Chapter 20

Deductive Question Answering from Multiple Resources

Richard Waldinger
Douglas E. Appelt
John Fry
David J. Israel
Peter Jarvis
David Martin
Susanne Riehemann
Mark E. Stickel
Mabry Tyson
Artificial Intelligence Center
SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025
`<lastname>@ai.sri.com`

Jerry Hobbs
Intelligent Systems Division,
USC Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292
`hobbs@isi.edu`

Jennifer L. Dungan
Ecosystem Science and Technology Branch
NASA Ames Research Center
Moffett Field CA 94035
`Jennifer.L.Dungan@nasa.gov`

Abstract

Questions in natural language are answered by consulting multiple sources and inferring answers from information they provide. An automated deduction system, equipped with an axiomatic application-domain theory, serves as the coordinator for the process. Sources include data bases, Web pages, programs, and unstructured text. Answers may contain text or visualizations. Although the approach is domain-independent, many of our experiments have dealt with geographic questions.

1. Introduction

Much question-answering research is restricted to finding a single information resource that contains the answer to a question, in a form that is a simple rephrasing of the question itself. The present work explores the possibility of constructing answers that are not known by any one source, but that rather must be inferred from information supplied by many sources. We are using an automated deduction system, with an axiomatic application-domain theory, as the central integrator.

Our research prototype, called QUARK (Question Answering through Reasoning and Knowledge–its geographical version is called Geo-Logica, Waldinger et al., 2003), has the following structure: The question is parsed and translated into a logical form by Gemini, a broad-coverage English parser. The logical form is phrased as a conjecture and submitted to the automated deduction system SNARK, a general-purpose theorem prover with special capabilities for temporal and spatial reasoning. SNARK is equipped with an application-domain theory that provides knowledge as to how the question can be decomposed. The capabilities of each knowledge resource are specified by axioms in this theory; these axioms serve to advertise the corresponding resource, so it can be invoked as appropriate while the proof search is underway. Each resource is linked to the theory by a procedural-attachment mechanism, which allows the theorem prover to behave as if knowledge possessed by the resource were represented axiomatically in the theory.

Although the approach is domain-independent, we have experimented most with geographical question answering. Among the resources we have linked are an information-extraction engine, TextPro; the Alexandria Digital Library Gazetteer, a geographical dictionary with 6 million entries; the CIA World Factbook; the ASCS Semantic Web search engine; and a variety of NASA data sources. Answers can contain text, maps, or visualizations of satellite image data.

For example, the question "Could Mohammed Atta have met with an Iraqi official?" was answered positively by consulting TextPro (to determine, from online text sources, that Atta went to Ruzyne, Czech Republic on a certain day in 2000 and that the Iraqi agent al-Ani was with the Iraqi embassy to the Czech Republic in 2000); date and temporal reasoning to verify that the two times do overlap; an axiom from the application-domain theory to determine that embassies are in capitals; the CIA World Factbook (via the Semantic Web search engine ASCS) to establish that the capital of the Czech Republic is Prague; the ADL Gazetteer to find the latitudes and longitudes of Prague and Ruzyne; and a geographical computation Web page from Northern Arizona University to determine that the distance between those two latitude-longitude (lat/long) pairs is less than seven miles, close enough to permit a visit. Further questioning causes the display of maps and satellite images of Ruzyne, illustrating its proximity to Prague. (In fact, the reality of this meeting is hotly disputed, but the point is that we can reproduce the reasoning that uncovers the possibility.)

In this paper, we introduce the central notion behind QUARK, the extraction of answers from mathematical proofs. We describe the individual components of

QUARK and illustrate its behavior on some typical questions. We then discuss possibilities for the future development of QUARK.

2. Extracting Answers from Proofs

Questions are presented to QUARK in English, but they are translated by the parser Gemini into a logical form, which we can write

`Q[?z], answer ?z.`

(Our convention is that variables are prefixed by question marks.) This is viewed as a conjecture that an entity `?z` exists for which `Q[?z]` is true; this is the conjecture that is submitted to the theorem prover SNARK. The understanding is that we would like to find an expression `?z` for which `Q[?z]`follows from the axioms of the application-domain theory; if we succeed, `?z` is our answer.

For instance, a question "Who is the mother of John?" would be translated into a logical query

`mother(?z, John) , answer: ?z`.

(The actual translation is more complicated but it simplifies to this form.) This is taken to mean that SNARK is to find an expression `?z`, for which

`mother(?z, John)`

is true. That expression will stand for John's mother.

