Representing Scientific Experiments: Implications for Ontology Design and Knowledge Sharing

Natalya Fridman Noy and Carole D. Hafner

College of Computer Science
Northeastern University
Boston, MA 02115
{natasha, hafner}@ccs.neu.edu

Abstract

As part of the development of knowledge sharing technology, it is necessary to consider a variety of domains and tasks in order to ensure that the shared framework is widely applicable. This paper describes an ontology design project in experimental molecular biology, focusing on extensions to previous ontological models and frame-based formalisms that allow us to handle problems in the representation of experimental science knowledge. We define *object histories*, which are used to track substances through a series of experimental processes, including those which transform their participants from one category to another. We define object and process *complexes* – temporary configurations with features of their own. We present extensions to a frame-based formalism to support these features. Additional features of our frame formalism include *slot groups* for identifying sets of relations with common properties, and partial filler restrictions that combine knowledge of the most likely slot values with the ability to handle unexpected values We demonstrate how these extensions enable the use of (relatively) domain independent inference rules, support intelligent information retrieval, and improve the quality of query interfaces; and we descrıbe the translation of our formalism into Ontolingua.

Introduction

The field of ontology development has become very active in recent years on the premise that it will encourage and enable knowledge sharing and reuse (Fikes et al. 1991). It is generally accepted that building an ontology for any real-world domain is a difficult task, and this task could be greatly facilitated if it were possible to reuse and modify ontologies created by others. For example, a model for representing biology experiments could take advantage of general ontologies of time and space, whose axioms would support inferences such as: if process A occurred before a process B, then every substep of A occurred before every substep of B; or: a DNA molecule that is part of a chromosome inside the cell nucleus is also located inside the cell nucleus.

ARPA has sponsored a knowledge-sharing effort to develop methodology and software for the sharing and reuse of knowledge. Two results of this effort were: the Knowledge Interchange Format (KIF) (Genesereth and Fikes 1992), a computer-oriented language for knowledge interchange based on first-order logic and augmented by meta-knowledge and non-monotonic reasoning rules; and Ontolingua (Farquhar, Fikes, and Rice 1996; Gruber 1992) – a language for definıng ontologies that provides a frame-like syntax in addition to full first-order logic of KIF. Ontolingua has become a de-facto standard for representing ontologies. The Ontolingua Server (http://www-ksl-svc.stanford.edu:5915/), maintained by the Knowledge Systems Laboratory at Stanford University, contains tools for designing and analyzing ontologies as well as a large shared Ontology library.

In addition to standard formalisms and tools for knowledge sharing, some common ontological foundations are needed, so that intelligent agents can use a common vocabulary in a way that is consistent (but not necessarily complete) with respect to each agent's knowledge. Agreement on a shared ontological framework among researchers is crucial to enable different groups working on ontology design in different domains to communicate with each other and share their results (Gruber 1993; Guarino, Carrara, and Giaretta 1994). As part of the process of developing such a shared framework, it will be necessary to experiment with a variety of domains and tasks, in order to ensure that the shared framework is widely applicable.

This paper describes an ontology design project in the domain of molecular biology experiments, focusing on several areas where standard formalisms, tools, or frameworks needed to be extended. Our primary goal was to develop a representation framework for biology experiments described in the literature, which would be capable of supporting intelligent (i.e. semantic-based) question answering. This ontology provides support for inferences about complex substances, participants, conditions and effects of processes that can be used for information retrieval, planning, simulation, and other tasks. We believe that the challenges we faced, and the solutions we found, are relevant to other domains, particularly experimental sciences.

Experiments described in molecular biology papers are similar to cooking recipes. First, the ingredients (chemicals, bacteria, plasmids, etc.) are listed, followed by a description of the processes performed on the ingredients (mix, spin, separate, analyze, etc.) Thus, substances and processes are central to any ontology of experimental sciences. Some processes occur naturally, such as bacteria growth; others (experimental procedures) are set up and controlled by an experimenter. All processes take substances as their inputs and then change or observe some of their properties, destroy them, transform them into a different substance, etc. The substances themselves can be quite complex: they can be objects with elaborate internal structure, populations of molecules or cells, or mixtures of other substances. Sometimes this conglomerate will have a name of its own, and sometimes it will be just a temporary configuration of other substances. Processes also range from simple, "atomic" events, to complex configurations of events and actions dependent on each other.

Below we describe some extensions to previous ontological models and frame-based formalisms that allow us to handle problems in the representation of experimental science knowledge. We then demonstrate how these extensions enable the use of (relatively) domain independent inference rules, support intelligent information retrieval, and improve the quality of query interfaces. Finally, we describe the translation of our formalism to Ontolingua.

