Profile of the MOLGEN System

By Zack Stoecker−Sylvia

GENERAL

Domain:

Biology (Genetics)

Main General Function:

Planning of laboratory experiments

System Name:

MOLGEN

Dates:

1979−1980

Researchers:

Mark Stefik (head developer of the most recent version)

Peter Friedland (head developer of an earlier version)

Douglas Brutlag, Jerry Feitelson, and Lawrence Kedes (members of the MOLGEN project)

Location:

Stanford University, Stanford CA

Language:

Unknown. Language of implementation is not stated in descriptions of the system.

Machine:

Unknown.

Brief Summary:

MOLGEN implements a hierarchical planning method with constraint propagation to plan laboratory experiments in molecular genetics. MOLGEN uses three separate control layers to separate real world laboratory objects and operators, planning methods, and control strategies. The higher layers dictate the actions of the lower layers. Additionally procedures and objects are abstracted into a frame hierarchy. MOLGEN begins with abstract plans and gradually refines them, using a least−commitment strategy whenever possible and using heuristics methods when least−commitment is not possible.

Related Systems:

a) GPS, ABSTRIPS, TEIRESIAS, HEARSAY−II, SU−X, SU−P

b) None

c) MOLGEN (Friedland, 1979)

CATEGORY TWO

Characterization of Givens:

Given: The goal of the experiment

Built−in: The procedures and objects available in a genetics laboratory

Characterization of Output:

The system outputs the steps taken to design the experiment, the constraints introduced at each step, and the results of each step.

Characterization of Data:

The data built into the system is reliable (some information is variable, but that is taken into account when performing procedures). The data is not complete. Molecular genetics is a constantly expanding field. Steps to design new objects with specific properties may be performed during the planning of an experiment.

Generic Tasks:

Hierarchical planning, refinement of abstract operators and objects, and constraint posting

Theoretical Commitment:

MOLGEN first combined hierarchical planning with constraint posting and a layered control structure. It uses the least−commitment strategy to avoid backtracking. Stefik claims that a least−commitment approach is superior to using agendas for planning. MOLGEN’s planning strategy can be used in nearly any planning domain with minimal modification.

Reality:

Humans often consider abstract plans before refining those plans into more specific plans. Additionally, the idea of constraint posting is sometimes loosely used in human planning.

CATEGORY THREE

Completeness:

The system has been completely implementing (excepting for advances in molecular genetics).

Use:

The system was not designed for actual laboratory use and was not used outside of testing.

Performance:

Stefik tested the system by creating a plan to develop bacteria that could produce rat−insulin. The system was able to identify 4 different possible experiments. The validity of constraint−posting to reduce search space was measured by observing the number of plans remaining in the search space as new constraints were added. Constraints narrowed the search space quickly, but MOLGEN's performance was not compared to other systems.

CATEGORY FOUR

Phases:

The system is organized into least−commitment planning phases and heuristic guessing phases. These phases are cycled through until a satisfactory plan is created.

Subfunctions:

The system performs selection when refining objects and operators and occasionally synthesizes new objects to meet requirements.

Use of Simulation Analysis:

MOLGEN constantly simulates the effects of plan steps to verify that those steps will meet their requirements.

System/Control Implementation Architecture:

MOLGEN uses three layers to separate more abstract control operations. The highest level cycles through control strategies, the middle level performs actual planning, and the lowest level contains the operations and objects available while recording the steps which MOLGEN has selected.

CATEGORY FIVE

Characterization of Structure Knowledge:

Knowledge about objects is divided into the three control layers depending on the purpose of those objects. Objects in the laboratory layer are arranged into an abstraction hierarchy.

Characterization of Process Knowledge:

Process knowledge is also divided into the three control layers depending on function. Operators require some number of objects to operate on and their expected result. Laboratory operators also contain information allowing the simulation of the operator's effects.

Deep or Surface:

Most of the planning knowledge will apply to multiple domains and therefore is surface knowledge (the mid−level planning knowledge is marginally tied to the laboratory knowledge). The laboratory knowledge is specific to the molecular genetics domain and is therefore deep knowledge.

CATEGORY SIX

Search Space:

MOLGEN searches through a space containing possible plans to accomplish its goals. States in the search space are not represented explicitly, but abstract plans can be seen to represented areas of the search space. More definite plans represent smaller search spaces. Each state is a refinement of a previous state.

Space Traversal:

The search space is traversed by refining abstract plans, operators, and objects.

Search Control Strategy:

The system uses a generic planning strategy to solve the problem. The system’s expert knowledge is primarily expressed by its planning strategies.

Standard Search Strategies:

The system uses Means Ends Analysis to narrow the search space.

Search Control Characterization:

The system uses a knowledge−based search control strategy.

Subproblems:

The system uses least−commitment whenever possible, which means that it is always on the right track. Subplans are nearly always dependent on other plans.

Search Control Representation:

Search control knowledge is explicitly stated in the topmost layer. Search control is represented as operations available to the interpreter.

Search Control Strength:

The search control knowledge is not based on any domain knowledge and is therefore weak.

CATEGORY SEVEN

Failure Method:

When a heuristic guess is determined to be incorrect because it introduces constraints which cannot be satisfied the program breaks from normal operation to backtrack to the point when the guess was made. The program attempts to determine which guess is responsible for the unsatisfiable constraint using domain independent knowledge. The program backtracks to the point when the guess was made.

Uncertainty:

Uncertainty in objects and operators is represented by abstract operators. These must all be made definite before MOLGEN is finished. Some operators can provide variable results, but this is expected in the domain and accounted for with further planning.

Management of Uncertainty:

The system manages uncertainty with further planning. Completed plans do not contain measures of uncertainty.

Management of Time:

Time is only represented by order of plans. Data is not time dependent.

CATEGORY EIGHT

Knowledge Representation Method:

Knowledge is represented by procedures and frame−based objects.

Knowledge Representation Generality:

Knowledge is programmed into the system. An Expert System building tool has been created to add new procedures and objects to the system.

Knowledge Structuring:

Laboratory level knowledge is represented in abstraction hierarchies. These hierarchies correspond to the problem domain. Design (middle) level knowledge is also grouped by operator purpose. These groupings correspond to the problem solving strategy.

CATEGORY NINE

Alternative Representations:

Laboratory level objects are based off of generic instances of their object type, but this is provided to compare those objects to a general object rather than to provide alternate solutions. Objects are always represented in the same manner.

Alternate Solution Methods:

The system does not use alternate solution methods. Plans may be refined in different orders, but the methods used are the same.

Optimization:

The system does not necessarily produce the best answer, simply a correct one.

Multiple Results:

The system does not provide multiple results.

CATEGORY TEN

Interaction:

The system does not interact with the user after the input of the goal.

Data Collection:

All data must be supplied before the program executes.

Data Format:

Unknown.

Acquisition:

The system does not acquire additional information from the user.

Learning:

The system does not learn from its own performance.

Explanation:

The system explains its results by providing a trace of the program's execution.

CATEGORY ELEVEN

Strengths:

The system combines many strong planning strategies into one system. Least−commitment allows the system to avoid backtracking and data abstraction allows for partial refinement of plans.

Weaknesses:

The system does not work well in an actual laboratory situation because it is unable to interact with experts during execution. In a standard laboratory, unexpected results during an experiment could be interpreted by the expert and followed up, whereas MOLGEN only creates complete plans.

Other:

This system has the potential to work well in many planning domains, however because the design layer directly interacts with domain dependent laboratory layer, to apply it to a new domain requires some reworking of the design layer as well as the domain dependent objects and operators.