Assume further that our application-domain theory contains the axiom

`mother(Sue, John)`,

meaning that Sue is the mother of John.
During the theorem proving process, this axiom will be unified with the query

`mother(?z, John)`,

causing the variable `?z` to be instantiated to `Sue`. This completes the proof of the existence of a mother of John, and tells us that the answer to our query is Sue.

This is a very simple example. Typically, the proof of the conjecture will take many steps, the axioms will have a more complex logical structure, and the answer will be a more complex object, whose components are only discovered gradually. But the principle is the same; we prove the existence of an entity that satisfies the conditions of our query, and we extract an answer, an expression that stands for such an entity, from the proof.

Although mothers are unique, it is common that a query will have many different possible answers. For instance, if we were looking for a child of Sue, we might find other answers besides John. SNARK will find one answer at a time, but can be reinvoked to find other proofs, some of which will yield different answers.

2.1. The Application-Domain Theory

The application-domain theory provides the meanings for the symbols that appear in the query and in the answer, describes the relationships between the various concepts, advertises the capabilities of the information sources, and serves as a repository of knowledge in its own right. For example, in the example concerning Mohammed Atta mentioned in the introduction, the text specified that the agent al-Ani was located at the Iraqi embassy to the Czech Republic, but did not say where that embassy was. We relied on a piece of background knowledge that states that the embassy to a country is in the capital of that country. This was represented by the axiom

```
in(embassy(?country1, ?country2),
   capital(?country2)).
```

Here `embassy(?country1, ?country2)` is the embassy of `?country1` to `?country2.` Our convention is that `?country1` and `?country2` are variables of sort `country;` during a proof they can only be instantiated to entities that are countries. The sort mechanism of SNARK is not merely a convenient abbreviation: it can reduce the size of the search space of a proof by restricting the sorts of entities to which a variable may be instantiated.

2.2. Representation of Places

Part of the effort of designing an application-domain theory is to find good ways of representing the important entities of the theory. In developing a geospatial theory for geographical reasoning, we were faced with the need to find a systematic way of naming places. While we could always represent a city name, such as Springfield, Illinois, United States by a long constant name

```
Springfield-Illinois-United-States,
```

the theorem prover has no mechanism for looking inside the names of constants. SNARK would behave no differently if the city were called `F17.` Instead, we use a systematic naming convention, in which countries are represented by their own names (e.g., `United-States`), subdivisions of countries are represented by terms such as

```
feature(state, Illinois, United-States),
```

and subdivisions of subdivisions are represented by even more deeply nested terms, such as

```
feature(city, Springfield,
          feature(state, Illinois,
                 United States)).
```

In general, we use expressions of form

```
feature(<type>, <name-indicator>, <region>)
```

to stand for the subregion of `<region>` whose geographical classification is `<type>` and whose internal name is `<name-indicator>`. These names are far more informative than long constant symbols. Axioms in the theory can tell us that the term above is of geographical classification `<type>`, is called `<name-indicator>`, and is a subregion of <region>. If we were being rigorous, we would represent countries by terms such as

```
feature(country, United-States, Earth),
```

but that has not yet been necessary.

The same kind of naming scheme may be useful for areas other than geography that have a hierarchical naming structure. For instance, in talking about religions, it is necessary to have a way of speaking about sects and denominations. Rather than represent the Sunni sect of Muslims as a constant string `Sunni-Muslim`, it is more informative to speak of

```
group(sect, Sunni, Muslim).
```

The name conveys the information that the Sunni sect is a denomination of the Muslim religion.

2.3. Conventional versus internal names

The internal naming scheme we use for representing places and other entities is not meant for use by the users of QUARK, nor is it accepted by any of the external sources we consult for information. The axiomatic theory has allows us to couple conventional names, which are strings, with the internal terms. For instance, the city of Springfield, Illinois, is represented conventionally by the string

`“Springfield, Illinois, United States”`.

The geospatial theory defines a relation `name` that relates the conventional string with its internal name. Thus,

```
name(“Springfield, Illinois, United States”,
      feature(city, Springfield,
              feature(state, Illinois,
                      United States)))
```

is true in the geospatial theory. Given either name, the theory contains enough information to construct the other.

3. Procedural Attachment

While some of the knowledge of QUARK resides in its application domain theory, the theory functions largely as a catalogue, index, and coordinator of external knowledge resources, sometimes called agents. Selected symbols of the theory are linked to these resources by the *procedural attachment* mechanism. When one of these symbols take a part in the proof search, the corresponding agent is invoked; the proof search resumes when the agent returns an answer.