Elements of an ontological framework for experimental sciences

Representing knowledge about experiments, and molecular biology experiments in particular, presents its own unique challenges. Many of these are described in (Fridman Noy 1997). In this section, we describe elements of our ontological framework that address some of these challenges: *object histories, object complexes* and *process complexes*.

Object Histories

One of the major challenges in representing processes in experimental sciences is representing effects of transformations, in particular transformations that can change the category of their participants (called *category conversions*). When batter is baked, for instance, the `batter` object "migrates" to a different category, `cake`. The stuff the object was made from is still the same, but its classification has changed. From the standpoint of a knowledge model, we could represent this migration as the original participant (in this case, `batter`) ceasing to exist and the new object (`cake`) coming into existence. This, however, is not an accurate reflection of the way people think about the situation: there needs to be a link between the original object and the newly created one. For instance, when one asks if there is sugar in the cake, if this link exists, it can be inferred that since sugar was in the batter, it is now in the cake (possibly, in some transformed form). The fundamental notion in knowledge representation that every individual object is defined as an instance of a category seems incompatible with a universe where objects can gradually change their category as a result of a transformation. Thus, a straightforward process model that represents inputs (participants) and outputs (objects that come into existence) is inadequate for modeling conversions, because a) it does not represent the fact that the inputs no longer exist and b) it does not represent the relationship between the outputs and the original inputs, one of the most important relationships being the fact that the stuff the inputs were made from is now the stuff the outputs are made from.

Our solution to this problem is introducing *Object Histories*. The idea of histories was suggested in (Hayes 1990) and is used in Qualitative Physics (Collins and Forbus 1987; Forbus 1984). However, it is generally assumed that objects do not change their category or identity. We extend the notion of object histories to account for these changes and to trace substances as they go through processes, including category conversions. In our ontology an `Object History` for an object A consists of: information about the process that "gave life" to A; a list of complex objects that A was part of; a list of processes that A participated in; information about the process that destroyed A and substances that it was transformed into. This information does not need to be complete in order to be used for inferencing and query answering. For instance, in the earlier example of batter and cake, an object history for a `sugar` object can include `batter`, then the `mix` and `beat` processes, `bake` process, `cake` object, possibly an `eat` process (that would probably destroy the sugar, as we may not want to consider what becomes of sugar after we eat it).

Object Complexes

Another useful structure we introduce to represent biology experiments is `Complexes` - joining of several objects in a temporary configuration that, taken as a whole, has meaningful properties. The idea of a `Complex` was inspired by Individual Views in Qualitative Process Theory (Forbus 1984). The example Forbus uses is the *Contained-Liquid* Individual View. This Individual View describes liquid in a container and relations imposed on both objects (liquid and container) by this binding. For an example of a `Complex` from molecular biology, consider a binding complex that arises in gene transcription and includes a site on DNA ("promoter binding site") and an enzyme. The immediate significance of this chemical binding complex is a precondition for the gene transcription process.

Note that there is a subtle distinction between the relations of `Participants` in a `Complex` to each other and relation between a whole and its parts: the existence of a whole in the latter case generally is not contingent on the existence of its parts, i.e. a car without a wheel is still a car. It is different for a `Complex`. For example, in the *Contained-Liquid* example, if there is no liquid, or no container, the instance of a *Contained-Liquid* Complex does not exist.

Treating `Complexes` as first-class objects in the `Things` hierarchy, allows us to sub-categorize `Complexes` depending on the relations between their components: for example, we have such categories as Containment Complex (when one participant in a Complex contains all the others), or Attachment Complex (where participants are attached to each other).

Process Complexes

Similar to Complexes for objects, there are Process Complexes that represent a set of events that can be viewed as a whole with aggregate properties. In our ontology there are several sub-categories of Process Complexes, based on how sub-processes in it are related to each other. Firstly, sub-steps in a Complex could be sequential or parallel (i.e. executed simultaneously). Sub-class Sequence Complex represents a simple sequence of Processes. Sub-class Combination Complex is used to represent Complexes with parallel, dependent substeps. Chromatography is an example of the latter Complex. In many instances of chromatography its substeps, adding a substance at one end of a column and eluting it from the other end, are dependent since in order for something to be eluted from the bottom of the column, something needs to be added at the top. The rates of addition and elution are directly proportional: the more you add, the more substance is eluted.

Another sub-class of Process Complex is Technique Complex. This class is used to represent a complex of a main process and a technique used to achieve it. In a sense, the main process is the goal for the technique process, and the technique is the means of executing the main process. In this case, inputs and outputs of the two processes (the main one and the technique) are the same. Descriptions could be different though. Consider, for example a process "harvest by centrifugation". Harvest is the goal-process and centrifugation is a technique. Both processes have cells in growth medium as their input and cells without the medium as their output.

