numerical values are propagated through the graph of reasoning steps, based upon a calculus defined by the user (handling and, or and not operations on the numerical values); that there are separate pathways for positive support and negative support for a proposition, allowing three-valued logic; and that inconsistencies are detected leading to the distribution of numerical values for the positive and negative support. See [19] for further details.

6.2 Types of uncertainty and types of assumption

In order to use assumptions to handle uncertainties and ambiguities in NL, we have found it useful to represent different types of uncertainty/ambiguity via different "types of assumption". In the current formulation [13] the different types of assumption are actually represented by different types of assumed proposition, ie the P in "it is assumed by the thing T

[13] This account is based upon the representation used in the completed work on the ELICIT task where it is the assumed proposition that holds the type of uncertainty; more recent work (eg [NLUNCERT]) provides a more elegant way to represent types of ambiguity, where it is the assumption itself that holds the type, but this approach is not used in this paper.

that P". For example, the two types of uncertainty/ambiguity noted in this paper may be expressed by different propositions.

The following sentence states a "world closure" type of assumed proposition:

there is a world closure named wc1 that closes the entity concept 'possible participant'.

which expresses the "fact" that our knowledge of the set of "possible participants" is complete (in effect we are closing the world in respect of this concept).

The following sentence states a "domain interpretation" type of assumed proposition:

there is a domain interpretation named prot_targ_non_target that
has "the preference for non-protected targets is treated as an absolute truth" as description.

which expresses the "fact" that a particular type of interpretation can be made on information provided by an ELICIT sentence about the domain.

To make these into assumptions, rather than given facts, it is necessary to write them as CE assumptions. For example the world closure is stated:

it is assumed by the agent dm that there is a world closure named wc1 that closes the entity concept 'possible participant'.

The example of the domain interpretation is slightly different, as it is to be stated within a rule. This will be covered in section 6.4.

6.3 Closing the world

At some point it is necessary for the user to decide that all of the possible participants are known. The exact event that causes the user to make that decision is not considered here. However once the decision is made then the following assumption should be asserted:

it is assumed by the agent dm that there is a world closure named wc1 that closes the entity concept 'possible participant'.

As noted above, in the set of ELICIT sentences there are actually two sentences that could define the set: "The Purple or Gold group may be involved" and The Azure, Brown, Coral, Violet, or Chartreuse groups may be planning an attack". In the processing described in section 5.3, we assumed that both of these sentences were to be used, and that in effect we should make the closed world assumption after receipt of all sentences. However an alternative approach (and one that is consistent with the nature of the ELICIT experimental framework itself) would be to make the assumption on receipt of (say) the first sentence, and by doing so, this leads to an interesting result, as will be described below.

In the account of the meta-rule to construct the "choice-point" rule, it was crucial that the closed world assumption had been made. Here we make that explicit, and under the control of the user by adding the assumed proposition to the premise of the meta-rule, thus adding the premise:

…
(there is a world closure named A that closes the entity concept POTENTIAL)
…

As well as providing a means whereby the POTENTIAL CONCEPT ('possible participant') can be passed to the rule, it also allows the user to control when the rule should be applied

(and hence the set of possible participants to be used), since the rule will only fire when the assumption is made. This mechanism was used to generate the full rule as used in the example above. A further subtlety was added, in that the meta-rule included the closed world assumption itself as a premise to the choice-point rule, generating:

```
[ wc1_cp_Violetgroup ]
if
  ( there is a non-participant named 'Azuregroup' ) and
  ( there is a non-participant named 'Browngroup' ) and
  ( there is a non-participant named 'Chartreusegroup' ) and
  ( there is a non-participant named 'Coralgroup' ) and
  ( there is a non-participant named 'Goldgroup' ) and
  ( there is a non-participant named 'Purplegroup' ) and
  ( there is a possible participant named 'Violetgroup' ) and
  ( the world closure wc1 closes the entity concept 'possible participant' )
then
  ( there is a participant named 'Violetgroup' ).
```

The effect seems to be the same, given that the world has been closed both at the point of rule creation and at the point of rule execution. However having the assumption in the choice-point rules is advantageous as the user can see (via the rationale when the choice-point rule is applied) that the conclusion of the participant really does depend upon the closed world, and hence the user has the option to undo the closure if they consider it is incorrect.