3.1. The mechanism

Procedural attachment was originally introduced to allow the more efficient representation of procedures in an axiomatic theory. For example, in a theory that involves numerical computation, it is possible to represent a function such as addition by a set of axioms that define it, but computation of the value of a term such as `plus(2, 2)` would then be cumbersome. If we procedurally attach the symbol `plus` to an arithmetic procedure, the procedure would invoke the addition function and return the value `4`. Terms with variables, such as `plus(?x, 2)`, would not be evaluated by the procedural attachment.

While we also use procedural attachment for efficiency, we get other benefits by linking to external knowledge sources, where it would be impossible to represent the same knowledge axiomatically. For example, the Alexandria Digital Library

Gazetteer not only contains information about 6 million place names but also is continually being extended and changed. Other symbols may be linked to sensors or other sources that are in constant flux.

The Gazetteer is linked to several symbols in our geospatial theory, including the symbol

```
adl-region-bounding-box(
   ?region-name, ?type-name,
   ?lat1, ?lat2, ?long1, ?long2)
```

Here, if `?region-name` is the name of a place whose geographical category is named `?type-name`, the four numbers `?lat1, ?lat2, ?long1, ?long2` stand for the bounding box of the place in question, in other words, the north and south latitudes and the east and west longitudes that enclose it.

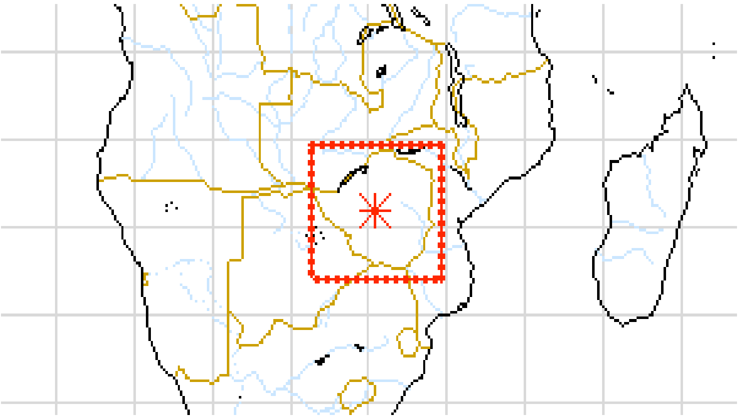

Figure 1. Bounding Box of Zimbabwe

The symbol is linked to the Gazetteer in such a way that the first two argument variables, ?region1 and ?type, are regarded as inputs, and the rest are regarded as outputs. This means that if the region and its type are instantiated, the Gazetteer will be invoked to find values for the latitudes and longitudes of the bounding box. For example, if, during the course of a proof, the expression

```
adl-region-bounding-box(
   “Zimbabwe”, “country”
   ?lat1, ?lat2, ?long1, ?long2)
```

appears, the Gazetteer will be invoked and the four output variables will be instantiated:

```
adl-region-bounding-box(
   “Zimbabwe”, “country”,
   “-15.22”,“-22.93”,“33.65”,“25.11”).
```

Henceforth, SNARK will behave as if the above formula were included in the theory as an axiom. This means that in proving a theorem, SNARK will act as if the six million lat/long pairs and bounding boxes provided by the Gazetteer were all included in the geospatial theory.

3.2. Axioms that Advertise the Capabilities of Agents

One of the chief virtues of using an axiomatic theory to coordinate multiple agents is that we can exploit axioms to advertise the capabilities of an agent. This means that, when we introduce a new agent into the system, we provide one or more axioms that expresses what that agent can do. This ensures that the agent is invoked when appropriate, even if (as is usually the case) none of the other agents know about the new one. Whenever a new subquery is formulated, those agents that are capable of answering it step forward, invoked not by name but as a by-product of the theorem-proving process.

For example, let us look at the axiom that advertises the ability of the Alexandria Digital Library Gazetteer to find the bounding box of a country:

```
region-to-bounding-box(?country) =
bounding-box(?lat1, ?lat2, ?long2, ?long2)
   <=
name(?region-name, ?country) &
adl-region-bounding-box(
   ?region-name, “country”,
   ?lat1, ?lat2, ?long1, ?long2).
```

In other words, to find the bounding box of a country, first find a name-string for the country, `?region-name`; then call the ADL Gazetteer to find the corresponding four numbers that constitute the bounding box.