Extensions to frame-based formalism

The formalism that we used to represent our ontology is described in full in (Fridman Noy 1997). Here we present some of its more interesting features (mainly, slot groups, axiom groups and complex value restrictions) and show how they help to handle the structures described above. We demonstrated the formality and portability of this formalism in (Fridman Noy 1997) by translating it to Ontolingua. Some of the features of this translation are described below. We demonstrate that not everything can be translated *directly* into Ontolingua, but show how still to store the information so that it can be extracted later.

We took a standard frame-based formalism (see, for example, (Minsky 1981) or (Chaudhri et al. 1997)) as a basis and then extended it. So, each frame consists of a category name, a super-category name, and a list of slots and slot groups (slot groups are introduced and described below).

```
Process Chromatography: Combination Complex
  Participants:
      object instance-of Tangible-Thing
  Substeps:
      load-process instance-of Combine
      elute-process instance-of Separate
```

Figure 1. Partial definition of a Chromatography process as a sub-category of a Combination Complex

Slot groups

In order to capture various aspects of object and category change that are then automatically translated into object histories, as well as represent complexes (both, object complexes and process complexes), we introduce *slot groups*. Slot groups add an extra dimension to slot definitions when necessary. This allows slots that have similar ontological function to be grouped together. This semantic role can then be employed in the inference rules and axioms. The slot group Participants in a process is a good example of this phenomenon. Each member of a Participant slot group is a slot in itself, with its own name that can be referred to in axioms and inference rules (e.g., object, growth-medium) and value restrictions. At the same time, we can refer to all the process Participants as a whole. Similarly, some processes have a slot group for newly created objects. When an instance frame of such a process is created in a knowledge base, we automatically create frames (and corresponding object histories) for the new objects. At the same time, axioms representing effects of a process can contain conditions on properties of each of these newly created objects.

A slot group consists of a slot group name followed by a list of slot definitions, where each slot has a name and, possibly, some value restrictions (described later).

```
<slot-group> ::= slot-group-name: {<slot >}*
<slot> ::= slot-name [value-restriction]
```

Specific slot groups and the inferences they license are determined by the ontology. The number of slot groups is usually small and reflects only very high-level assumptions about the data. As an example we will describe here some of the slot groups used in our ontology.

Each Process has a Participant group that contains objects participating in the process. They are differentiated inside by various roles reflected in the corresponding slot names (e.g., solution, catalyst, etc.).

Substeps in a Process Complex is also a slot group. For example, in Chromatography, which is a sub-class of Combination Complex consisting of two simultaneous inter-related processes (loading into a column and eluting from a column), the two substeps with their corresponding roles are: load-process and elute-process (see Figure 1).

A number of slot groups are used to trigger corresponding updates in object histories by the inference rules. For example:

- Objects-created: list of new objects created as a result of a process, their categories and value restrictions.

```
Process Transform : Process
  Participants:
      original instance-of Tangible-Thing
      catalyst instance-of Chemical
  Objects-created:
      new instance-of Tangible-Thing
  Objects-converted:
      conversion (original, new)
```

Figure 2. Sample definition of a `Transform` process.

Instances of each of these objects are created. This, in turn, triggers creation of corresponding object histories.

- `Objects-converted`: list of *conversion* slots. The value of each slot is a pair: original object (or list of objects), and the object it was converted into.

Consider, for example, a simple transformation process that transforms some *original* substance into some *new* substance (a chemical reaction with a catalyst present, for instance) presented in Figure 2. Here an instance of `Tangible Thing` is created (`new`) along with an instance of its object history. An instance of this `Transform` process will be put as the final process in the `original`'s object history and as the first process in the `new`'s object history. In the object history of `new` it would also be noted that it was derived from `original`.

Axiom groups

Most frame-based formalisms allow a set of axioms to be associated with a concept definition. Axioms can be used to specify restrictions on the values of the slots that cannot be specified by simple slot restrictions, such as conditions involving dependencies among the values of several slots. Usually an axiom associated with a frame stipulates that a condition must be true for any instance of the concept. However, in modeling knowledge about processes, it is useful to distinguish axioms that describe preconditions of a process and axioms that describe process' effects.

For instance, an effects axiom for a `mix` process can state that all the inputs are now ingredients in the mixture and any process applied to the mixture (such as heating) is indirectly applied to the original inputs also. Then, if a user asks whether a certain sugar (assuming it was put into the mixture) was ever heated, the answer will be positive. However, if the mixture was heated before the sugar was added to it, then the answer should be negative.

