SPEX: A Second-Generation Experiment Design System

Yumi Iwasaki and Peter Friedland

Heuristic Programming Project, Computer Science Department
Stanford University, Stanford, Ca. 94305

Abstract

The design of laboratory experiments is a complex and important scientific task. The MOLGEN project has been developing computer systems for automating the design process in the domain of molecular biology. SPEX is a second-generation system which synthesizes the best ideas of two previous MOLGEN hierarchical planning systems: stepwise refinement of skeletal plans and a layered control structure. It has been tested successfully on several problems in the task domain and promises to serve as a testbed for future work in explanation, experiment debugging, and empirical evaluation of different basic design strategies.

1. Introduction

Experiment design is the process of choosing an ordered set of laboratory operations to accomplish some given analytical or synthetic goal. This process is one of the fundamental tasks of experimental scientists; it involves large amounts of domain expertise and specialized design heuristics. The design of such experiment plans has been one of the fundamental research efforts of the MOLGEN project at Stanford. SPEX (Skeletal Planner of EXperiments) is a second-generation experiment design system. It is a synthesis of the best ideas of two previous planning systems, and will serve as a "laboratory" for the empirical testing of design strategies at many levels. SPEX will also be used for MOLGEN work on experiment verification, optimization, and debugging. This paper is a report of the work in progress on SPEX.

1.1. Previous MOLGEN Planning Systems

Friedland developed an experiment planning system using the methodology of hierarchical planning by progressive refinement of skeletal plans [1] [2]. A skeletal plan is a linear sequence of several abstract steps; actual plans are generated by refining each of the abstract steps to use specific techniques and objects by going down a general-to-specific hierarchy of laboratory operations stored within a knowledge base built by experiment molecular biologists. Friedland's experiment planner chooses a skeletal-plan suitable for the given goal of an experiment and refines each step by choosing the best specialization of the laboratory method at each level of abstraction.

Stefik developed another experiment design system [3]. His hierarchical planner first constructs an abstract plan by simple difference reduction, and then generates a specific plan from that abstract plan by propagating constraints. It has a multi-layered control structure to separate out different levels of decisions to be made by the planner [4].

The two systems were complementary. Friedland's system made efficient use of large amounts of domain knowledge to produce practical, but not necessarily optimal experiment designs for a large subset of analytical tasks in molecular biology. The assumption of near-independence of abstract plan-steps worked well in the great majority of cases. Stefik's system took much longer to plan reasonable experiments, but worked better when plan-steps were highly dependent and kept a much richer description of the planning process, this because of the well-designed control structure.

SPEX was developed to synthesize two fundamental ideas from these planners, namely Friedland's skeletal-plan refinement and Stefik's multi-layered control structure, in the hope of making further progress in the construction of a design system that would be used by experts. In addition, SPEX has a greatly enhanced capacity to simulate the changing world state during an experiment. The remainder of this paper describes the layered control structure and the simulation mechanism used by SPEX.

Like Friedland's and Stefik's systems, the knowledge base of SPEX is constructed using the Unit System [5], a frame-based knowledge representation system developed in the MOLGEN project. In SPEX, the Unit System is also used to represent a trace of the planning process and the changing states of objects in the world.

2. Method

2.1. Layers of Control

In order to leave a trace of a planning process, it is necessary to identify the different kinds of operations the planner is expected to perform and to represent the entire process as a sequence of such operations and their consequences. The notion of a multi-layered control structure was introduced and operations at three different levels in the planning process were identified and represented within SPEX.

The bottom level, called the *Domain Space* by Stefik, consists of the objects and operators in the task domain, termed *lab-steps*. They are experiment goals, skeletal-plans, and laboratory techniques. On top of the Domain Space exists the *Design Space*, which consists of the various planning tasks performed by SPEX, for example, the tasks of finding an appropriate skeletal-plan, expanding a skeletal-plan, or refining a technique. These are termed *plan-steps*. When such tasks are executed, they create or delete lab-steps in the Domain Space, or create new tasks in the Design Space. Finally, the third layer, the *Strategy Space*, consists of several different strategies to control the execution of tasks in the Design Space.

Different types of decisions are made in the three different spaces. In the Domain Space, decisions are biology-oriented. The two major types of decisions are environmental, i.e. whether environmental conditions and structural properties allow a given laboratory technique to be used, and detailed selection, i.e. the process of deciding on the basis of selection heuristics among several techniques all of which will carry out the specified experimental goal. In the Design Space, decisions are more goal-oriented. These decisions relate the specific goal of a step in the skeletal plan to the potential utility of a laboratory technique for satisfying that goal. Finally, in the Strategy Space, choices are made among various design strategies, whether to refine in a breadth-first, depth-first, or heuristic manner, for example.