Suppose a subquery is formulated to find the bounding box of a country, e.g.

```
region-to-bounding-box(Zimbabwe) = ?bounding-box.
```

This query will be unified with the consequent of the axioms (the part above the left-arrow, <=). Unification is a two-way pattern-matching process; variables in both expressions are instantiated. The variable `?country` is instantiated to `Zimbabwe`; the variable `?bounding-box` is instantiated to

```
bounding-box(?lat1, ?lat2, ?long2, ?long2).
```

SNARK will invoke other axioms (and a procedural attachment) to find that `?region-name` can be instantiated to the string `"Zimbabwe"`. Finally, the ADL Gazetteer is invoked via procedural attachment, as in the preceding section. As a result, the four output variables `?lat1, ?lat2, ?long2, ?long2` are instantiated to `"-15.22","-22.93","33.65","25.11"`. These are the four components of the desired bounding box.

Finding bounding boxes of countries is simple because there are no two countries with the same name. Finding bounding boxes of subregions of countries is more difficult, because different subregions of different countries may have the same name, and we must be careful not to find the wrong one. Another, more complicated axiom advertises the capabilities of the ADL Gazetteer to find states, cities, lakes, and other subregions.

One problem that we face in creating interoperability for multiple disparate agents is that they do not adhere to uniform conventions in notation. For example, for agents that deal with latitudes and longitudes, some adopt a decimal notation, while others employ the older notation of degrees, minutes, and seconds. Of those that use decimal notation, some will write `"32S"` while others prefer `"-32"` for the latitude 32 degrees south of the Equator. Some of the least glamorous but most important agents are those that convert from one notation to another. These are also invoked by procedural attachment, and are summoned by providing axioms that advertise their capabilities.

4. Components of QUARK

In this section we will describe the components of QUARK, in roughly the order in which they come into play in answering a question.

4.1. Gemini

Questions to QUARK are translated into a logical form by Gemini (Dowding et al., 1993), a mature, robust parsing and interpretation system that has been used by several projects at SRI, Stanford, NASA, and elsewhere over the past ten years. Although currently, in QUARK, Gemini is only used to parse questions, in the future it may be used to parse information supplied by the user in dialogue and text from other source material. Gemini may also be used to generate text to present and explain answers to questions.

A broad-coverage English grammar and lexicon for QUARK was compiled from several earlier projects. The open-ended nature of QUARK queries required a much larger vocabulary than previous Gemini projects. More than 50,000 new items were added to the lexicon, including 6000 adjectives and 35,000 nouns from Wordnet, and 400 geographical terms from the Alexandria Digital Library Gazetteer and NASA sources.

Gemini also has a capability for guessing the part of speech of an out-of-vocabulary word and temporarily adding that word to the lexicon. This has proved necessary for dealing with the large number of place names, personal names, and specialized vocabulary that occur in QUARK questions (“Mohammed Atta”, “Jalalabad”, “ATGW-3LR missile”) that cannot be catalogued in advance.

4.2. SNARK

Theorem provers have traditionally been developed to excel at mathematical reasoning, which requires finding non-obvious proofs over theories defined by relatively small sets of axioms. In contrast, SNARK has been developed for applications in artificial intelligence and software engineering, which requires straightforward reasoning on theories defined by large axiom sets.

SNARK (Stickel et al., 2000) is a first-order logic theorem prover with resolution (for general deductive reasoning) and paramodulation (for reasoning about equality), implemented in Common Lisp. It has a sort mechanism, which allows all expressions to be categorized according to a hierarchical sort structure. It is particularly well suited for question-answering applications, for several reasons:

> It has strategic controls that allow us to tailor it to exhibit high performance in selected application domains.
>
> It has a mechanism for extracting answers from proofs.

It has a procedural-attachment mechanism.

It has built-in procedures for reasoning efficiently about space and time.

SNARK is used in NASA's system Amphion (Lowry et al., 1994), for automatic software composition, and in the Kestrel Institute's software development environment, SPECWARE (Kestrel Institute, 2002), as well as several SRI projects.

5. Agents

In this section we describe some of the agents invoked by SNARK's procedural attachment mechanism.

5.1. Alexandria Digital Library Gazetteer

The Alexandria Digital Library Gazetteer (Hill et al., 1999) is a dictionary of about six million place names. For each place, it provides a latitude and longitude or bounding box; a geographical type; a map; a list of super-regions, i.e., regions that include the place; and a list of variant names and spellings. Given a name and a type, it can find places of the appropriate type with that name. It is capable of searching within a bounding box, so it will find only places that are within the box. If the name is not specified, it can find all places of the appropriate type; e.g., it can find all the lakes or airports within a given bounding box. The server is located at the University of California at Santa Barbara.