We realize this distinction between axioms by allowing two groups of axioms in a process frame: one preceded by a keyword `Conditions` and the other by a keyword `Effects`. This distinction can then be used by inference rules.

```
Thing DNA : Nucleic-Acid
  Type        chromosomal genomic ...
  Source      synthesized ...      or
              instance-of bacteria
  Composition
       single-stranded double-stranded
  Labeled-or-not    no             or
                    instance-of label
```

Figure 3. Sample definition of the category DNA. It has four simple slots with various kinds of value restrictions.

Specifying slot value restrictions

Another feature of our frame-based formalism is an expanded way of specifying value restrictions for a slot.

Figure 3 shows a sample definition of the category DNA. DNA, for example, can be *chromosomal*, *genomic*, or some other, unspecified, type. So, the `Type` slot has a list of fillers with ellipsis that indicates an open value set. DNA can be *synthesized* or come from bacteria. Thus, the `Source` slot can either contain the filler value *synthesized* (or some other, unspecified filler), or an instance of a `Bacteria` class. For the `Composition` slot possible fillers are limited to: `single-stranded` or `double-stranded`. A DNA molecule might not be labeled radioactively (in which case, the filler for the last slot is *no*), or it might be labeled by some radioactive `label` (which should then fill the value of the slot).

The use of open value sets for slot fillers accomplishes several things: Having a list of specific expected fillers can help in natural-language processing, since they can be used to infer the presence of a frame from the presence of one of its fillers (for instance, the presence of DNA from the use of *genomic*). The list of specific fillers also can be used to provide users with a set of suggested values to choose from when interactively specifying a query. On the other hand, allowing other possible values accounts for evolving domains (which experimental sciences certainly are): if new values are invented, they would easily fit into the existing knowledge base, since there was no strict limit on what can fill the slot.

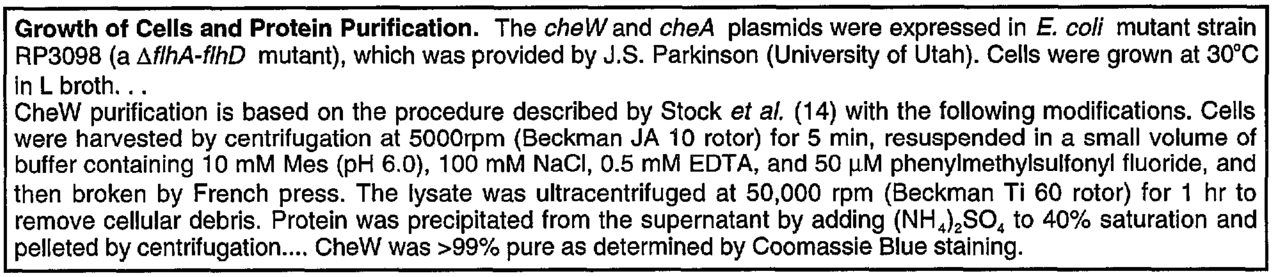
In our formalism we also allow for various combinations of restrictions on slot values. The value restriction on a slot is either a list of possible filler values or specification of a category that the slot value should belong to, or both.

```
<value restriction> ::=
    <list of fillers> | <class restriction> |
    <list of fillers> or <class restriction>
<class restriction> ::=
    instance-of class-name {or class-name}*
```

If a list of fillers for a slot value is specified and it is not followed by ellipsis, the range of values for the slot is limited to the values from this list. If ellipsis follows the list of fillers, the slot can take on other values as well (in the latter case, the list of fillers usually represents the most likely values):

```
<list of fillers> ::= {filler-value }+ [...]
```

This added richness in specification is used in the query interface for providing an easier way for the user to fill in query frames.



Growth of Cells and Protein Purification. The *cheW* and *cheA* plasmids were expressed in *E. coli* mutant strain RP3098 (a Δ*flhA-flhD* mutant), which was provided by J.S. Parkinson (University of Utah). Cells were grown at 30°C in L broth. . .

CheW purification is based on the procedure described by Stock *et al.* (14) with the following modifications. Cells were harvested by centrifugation at 5000rpm (Beckman JA 10 rotor) for 5 min, resuspended in a small volume of buffer containing 10 mM Mes (pH 6.0), 100 mM NaCl, 0.5 mM EDTA, and 50 μM phenylmethylsulfonyl fluoride, and then broken by French press. The lysate was ultracentrifuged at 50,000 rpm (Beckman Ti 60 rotor) for 1 hr to remove cellular debris. Protein was precipitated from the supernatant by adding $(NH_4)_2SO_4$ to 40% saturation and pelleted by centrifugation.... CheW was >99% pure as determined by Coomassie Blue staining.