2.2. Modeling World Objects

Neither previous MOLGEN design system did a thorough job of monitoring the state of the laboratory environment during the simulated execution of an experiment plan. But, the laboratory environment, i.e. the physical conditions and the molecular structures involved in the experiment, undergoes changes in the course of carrying out the experiment. Therefore, for the planning of an experiment design to be consistently successful, it is essential to predict the changes caused by carrying out a step in the plan. SPEX simulates those changes, using the part of its knowledge base that contains the potential effects of carrying out an laboratory technique. The predicted states of the world at certain points in the experiment are used as part of the selection criteria in choosing the appropriate technique for the next step in the plan. They are also used to make sure that the preconditions for a chosen technique are met before the application of the technique. If any of the preconditions are not satisfied, a sub-plan to modify the world state is produced and inserted in the original plan.

3. Implementation

SPEX is implemented in Interlisp, making extensive use of the Unit System in order to represent each operation in different planning spaces. In the Design Space, there are at present seven types of plan-steps: obtaining the goal of an experiment, choosing a skeletal plan, expanding a skeletal plan, refining a laboratory technique, ruling in or out a laboratory technique, comparing alternative laboratory techniques, and checking the world state to verify that the preconditions for application of a technique are satisfied.

When a new task needs to be generated, a plan-step unit is created to represent the task and to leave a record of SPEX's performance. A plan-step unit has slots containing such information as when the task was created, when it was performed and what the consequence was. Some of the slots in the prototypical plan-step unit are shown in Figure 3-1.

```
Unit: PLAN-STEPS
-----
CREATED-BY:        <UNIT>
```
A pointer to the unit describing the plan-step which caused this plan-step to be created.

```
-----
LAB-STEP:          <UNIT>
```
A pointer to the lab-step directly created by this plan-step.

```
-----
NEXT-PLAN-STEP:    <UNIT>
```
The next plan-step fetched.

```
-----
LAST-PLAN-STEP:    <UNIT>
```
The previous plan-step fetched.

```
-----
STATUS:            <STRING>
```
The status of this plan-step; either "succeeded", "postponed," or "created".

Figure 3-1: Slots in a plan-step

Lab-steps in the Domain Space are also represented by units. Lab-step units initially record the Domain Space results of decisions made in the Design Space, but may also record later decisions made in the Domain space. Some of the slots in the prototypical lab-step unit are shown in Figure 3-2.

```
Unit: LAB-STEP
-----
FOLLOWING-STEP:   <UNIT>
```
A pointer to the next lab-step in the plan.

```
-----
PRECEDING-STEP:   <UNIT>
```
A pointer to the preceding lab-step in the plan.

```
-----
CREATED-BY:       <UNIT>
```
A pointer to the plan-step which created this lab-step.

```
-----
STATUS:           <STRING>
```
When a lab-step has just been created, it is given the status "just-created". Later when it is ruled in or out, the status is changed to "ruled-in" or "ruled-out".

Figure 3-2: Slots in a lab-step

SPEX uses an agenda mechanism to keep track of all the pending tasks. The Strategy Space of SPEX consists of simple strategies about how to choose the task from the agenda. Currently, a strategy is chosen at the beginning of a session by the user and it is used throughout the session. There are three strategies currently available in the system. With Strategy 1, the agenda is used like a queue and tasks are fetched in first-in, first-out manner. With Strategy 2, the agenda is normally used like a stack and tasks are fetched in first-in, last-out fashion. With Strategy 3, the plan-steps are given priorities according to their types. Tasks of a higher priority are executed before any tasks of lower priorities. We are currently experimenting with various prioritizing schemes for the different plan-step types.

3.1. Simulating the World State

At the beginning of a planning session, the user is asked to provide a description of the world objects. This includes the current physical environment (temperature, pH, etc.) of the experiment and what he knows about the detailed molecular structure of his experimental objects, normally nucleic acid sequences. A unit is created to represent the initial description for each world object. When a skeletal-plan is expanded to individual steps, units are created to represent the simulated state of each world object before and after application of each step. This is done by utilizing simulation information stored in the laboratory-techniques hierarchy of the knowledge base. Figure 3-3 shows some of the slots in the prototypical unit for describing nucleic acid structures. This unit is called DNA-STRUCTURE. Its composition has evolved over several years of collaborative work by several of the molecular biologists associated with the MOLGEN project. It represents all of the potential information a scientist might wish to supply about a particular nucleic acid structure. In an actual experiment, a scientist would be able to fill only a few of the over 50 slots in the prototypical unit. Figure 3-4 shows the actual values given to the sample slots during the planning of an experiment design by SPEX.

```
Unit: DNA-STRUCTURES
-----
STRANDEDNESS:     <STRING>
    One of: ["HYBRID" "SS" "DS"]
-----
LENGTH:             <INTEGER>    [1 10M]
    MEASUREMENT UNIT     BASE-PAIRS
-----
#-TERMINI:          <INTEGER>    [0 4]
-----
#-FORKS:            <INTEGER>    [0 2]
-----
TOPOLOGY:           <STRING>     One of:
["DELTA-FORM" "THETA-FORM" "EYE-FORM" "Y-FORM" "LINEAR"
          "CIRCULAR"]
-----
TYPE:               <STRING>
    One of: ["HYBRID" "RNA" "DNA" ]
```