5.2. CIA World Factbook

The CIA World Factbook (Central Intelligence Agency, 2002) is an almanac of most of the world's countries, including geographic, economic, political, and military information about each. We consult the Factbook for such information as bordering countries, principal subdivisions, capital cities, and religious makeup. The Factbook does not provide bounding boxes or latitudes and longitudes, and it does not have detailed information about cities.

5.3. ASCS Search Engine

The Agent Semantic Communications Service (ASCS; Pease et al., 2002a), developed by Teknowledge, is a search engine that collects and indexes all pages on the Semantic Web that are annotated with markup from the DARPA Agent Markup Language (DAML). Although we have extracted some information from the CIA World Factbook directly, most of the knowledge we get from the Factbook is obtained through ASCS, because DAMLized versions of the Factbook have been produced.

5.4. TextPro Information Extraction Engine

TextPro (Appelt and Martin, 1999) can preprocess selected text documents and extract relational information which is stored in a database. This information can be queried as the proof is in progress. Presently information is being extracted from documents on weapons of mass destruction, terrorism, and related topics provided to the Aquaint program by the Center for Nonproliferation Studies, in Monterey, California.

5.5. Geographical computation agents

QUARK consults a number of agents that perform geographical computations, such as, given two lat/long pairs, find the distance between them or, given a lat/long pair and number, find the lat/long pair that distance north of the given pair. These include a Website at the Northern Arizona University and some internal agents.

5.6. Visualization agents

Quark invokes a number of providers of maps and satellite image terrain visualization systems. NIMA's Geospatial Engine and Generic Mapping Tools supply maps for a given region; the agent can select features to highlight or provide points or vectors to be superimposed on the map. Satellite imagery is provided by USGS's Landsat Project and the NASA Goddard Distributed Active Archive. TerraVision (Reddy et al., 1999) presents a flight-simulator-like three-dimensional view of a selected region; the user can navigate around the selected region using a mouse as a controller.

6. Sample Problems

In this section we examine the behavior of QUARK on some sample problems. These illustrate QUARK's ability to deal with queries involving multiple conditions and to invoke multiple agents in answering them.

6.1. The Petrified Forest

Suppose our query is "Show a petrified forest in Zimbabwe that is north of the capital of Botswana and within 200 miles of Lusaka, Zambia." This is parsed by Gemini, which produces the following logical form:

```
show(?x) &
patient(?x, ?y) &
petrified-forest(?y) &
in(?y, Zimbabwe) &
north(?z, ?y) &
source(?z, ?u) &
capital-of(?u, Botswana) &
within-distance-of(
   ?y, ?v,
   feature(city, Lusaka, Zambia) &
mile-unit(?v) &
count-of(?v, 200)

answer:  ?x
```

This might be translated more literally as "Find a showing ?x of ?y where ?y is a petrified forest and ?z is a northness of ?y and the object of the northness is ?u, the capital of Botswana, and the distance of the petrified forest ?y from the city of Lusaka, Zambia is ?v, where the unit of ?v is miles and the magnitude of ?v is 200."

The geospatial theory has axioms for each of the concepts in this logical form. Rather than reproducing the proof, let us see what agents are invoked to solve the problem.

The ADL Gazetteer finds the bounding box of Zimbabwe and then searches within it for a petrified forest. It finds one, the "Makuku Fossil Forest." The lat/long for the forest is obtained.

The CIA World Factbook reveals that the capital of Botswana is Gabarone. The ADL Gazetteer finds the bounding box for Botwana and then searches within it for the lat/long of Gabarone. A procedural attachment is invoked to compare the latitudes and verify that the Makuku Fossil Forest is indeed north of Gabarone.

The ADL Gazetteer also finds the bounding box for Zambia and searches within it for a lat/long for the city of Lusaka. The Northern Arizona University

Geographical Computation agent is invoked to determine the distance between the Makuku Fossil Forest and Lusaka, 112 miles. This is less than the specified distance of 200 miles.

The lat/long for the fossil forest is then passed to TerraVision, which displays the region around it:

Figure 2. The Makuku Fossil Forest

6.2. The Neighboring Spray Attack

Much of our recent effort has been integrating the TextPro information extraction effort into QUARK. Part of the ARDA Aquaint program has focused on texts supplied by the Center for Nonproliferation Studies in Monterey, California. Some of this material deals with attacks and other episodes using weapons of mass destruction, including chemical and biological weapons. By incorporating TextPro into Quark, we can answer questions that require a collaboration between text and other sources.