However, suppose that the user decides to apply the world closure on receipt of the first sentence only ("The Purple or Gold group may be involved"). Then the meta-rule will generate a **choice-point inconsistency detection** rule:

```
[ wc1_cp_incon_rule ]
if
  ( there is a non-participant named 'Goldgroup' ) and
  ( there is a non-participant named 'Purplegroup' ) and
  ( the world closure wc1 closes the entity concept 'possible participant' )
then
  ( the choice point inconsistency cp_incon has the entity concept 'possible participant' as
potential concept ).
```

which under the ELICIT example will actually fire and generate an inconsistency (since the table of ruled-out participants includes all of these!). The proof table for the inconsistency is shown below:

wc1 is a world closure	
wc1 closes possible participant	ass310
Purplegroup is a non-participant	
Goldgroup is a non-participant	
cp_incon is a choice point inconsistency	wc1_cp_incon_rule

and shows it was dependent upon an assumption (ass310) which supported the assumed proposition (wc1 closes possible participant). Thus the user can see that the world closure was made too soon, and that both ELICIT sentences are needed to define the set of possible participants.

6.4 Sentence and domain interpretations

Another major source of assumptions[14] in the ELICIT task was the interpretation of the sentences that provides the basic information, as was described above, see [6]. These were modelled as "sentence interpretations" or "domain interpretations[15]". In the example above of the preference for non-protected targets, a domain interpretation was used, as it seems to refer to a general property of the groups involved rather than a specific sentence. However in this case, it is thought that the assumption is relevant to the processing of a specific rule rather than a general statement, as was the case for the closed world assumption. Some assumptions seem to be best made in the context of some other conditions, and in these cases we wish to place the assumption within the premise of a rule rather than as a general context-free statement. In order to achieve this, we have modelled a further relationship between (in effect) an assumption and a rule, stating that the assumption "can be made by" the rule:

```
it is assumed by the agent dm that the domain interpretation prot_targ_non_target can be
made by the rule rule_prot_targ_non_target.
```

This relationship is intended to be used in the general rule form of:

```
[RULENAME]
if
  PRECONDITIONS and
  the ASSUMPTIONTYPE ID can be made by the rule RULENAME
then
  CONCLUSIONS
```

which is intended to mean that the CONCLUSIONS do not entirely logically follow from the PREMISES but require an additional assumption (or leap of faith) that certain additional conditions hold that are not made explicit, but are to be accepted. For example in the rule:

```
[ rule_prot_targ_non_target ]
if
  ( the potential target T is a protected thing ) and
  ( the domain interpretation prot_targ_non_target
    can be made by the rule rule_prot_targ_non_target )
then
  ( the potential target T is a non-target ).
```

states that if a potential target T is protected and we accept that the interpretation of protected targets are not the focus of the attack, then it follows that T is not a target. It is still necessary to construction the assumption itself, as in:

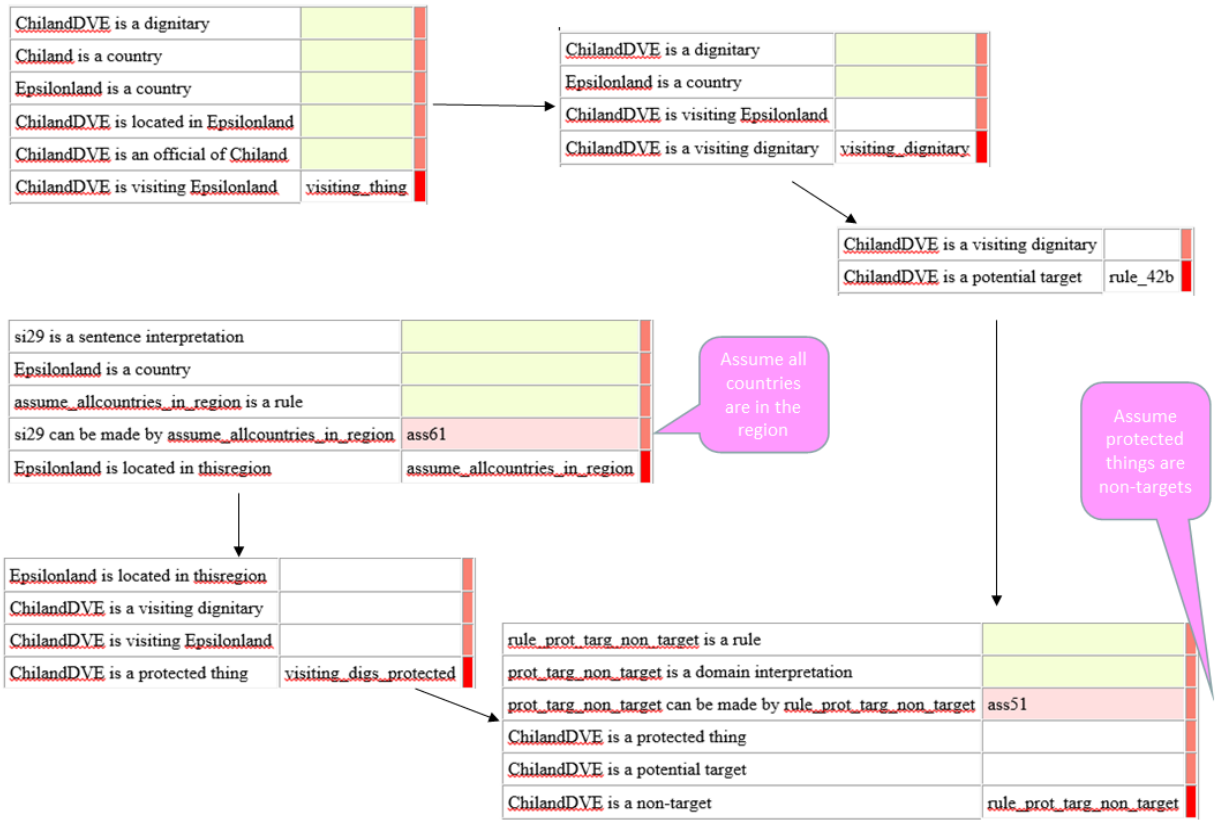
```
it is assumed by the agent dm that the domain interpretation prot_targ_non_target
can be made by the rule rule_prot_targ_non_target.
```