Figure 4. A potential target for retrieval (an excerpt from (Gegner and Dahlquist 1991)).

Using the framework for inference and query answering

Inference using slot groups

The slot groups and axiom groups in an ontology should be small in number and relatively domain independent. This means that knowledge about slot groups can be expressed in high-level inference rules that do not depend on specific frame attributes. We will present here a few examples of inference rules that use slot groups.[1]

For the first example consider again the `Transform` process in Figure 2, with slot groups: `Participants`, `Objects-created`, `Objects-converted`. Each slot group triggers corresponding updates in the object histories. For instance, each `Participant` has an instance of the `Transform` process added to its object history:

```
(=> (Process ?x)                              (1)
  (∀ (?y)
   (=> (member-of ?y (Participants ?x))
        (member-of ?x
         (Processes (Object-history ?y))))))
```

For members of the `Objects-created` group, a new instance and object history are created, with the `Transform` process as the creator:

```
(=> (Process ?x)                              (2)
  (∀ (?y)
   (=> (member-of ?y (Objects-created ?x))
        (= (Creator-Process (Object-history ?y))
           ?x))))
```

Another example involves `Process Complexes`. In a `Combination Complex`, all the `Substeps` occur at the same time and temperature (recall, that `Combination Complex` consists of a number of simultaneous inter-related substeps). This can be expressed with the following rule:

```
(=> (Combination-Complex ?x)                  (3)
  (∀ (?y ?z)
     (=> (and (member-of ?y (Substeps ?x))
              (member-of ?z (Substeps ?x)))
         (= (duration ?x) (duration ?y)
            (duration ?z)))))
```

That is, knowing the `duration` for either the whole `Combination Complex`, or any of its `Substeps`, allows us to fill in this value for all the others. The same can be stated for the `temperature` of these processes, or, say, `Object` in a `Technique Complex` and its `Substeps`. This rule does not rely on the roles of specific `Substeps`, which are themselves slots in the `Process Complex` frame.

Inference in query answering

Intelligent Information Retrieval was the initial goal of our ontology design effort, so the extent to which the ontology supports inferencing for this type of retrieval is an important measure of our success. In this section we will show how the knowledge encoded in our ontology and described in the previous section can be used to answer queries more intelligently. Two retrieval heuristics are described here: indirect match of transformants and technique abstraction.

Example paragraph

The queries below will be illustrated by an excerpt from (Gegner and Dahlquist 1991) presented in Figure 4. This excerpt describes a sequence of steps to purify CheW protein from a certain strain of *E. coli* bacteria (namely, strain RP3098).

The sequence starts out with a strain of *E. coli* bacteria which is grown to get the necessary number of cells. The grown cells contain CheW protein which now needs to be purified. The purification process consists of first breaking the cell walls to create *lysate* (an unstructured mixture of pieces of the walls and cell elements) and then gradually removing substances other than CheW from the mixture and achieving higher and higher concentration of CheW in the mixture that remains. Along the way, various substances (buffers, chemicals) are added to the mixture and then removed, carrying some of the unwanted stuff away with them. In the end, all that is left is a mixture 99% of which is CheW protein.

We will now describe two queries that can be answered better by using the inference rules presented in the previous section. As will be discussed in these examples, this approach increases the *recall* of the information retrieval as compared to other knowledge-based systems by utilizing the more extensive information stored in our knowledge base. At the same time, this approach also increases the *precision* of information retrieval compared to statistical, keyword-based systems that would bring many incorrect

[1] We present declarative inference rules here; they get their operational semantics when used in the inference engine

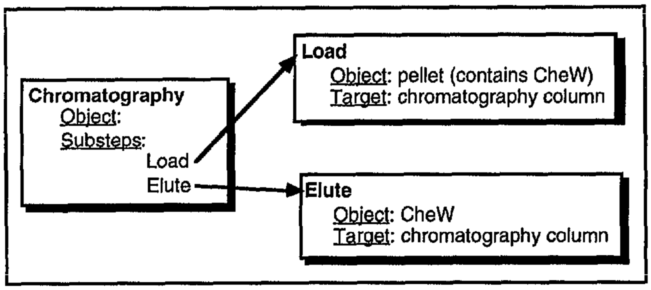

Chromatography
Object:
Substeps:
Load
Elute
Load
Object: pellet (contains CheW)
Target: chromatography column
Elute
Object: CheW
Target: chromatography column

Figure 5. Illustration of implicit technique recognition. Arrows represent reference pointers in this case.

answers based only on presence and/or proximity of words in a sentence or paragraph.