For instance, suppose our question is “Display a NIMA map of the location of a chemical attack in a country that borders Afghanistan.” While the text sources will tell where chemical attacks have occurred, they may not mention whether the country bordered Afghanistan. Also, the text sources may not include a map—that may need to be obtained from other sources.

The logical form of the question, produced by Gemini from the text of the query, is (with some simplification)

```
display(?x) &
patient(?x, ?y) &
nima(?y) &
map(?y) &
for(?y, ?z) &
location(?z) &
in(?z, ?u) &
country(?u) &
borders(?u, Afghanistan) &
in(?v, ?z) &
attack(?v) &
chemical(?v)

answer: ?x
```

This is the logical form that is submitted to the theorem prover SNARK. Let us consider what agents SNARK consults in the process of proving the theorem that corresponds to this form. The CIA World Factbook, invoked via ASCS, produces a number of countries that border Afghanistan. For each of these, TextPro consults its Center for Nonproliferation Studies database for attacks that are classified as chemical. For one of the bordering countries, Tajikistan, it finds the following report:

> Six soldiers, three civilians, and one wife of a Russian Embassy worker were killed after drinking champagne laced with cyanide. The locally produced champagne was on sale at a kiosk near a military compound housing members of a Russian led peace keeping force.

The incident is said to have occurred in Dushanbe, Tajikistan. The ADL Gazetteer produces a lat/long pair for Dushanbe, and NIMA's geospatial engine produces two sorts of maps, a political map (displaying national borders) and a DTED (Digital Terrain Elevation Data) map, for the region.

Figure 3. NIMA Political Map for Dushanbe, Tajikistan.

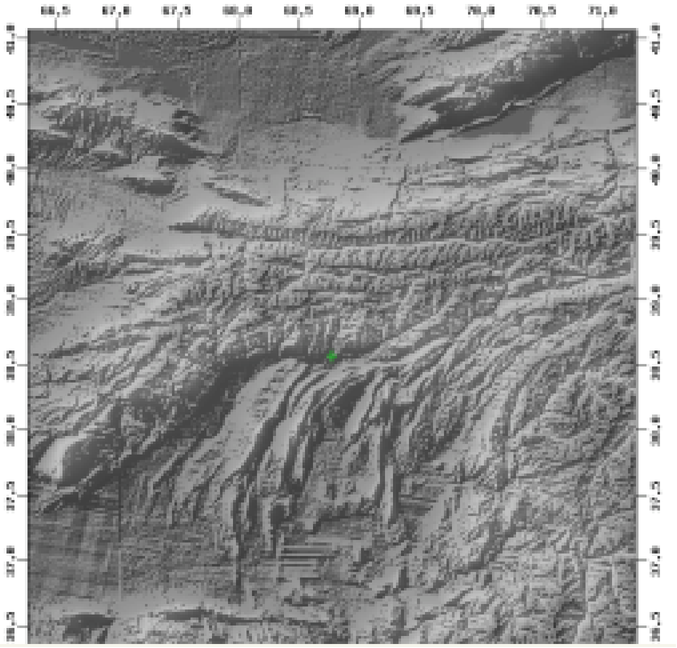

Figure 4: NIMA DTED Map For Dushanbe, Tajikistan.

We can also request a TerraVision view of the same region. If we ask QUARK "When was the chemical attack in Tajikistan?" it will answer "1994."

6.3. The Alleged Secret Meeting

In the introduction, we referred to the problem of discovering a possible meeting between the September 11th hijacker Mohammed Atta and an Iraqi intelligence agent in the introduction. The question QUARK handled was actually "Show the place in which Mohammed Atta met with an Iraqi official." (Although Gemini is capable of dealing with the more nuanced request "Show the place in which Mohammed Atta *could have* met with an Iraqi official," the resulting logical form does not fit within SNARK's first-order logic. Instead, we treat the word "met" as if it meant "could have met," a stop-gap measure.)