The result is that all protected targets are categorised by the rule as non-targets, but that this conclusion is dependent upon the assumption supporting the domain interpretation prot_targ_non_targ. An example of this reasoning is shown below, in the inference that "ChilandDVE" (a visiting dignitary from Chiland visiting Epsilonland) is a non-target, since there is another rule (visiting_digs_protected) indicating that all visiting dignitaries in the region are protected. (There is a further sentence interpretation asserting that Epsilonland is

[14] It should be mentioned that the ELICIT researchers did not seem to notice these ambiguities, and at least one researcher does not accept their existence.

[15] The distinction is not entirely stable, and it may be dropped in the future

actually in "the region", not something that was explicitly stated in the ELICIT contextual information). The diagram shows that the conclusion (ChilandDVE is a non-target) is dependent upon two assumptions (a domain interpretation and a sentence interpretation) shown in pink and by the bubbles).



Given this rationale, it is the case that if either of the assumptions/interpretations were withdrawn, or defeased, then the conclusion that ChilandDVE is a non target would no longer be true. Given that the non-target-ness of ChilandDVE led via a choice-point rule on the concept 'possible target' to an inference of the actual target, this inference would also be withdrawn, with the result that the target (the WHAT) would no longer be known.

In the hand formulation of the ELICIT sentences, all interpretations indicating a source of ambiguity or uncertainty were encoded in the form of assumptions, allowing these sources to be visible to the user in the rationale for the conclusions.

7 Conclusions

This paper has sought to show how ITA Controlled English, a form of natural English with a reduced syntax but based upon a conceptual model, can be used to guide NL processing, in the mapping of deep but linguistic semantics to domain semantics, and then to infer information in a fairly complex problem-solving task. CE can also model uncertainty and ambiguities, in the form of assumptions, allowing the capturing of sources of uncertainty and ambiguity inherent in the original sentences. The results of the reasoning can be made visible to the user, including the sources of uncertainty.

The processing of sentences and reasoning has been applied to a realistic task that is used elsewhere to experiment with collaborative problem-solving, and it is hoped that the formulation of the problem may be useful to these experiments. It is the aim to build a NL-processing agent that can be involved in other ITA research on collaborative sensemaking based upon the ELICIT task.

This work is only a start on this direction. We aim to extend the range of sentences handled by our system, seeking to apply the techniques to the MRS test suite to ensure coverage of the main linguistic phenomena to be found in the output of the ERG system. A significant issue is the need for considerable domain-specific and common sense knowledge for the disambiguation of the ELICIT sentences (which are not intended to be especially complex). Currently we have to a large extent avoided this problem by the human simplification of sentences, and it is necessary to start to address this. Our hope is that by using information from a domain model, we may determine how domain-knowledge can more deeply influence the processing and disambiguation of these types of sentence.

The capabilities we are researching have the potential to provide support for the users and analysts in the types of cognitive problem solving task described in the introduction.