Indirect match of transformants

Consider the following query:

> *Show me the papers that describe RP3098 cells being ultracentrifuged.*

In the paragraph in Figure 4 the ultracentrifugation process was applied to the lysate (and not to the RP3098 cells). So, this paper might not be brought up as an answer. However, since the lysate is a direct transformant of the cells and thus the ultracentrifugation was indirectly applied to the cells, it could be desirable to present this paragraph as an answer to the query above. In our system, when an instance of a `break` process that produced the lysate, is created, the object history for this instance of `lysate` is updated to contain the `cells` as the `Original-object` for the lysate. Thus, when the `lysate` is put as an `object` in the `ultracentrifugation` frame, it can be easily inferred, that this process is applied to the direct transformant of the `cells`.

Technique abstraction

Consider the following sentence from Figure 4:

> *The pellet was ... loaded onto a Whitman DE-52 column. Protein was eluted from the column with a linear gradient of ...*

Even though it is not explicitly mentioned here, this sequence of events describes a chromatography process. Chromatography is the *complex* of these experiment substeps (see Figure 5). It is easy to imagine a query pertaining to this technique:

> *Show me the papers where chromatography was used in the process of purifying CheW*

So, this paper should be retrieved as the result. However, this answer is possible only if we consider the `Chromatography Combination Complex`, the presence of which could be inferred by the presence of two other processes, `load` and `elute`. Besides, the object of the two process (load and elute) - the protein, would be automatically placed in the object of the Process Complex, by a rule similar to rule (3) in the previous section.

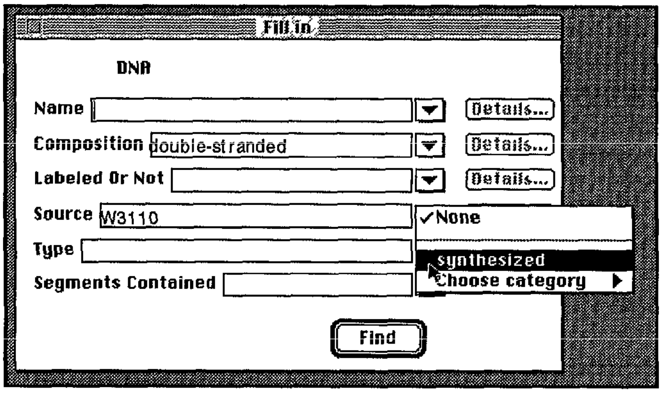
Using the framework for user interaction

To evaluate the practical usefulness of the ontology, we implemented a proof-of-concept prototype of an intelligent information retrieval system: M&M Query System, designed to assist biologists in accessing on-line texts of the Materials and Methods sections of research papers.

First, research papers in the database are annotated with frames based on the corresponding knowledge model (in turn, using the features described here). Frames along with the texts of papers they are linked to, are stored in a database. This database can then be used by biologists to search for specific information in the papers. After a query is entered (in the form of a concept list or filled-in frames), it is presented to the search engine that matches it to the frames in the knowledge base. The result comes back in the form of a list of relevant papers. The user can then choose any paper (or papers) to be displayed. To point out the more relevant part of the paper and to provide simple feedback of why this particular paper was brought up, the parts of the paper associated with the frame are highlighted.

One of the query modes in our system is *frame fill-in query*, where a user is given a (possibly simplified) frame for a concept s/he is interested in. The user then specifies some of the values in the frame slots to restrict the search field. These values are matched with the ones in the frame database. This query mode makes use of some of the features of our frame formalism. Figure 6 presents an example of such a query. There are several ways in which the domain knowledge (specified when classes are defined) is used to assist the user in filling out slot values: for instance, if the class definition specifies a list of fillers for a specific slot, this list appears in the pop-up menu next to the slot; any value from this list can be chosen as a fill-in; if there is a class restriction on the slot, the list of these categories can be presented to the user as well. Otherwise, s/he can browse the list of all categories to fill in the value for the slot

An important feature of this interface is that it does not depend on the specific knowledge in the knowledge base. Any frame-based knowledge model, as long as it follows


Fill in

DNA

Name

Composition double-stranded

Labeled Or Not

Source W3110

Type

Segments Contained

Figure 6. A frame fill-in dialog for a DNA class (short form).

```
(DEFINE-FRAMEDNA
       :OWN-SLOTS
      ((ARITY 1) (DOCUMENTATION "Describes properties of DNA")
        (INSTANCE-OF CLASS)
       (SUBCLASS-OF NUCLEIC-ACID))
        TEMPLATE-SLOTS
    ((SOURCE (SLOT-FILLERS '(synthesized)) (MORE-FILLERS-ALLOWED TRUE) (VALUE-TYPE BACTERIA))
      (TYPE (SLOT-FILLERS '(chromosomal genomic)) (MORE-FILLERS-ALLOWED TRUE))
     (COMPOSITION (SLOT-FILLERS '(double-stranded single-stranded)))
      (LABELED-OR-NOT (SLOT-FILLERS '(no)) (VALUE-TYPE LABEL))))
```

Figure 7 Ontolingua definition for DNA *class from Figure 3 (http·//www-ksl-svc.stanford edu:5915/doc/ontolingua/reference-manual).*

the formalism, can be plugged in this system and the user will be guided through the new hierarchies and new frames.