The logical form is

```
show(?x) &
patient(?x, ?y) &
place(?y) &
in(?z, ?y) &
meet(?z) &
actor(?z, Mohammed-Atta) &
with(?z, ?u) &
iraqi(?u) &
official(?u) &

answer: ?x.
```

In proving the above theorem, SNARK uses an axiom that states that two people met (i.e. could have met) if they were at places that were sufficiently close together. Originally, the axiom insisted that the people were there at the same time, and used SNARK's temporal reasoning capabilities to ensure that the times the participants were at the meeting place did overlap. But currently TextPro does not have the ability to extract temporal information from text, so we dropped the temporal conditions from the axiom. The resulting axiom is

```
could-have-met(?person1, ?person2)
  <=
in(?person1, ?region1) &
in(?person2, ?region2) &
near(?region1, ?region2).
```

The axiom for "`near`" currently says that two regions are near if they are within 50 miles; a more realistic axiom would take into account the length of time the people were nearby, the method of transportation, and other factors.

In proving the theorem, SNARK uses this axiom and considers places in which Mohammed Atta went. Invoking TextPro (on text we have deliberately planted) it discovers that Mohammed Atta went to the city of Ruzyne, Czech Republic. It consults the ADL Gazetteer to find the bounding box for the Czech Republic and then the latitude and longitude of Ruzyne.

It similarly discovers that Ahmad Khalil Ibrahim Samir al-Ani, an agent of the Mukhabarat, the intelligence service of Iraq, was stationed at the Iraqi embassy to the Czech Republic. Although the text we provided did not state explicitly that al-Ani was an Iraqi official, there is an axiom in the application domain theory that employees of branches of the government of a country are officials of the country. Also the text did not state the location of the embassy, but the theory contains an axiom that embassies are located in the capital of the host country.

The CIA World Factbook (via ASCS) provided the information that the capital of the Czech republic is Prague. The ADL Gazetteer then gave SNARK the latitude and longitude for Prague (it prefers the spelling Praha but recognizes variants), and the website at Northern Arizona University that performs geographical computations reveals that the distance between Prague and Ruzyne is 6.7 miles, close enough for a meeting to have occurred.

The latitude and longitude of Ruzyne are sent to Terravision, which displays a satellite image of the region. The user can then zoom into the region for a closer look. QUARK can of course also provide NIMA or Generic Mapping Tools Maps of the area. If we ask QUARK "How far is it from Ruzyne, Czech Republic to the capital of the Czech Republic?" it will answer "6.7 miles."

Figure 5. TerraVision View of Ruzyne, Czech Republic.

7. Related Work

This work has its roots in early work in deductive question answering and program synthesis, including that of Green (1969) and Manna and Waldinger (1980). The large knowledge base Cyc (Lenat and Guha, 1994) uses theorem proving as a part of the question-answering process. Teknowledge has been building a public upper-level ontology for general world knowledge (Pease et al., 2002b). LCC (Moldovan et al., 2003) has been using a theorem prover to incorporate inference into the information-extraction process. The approach of using procedural attachment from a theorem prover to coordinate multiple agents is relatively new (see Infomaster (Genesereth et al., 1997) and Ariadne (Knoblock and Minton, 1998), for example.) Fonseca et al. (2002) are developing a geographical ontology, with vocabulary but as yet no axioms. Hobbs et al. (2003) are building an axiomatic geographic theory and ontology within the DAML framework. These should all be valuable resources for us.

8. Future Research

Up to now we have been developing QUARK's ability to answer individual questions. In our future research, we will begin to have QUARK engage in a dialogue with its user, in which the user can provide background information as well as questions, and in which new questions can ask for modifications of previous questions or elaboration on the answers. Already QUARK can accept assertions made by its user, which are translated by Gemini into logic and stored temporarily in a local knowledge base, to be invoked in answering future questions. Certain questions will be taken to represent a schema of related questions (e.g. Who is X? may mean to find the role, background, a picture of X, and other personal information).

We will also extend QUARK's ability to explain and justify its answers. Currently it can produce a textual or visual presentation of its answer and a list of facts on which it was based, including quotations of source material. It can also produce the proof from which the answer was extracted, but most users will not be inclined to read it. Gemini, however, has an advanced ability to generate coherent text from logical expressions, which we have not yet exploited in QUARK.

QUARK often has alternative sources for the same information, but it makes no attempt to weigh the relative benefits of consulting one source rather than another. Some sources may be more time-consuming to consult; some may be more reliable. We might include multiple gazetteers so that each could make up for gaps in the others. Some sources may be temporarily unavailable. In the future we will devote some effort to weighing the relative advantages of alternative sources, and perhaps explore choices in parallel.