This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

8 Appendix: CE extracted from Simplified Sentences

The following tables show the original ELICIT sentences, the simplified sentences and the CE facts and rules extracted from these sentences:

Original ELICIT	**Simplified**	**CE facts extracted**
The Lion is involved	The Lion is a participant	there is a participant named Lion.
The Lion attacks in daylight	The Lion operates in the daytime.	the operative 'Lion' operates in the time interval daytime.
The Azure, Brown, Coral, Violet, or Chartreuse groups may be planning an attack.	The Azuregroup may be a participant. The Browngroup may be a participant. …	there is a possible participant named Azuregroup. there is a possible participant named Browngroup …
The Chartreuse group is not involved	The Chartreusegroup is not a participant.	there is a non-participant named Chartreusegroup.
The Purple or Gold group may be involved	The Purplegroup may be a participant. The Goldgroup may be a participant.	there is a possible participant named Purplegroup. …
All of the members of the Azure group are now in custody	The Azuregroup is not operational	there is a non-operational agent named Azuregroup.
Reports from the Coral group indicate a reorganisation	The Coralgroup is not operational	there is a non-operational agent named Coralgroup.
The Brown group is recruiting locals - intentions unknown	The Browngroup is recruiting locals.	the group 'Browngroup' recruits the local agent 'a local agent'.
The Lion will not risk working with locals	The Lion does not work with locals.	the operative 'Lion' cannot work with the local agent 'a local agent'.
The Azure and Brown groups prefer to attack at night	The Azuregroup operates in the nighttime. The Browngroup operates in the nighttime.	the group 'Azuregroup' operates in the time interval nighttime. the group ‘Browngroup' operates in the time interval nighttime.
The Violet group prefers to operate in daylight	The Violetgroup operates in the daytime.	the group ‘Violetgroup' operates in the time interval daytime.

The Coral group prefers to attack at night	The Coralgroup operates in the nighttime.	the group ‘Coralgroup' operates in the time interval nighttime.
The Purple group prefers to attack in daylight	The Purplegroup operates in the daytime.	the group Purplegroup' operates in the time interval daytime.
The Brown group needs time to regroup	The Browngroup is not operational	there is a non-operational agent named Browngroup.

Original ELICIT	**Simplified**	**CE rules extracted**
The Lion is known to work only with the Azure, Brown, or Violet groups	The Lion only works with the Azuregroup and Browngroup and Violetgroup.	[only_e3_26] if (there is a agent named 'Lion') and (the agent A is different to the agent Azuregroup) and (the agent A is different to the agent Browngroup) and (the agent A is different to the agent Violetgroup) then (the agent Lion cannot work with the agent A).

9 References

[1] International Technology Alliance, https://www.usukita.org/
[2] http://www.dodccrp.org/html4/elicit.html
[3] Mott, D. Summary of CE, https://www.usukita.org/papers/5658/details.html (2010)
[4] Flickinger, D., The English Resource Grammar, LOGON technical report #2007-7, www.emmtee.net/reports/7.pdf
[5] Copestake, Ann., Flickinger, D., Sag, I. A., and Pollard, C., Minimal Recursion Semantics: an introduction. Research on Language and Computation, 3(2-3):281–332. (2005)
[6] Mott D., On Interpreting ELICIT sentences, https://www.usukitacs.com/node/2603 (2014)
[7] Mott, D.,Conceptualising ELICIT sentences, https://www.usukitacs.com/node/2604. (2014)
[8] James F. Allen: Maintaining knowledge about temporal intervals. In: Communications of the ACM. 26 November 1983. ACM Press. pp. 832–843, ISSN 0001-0782
[9] http://www.delph-in.net/
[10] The PET parser, http://moin.delph-in.net/PetTop
[11] http://svn.emmtee.net/trunk/uio/wesearch/esd.txt
[12] Mott, D., Braines, D., Poteet, S., Kao, A., Controlled Natural Language to facilitate information extraction using Controlled English, ACITA 2012.
[13] Levin, B., & Rappaport Hovav, M. Argument Realisation, Cambridge University Press, Jun 20, 2005
[14] http://verbs.colorado.edu/~mpalmer/projects/verbnet.html
[15] Mott, D., Moving from ERG linguistic semantics to CE domain semantics, May 2015, https://www.usukitacs.com/node/2568
[16] Mott, D. ELICIT domain model for CE-based analysis, May 2014, https://www.usukitacs.com/node/2698
[17] Mott, D., ELICIT Simple Formulation of Extracted facts for CE-based analysis, May 2014, https://www.usukitacs.com/node/2699
[18] de Kleer, J. An assumption-based TMS. Artificial Intelligence, 28:127-162. (1986)
[19] Mott, D., CE-based mechanisms for handling ambiguity in Natural Language, https://www.usukitacs.com/node/2612. (2014)
[20] Xue, P., Poteet, S., Kao, A., & Mott, D, Uncertainty Expressions in Natural Language and their CE Representation, submitted to MILCOM14, https://www.usukitacs.com/node/2677.
[21] Bender, E.M., personal communication, August 2013.
[22] Mott, D., Giammanco, C., Braines, D., Dorneich, M., and Patel, D., (2010). Hybrid Rationale and Controlled Natural Language for Shared Understanding. In Proceedings of the Fourth Annual Conference of the International Technology Alliance, London, UK, September 2010.