Translation to Ontolingua

To validate the formality and portability of our formalism, we translated the ontology into Ontolingua (Farquhar, Fikes, and Rice 1996) which has become a standard repository for ontologies for knowledge sharing.

As described above, Ontolingua supports a frame-based formalism, and, thus, it lends itself easily as a translation target for our formalism. Although not all the features of our formalism could be translated directly into Ontolingua, the information could still be stored (in most cases, using facets) and then extracted back if necessary.

Facets in Ontolingua are relations associated with the slots that allow specification of various constraints on the slots. For instance, commonly used Ontolingua facets include `Slot-Value-Type` to specify the class the slot values should belong to, or `Slot-Cardinality` to constrain the cardinality of a slot. We add a set of extra facets to encode features that can be expressed in our formalism but not in the standard frame formalism. These features include slot belonging to a particular slot group, list of possible slot fillers, etc. Figure 7, for instance, demonstrates how the `DNA` category from Figure 3 is represented in Ontolingua directly. In this example, every slot has a facet added to it. We associate a facet `Slot-Fillers` with a slot if a list of fillers is available. The value of the facet is the list itself. If fillers not from the list are allowed (denoted by ellipsis in our formalism), a facet `More-Fillers-Allowed` with a `true` value is added to the slot (e.g., `Source` slot in Figure 7).

The formalism features described in this paper require two groups of extra facets:

- For each slot group, there is a facet *<Slot-Group-Name>*`-Group-Member` for specifying a slot group that a slot belongs to. When a slot belongs to a certain slot group, the corresponding facet is then associated with this slot and is given a `True` value. Since the number of slot groups is limited and is one of characteristics of an ontology, these facets should be defined before defining any of the other frames in a knowledge base.
- Facets are used to specify restrictions on slot values that go beyond simple class restriction. `Slot-fillers` facet specifies the list of slot-fillers for a slot. When the facet is associated with a slot, its value is a list of possible fillers. A facet `More-Fillers-Allowed` is added to a slot and given a value `True` if values not from the `Slot-Fillers` list could also be used for the slot (open value set).

There were a few other facets that were necessary to encode all the information that our formalism allows to specify, in Ontolingua. Even though this encoding did not allow these features to be available directly (since Ontolingua does not have provisions for them), all of them could be stored in it and then extracted back when necessary. That is, frames encoded in our formalism can be ported into Ontolingua and exported back without loss of information

Related work

Recently a number of research groups have created ontologies for different domains and purposes. In (Fridman Noy and Hafner 1997) we summarize and compare the contents, structure, design and evaluation methodologies and applications of ten projects representing the range of current work. The features of our ontology described in this paper address issues that were not fully or not at all addressed in these earlier projects.

In addition to research explicitly aimed at ontology design, research in Qualitative Physics (Collins and Forbus 1987; Hayes 1990); and in particular Qualitative Process Theory (Forbus 1984) has influenced our ontology framework. Our idea of `Complexes` is related to the notion of Individual Views from QPT. Reifying this notion to be a first-class object in the hierarchy of `Things` allows us to sub-categorize `Complexes` based on relations between their components. We also extend this notion into the Process sub-ontology and introduce `Process Complexes`. Our use of Object Histories extends the classic notion of histories to account for the fact that objects change not only their properties, but also their categories as a result of processes. Object histories in our model trace substances through *category conversions* as well as other processes.

Some work in description logic explores the idea of extending frame-based formalisms for more elaborate description of possible slot-fillers (see, for example, (Brachman et al. 1991)). We believe that our approach simplifies these specifications as compared to, say, CLASSIC, at the same time allowing for the richness of open-value sets to assist in natural-language processing and to account for evolving domains. Our treatment of process configurations also shares some characteristics with the *components* in (Clark and Porter 1997), which are abstract "mini-theories", or patterns of interactions between concepts. Each *component* consists of participants, their roles, and axioms. However, *components* are not related in hierarchical fashion (which makes re-use of parts of the descriptions more difficult).