The theorem prover SNARK has well developed capabilities for reasoning about space and time, which we have not exploited very much yet in QUARK. It includes time and date arithmetic and an implementation of the Allen Temporal Interval Calculus for reasoning about time; we plan to use this for detecting and reasoning about environmental change, itineraries, and other objects that have a temporal as well as a spatial dimension. Eventually we should be able to produce animations as well as static visualizations. SNARK also has an implementation of the RCC8 calculus for reasoning about regions in space; use of this procedure makes it unnecessary to state certain axioms, like the transitivity of the subregion relation, that have an explosive effect on the size of the search space of the proof.

We plan to develop QUARK's abilities to draw conclusions from data (e.g., finding sums and maxima, computing averages, and performing other computations and statistical inferences. It should be able to produce new tables, charts, and similar visualizations of numerical data.

9. Acknowledgements

This work has been supported by the ARDA Aquaint program, the DARPA DAML program, and the NASA Intelligent Systems program.

10. References

Appelt, D. E., Martin, D. L. 1999. Named Entity Recognition in Speech: Approach and Results using the TextPro System.

`http://www.nist.gov/speech/publications/darpa99/html/ie30/ie30.htm`

Central Intelligence Agency.2002. The World Factbook.

`http://www.cia.gov/cia/publications/factbook/`

Dowding, J., Gawron, J. M., Appelt, D., Bear, J., Cherny, L., Moore, R., and Moran, D. 1993. GEMINI: A natural language system for spoken-language understanding. *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, 54–61.

Fonseca, F., Egenhofer, M., Davis, C., and Câmara, G. 2001. Semantic Granularity in Ontology-Driven Geographic Information Systems. *AMAI Annals of Mathematics and Artificial Intelligence – Special Issue on Spatial and Temporal Granularity*: **36**.1–2, 121–51.

Genesereth, M., Keller, A. M., and Duschka, O. M. 1997. Infomaster: an Information Integration System. *SIGMOD Record (ACM Special Interest Group on Management of Data)*. **26**, 539–542.

Green, C. C. 1969. Application of Theorem Proving to Problem Solving. Proceedings of the International Joint Conference on Artificial Intelligence. 219-239.

Hill, L. J., Frew, J., and Zheng, Q. 1999. Geographic Names: The Implementation of a Gazetteer in a Georeferenced Digital Library. *D-Lib*

Hobbs, J. et al. 2003. A DAML Spatial Ontology.

`http://www.daml.org/listarchive/daml-spatial/0002.htm.`

Kestrel Institute. 2003. Specware. `http://www.kestrel.edu/HTML/prototypes/specware.html`

Knoblock, C. A., and Minton, S. 1998. The Ariadne Approach to Web-Based Information Integration. *IEEE Intelligent Systems*. **13,** 5. 17–20.

Lenat, D. B., and Guha, R. V. 1994. Enabling Agents to Work Together. *Communications of the ACM*, **37**, 7 .

Lowry, M., Philpot, A., Pressburger, T., Underwood, I., Waldinger, R., and Stickel, M. Feb. 1994. Amphion: Automatic Programming for the NAIF Toolkit. *NASA Science Information Systems Newsletter*, **31**, 22–25.

Manna, Z., and Waldinger, R. 1980. A deductive approach to program synthesis. *ACM Transactions on Programming Languages and Systems*, **2**, 90-121.

Moldovan, D, Clark, C., Harabagiu, S., and Maiorano, S. 2003. COGEX: A Logic Prover for Question Answering. *Proceedings of the HLT-NAACL,* 87–93.

Pease, A., Li, J., and Barbee, C. 2002a. DAML Agent Semantic Communications Service (ASCS). `http://oak.teknowledge.com:8080/daml/damlquery.jsp`

Pease, A., Niles, I., and Li, J. 2002b. The Suggested Upper Merged Ontology: A Large Ontology for the Semantic Web and its Applications. In *Working Notes of the AAAI-2002 Workshop on Ontologies and the Semantic Web*, Edmonton, Canada, July 28-August 1, 2002.

Reddy, M., LeClerc, Y.-G., Iverson, L., and Bletter, N. 1999. TerraVision II: Visualizing Massive Terrain Databases in VRML. *IEEE Computer Graphics And Applications (Special Issue on VRML)*. **19**, 2. 30–38.

Stickel, M. E., Waldinger, R. J., Chaudhri, V. K. 2000. A Guide to SNARK. SRI International, Menlo Park, CA.

R. Waldinger, P. Jarvis, and J. Dungan. Pointing to Places in a Deductive Geospatial Theory, *Workshop on Analysis of Geographical References; Human Language Technology Conference* , Edmonton, Canada, May-June 2003, 10–17.