Figure 3-3: Slots in DNA-STRUCTURE

```
Unit: STRUC-1
-----
STRANDEDNESS:      <STRING>      "SS"
-----
LENGTH:            <INTEGER>     400
    MEASUREMENT UNIT     BASE-PAIRS
-----
#-TERMINI:         <INTEGER>     2
-----
#-FORKS            <INTEGER>     0
-----
TOPOLOGY           <STRING>      "LINEAR"
-----
TYPE:              <STRING>      "DNA"
```

Figure 3-4: Slots in an instance of DNA-STRUCTURE

4. Results and Discussion

SPEX has been successfully tested on several of the same problems used as test cases in Friedland's thesis [1]. Combining the ideas of skeletal-plan refinement and multi-layered control structure proved useful in keeping the Domain Space efficiency of Friedland's system, while introducing the Planning and Strategy Space flexibility of Stefik's system. The greatly improved simulation mechanism increased the reliability of the decisions made and allowed SPEX to suggest, in detail, ways of correcting low-level incompatibilites between chosen laboratory techniques.

SPEX keeps a clear trace of all the decisions made during the experiment design process. This trace will now be used for a variety of purposes in ongoing MOLGEN research. One goal of this research to add a detailed explanation capability to SPEX. Earlier MOLGEN experiment design systems only provided explanation by listing the domain-specific rules used to make decisions at the level of laboratory techniques. The planning trace and the modular nature of the different planning spaces will allow this explanation facility to be expanded. The user will be able to direct his questioning to domain, design, or strategic motivations. We envision an initial explanation facility similar to that used by MYCIN [6]. The user asks questions using a set of key words as "why", "when", "how", etc. We believe that besides promoting the use of SPEX among expert users for whom full explanations are a necessity, the explanation facility will also greatly facilitate debugging of the knowledge base.

A necessary complement to the task of experiment design is the task of experiment debugging. Initial experiment designs produced by even the very best of scientists rarely work perfectly the first time. We are certain that the same will be be true of SPEX-produced experiment designs. Given the sequence of techniques employed in an experiment, a debugger will compare the predicted states of the world before and after each step and the actual world states during the experiment (supplied by the molecular biologist user). Then, it will point out the steps which might possibly have gone wrong and suggest solutions: alternative techniques or a remedial procedure to be performed. If the "buggy" experiment design had come from a human scientist, then the debugging information will enable him to correct his personal "knowledge base." In a similar manner, we believe that the comparison of actual laboratory results to the previously predicted results should allow the automatic improvement of the knowledge base in the case of a SPEX-generated experiment design. We do not anticipate major difficulties in building the experiment debugger given the existing mechanisms of SPEX.

In summary, SPEX represents a synthesis of several methodologies for design, namely skeletal-plan refinement and the multi-layered control structure. It is a framework for a general-purpose design testing and debugging system, which can be easily tailored to do planning in any specific application domain. There are no molecular-biology specific mechanisms inherent in SPEX; all of the domain-specific knowledge is in the associated knowledge base. SPEX can also be used to test different basic design strategies by the implementation of many additional strategies in its Strategy Space.. We believe that the best way to determine the efficacy of the many different potential strategies is empirical, and SPEX will be useful as a laboratory for these experiments.

5. Acknowledgments

This work is part of the MOLGEN project, a joint research effort among the Departments of Computer Science, Medicine, and Biochemistry at Stanford University. The research has been supported under NSF grant MCS80-16247. Computational resources have been provided by the SUMEX-AIM National Biomedical Research Resource, NIH grant RR-00785-08, and by the Department of Computer Science.

We wish to thank our many enthusiastic MOLGEN collaborators for their assistance in this work. We are especially grateful to René Bach and Larry Kedes for providing the molecular biology expertise necessary to test SPEX and to Bruce Buchanan and Mike Genesereth for advice on the artificial intelligence methodologies employed.

References

1. Friedland, P.E., *Knowledge-Based Experiment Design In Molecular Genetics,* PhD dissertation, Stanford University, October 1979.

2. Friedland, P.E., "Knowledge-Based Experiment Design in Moleculare Genetics," *IJCAI-79,* The International Joint Conference on Artificial Intelligence, 1979, pp. 285-287.

3. Stefik, M.J., *Planning with Constraints,* PhD dissertation, Stanford University, January 1980.

4. Stefik, M.J., "Planning and Meta-Planning," HPP-memo HPP-80-13, Stanford University Heuristic Programming Project, 1980.

5. Smith, R.G., Friedland, P.E., "Unit Package User's Guide," HPP-memo HPP-80-28, Stanford University Heuristic Programming Project, 1980.

6. Scott, A.C., Clancey, W.J., Davis, R., Shortliffe, E.H., "Explanation Capabilities of Production-Based Consultation System," *American Journal of Computational Linguistics,* 1979, .