Conclusions

In this paper we presented extensions to previous ontological models and standard frame-based formalism that are necessary to adequately represent knowledge about scientific experiments described in the literature. We also showed how these extensions can improve the quality of query answering and user interfaces.

The ontology elements described here include object histories, and object and process complexes. The formalism extensions are based on the ontology that we developed and include slot groups, axiom groups, and complex value restrictions on the slots. Slot groups are used to represent object and process complexes (temporary configurations with features of their own) and object histories (used to trace substances through processes including the processes that change categories of their participants). Axiom groups are used to distinguish between conditions that need to be true for a process to take place and those that are true after the process. All these features of the ontology are, in turn, used by an inference engine for query answering.

This formalism was used in an a prototype of an Intelligent Information Retrieval System (M&M Query System).

As foundations for shared ontologies and formalisms are considered, the requirements of such a large and important domain as experimental sciences should certainly be considered and accounted for in such an effort.

Acknowledgments

The authors thank the reviewers for their feedback. This research was supported in part by the National Science Foundation under grants IRI-9117030 and IRI-9633661.

References

Brachman, R. J., McGuiness, D. L., Patel-Schneider, P. F., Resnik, L. A., and Borgida, A. 1991. Living with CLASSIC: When and how to use KL-ONE-like language. *Principles of Semantic Networks*. J. F. Sowa, ed.: Morgan Kaufmann: 401-456.

Chaudhri, V., Farquhar, A., Fikes, R., Karp, P., and Rice, J. 1997. The Generic Frame Protocol 2.0, Technical Report, KSL-97-05, Knowledge Systems Laboratory, Stanford University

Clark, P. and Porter, B. 1997. Building Concept Representations from Reusable Components. In Proceedings of Fourteenth National Conference on Artificial Intelligence, 369-376. Providence, RI: AAAI Press.

Collins, J. W. and Forbus, K. D. 1987. Reasoning About Fluids Via Molecular Collections. In Proceedings of Sixth National Conference on Artificial Intelligence, 590-594. Seattle, WA: AAAI Press.

Farquhar, A., Fikes, R., and Rice, J. 1996. The Ontolingua Server: a Tool for Collaborative Ontology Construction. In Proceedings of Tenth Knowledge Acquisition for Knowledge-Based Systems Workshop. Banff, Canada.

Fikes, R., Cutkosky, M., Gruber, T., and Baalen, J. v. 1991. Knowledge Sharing Technology Project Overview, KSL 91-71, Knowledge System Laboratory, Stanford University

Forbus, K. D. 1984. Qualitative Process Theory. *Artificial Intelligence* 24: 85-168.

Fridman Noy, N. 1997. Knowledge Representation for Intelligent Information Retrieval in Experimental Sciences. Ph.D. diss., College of Computer Science, Northeastern University.

Fridman Noy, N. and Hafnèr, C. 1997. The State of the Art in Ontology Design: A Survey and Comparative Review. *AI Magazine* 18(3): 53-73.

Gegner, J. A. and Dahlquist, F. W. 1991. Signal transduction in bacteria: CheW forms a reversible complex with the protein kinase CheA. *Proceedings National Academy Sciences* 88: 750-754.

Genesereth, M. R. and Fikes, R. E. 1992. Knowledge Interchange Format, Version 0.3, Reference Manual, Logic-92-1, Knowledge Systems Laboratory, Stanford University

Gruber, T. R. 1992. Ontolingua: A Mechanism to Support Portable Ontologies, Knowledge Systems Laboratory, Stanford University

Gruber, T. R. 1993. Toward Principles for the Design of Ontologies Used for Knowledge Sharing, KSL 93-04, Knowledge Systems Laboratory, Stanford University

Guarino, N., Carrara, M., and Giaretta, P. 1994. Formalizing Ontological Commitments. In Proceedings of Twelfth National Conference on Artificial Intelligence (AAAI '94), 560-568. Seattle, Washington: AAAI Press/ The MIT Press.

Hayes, P. J. 1990. Naive Physics I: Ontology for liquids. *Readings in Qualitative Reasoning about Physical Systems*. D. S. Weld and J. de Kleer, eds.: Morgan Kaufmann, San Mateo, CA: 484-502.

http://www-ksl-svc.stanford.edu:5915/ Stanford KSL Network Services. Palo Alto, CA: Stanford University Knowledge Systems Laboratory.

http://www-ksl-svc.stanford.edu:5915/doc/ontolingua/reference-manual Ontolingua System Reference Manual: Knowledge Systems Lab, Stanford University.

Minsky, M. 1981. A Framework for Representing Knowledge. *Readings in Knowledge Representation*. R. Brachman and H. Levesque, eds.: Morgan Kaufmann Publishers, INC: 245-262.