Lilith: From Childhood to Adolescence

Guido Sello

Dipartimento di Chimica Organica e Industriale, Universitá degli Studi di Milano, via Venezian 21, Milan, Italy

Received June 28, 1993®

Computer assisted organic synthesis is the best to date effort for the understanding and the optimization of the synthesis planning activity which represents one of the most exciting and funny tasks for a chemist. Several approaches have been developed in past years,[1] each one contributing ideas and theoretical achievements which resulted in several beautiful works. Our contribution to the field, named Lilith, is still one of the youngest, but nevertheless it has come time for her to grow a little more, leaving childhood for adolescence. In her young age Lilith acquired some important attributes which qualified her behavior (a strategy, a reactivity analysis, a first level of evaluation capability); now she is going to complete her power improving her possibility of controlling operations and results. In this perspective a completely new and powerful nucleus of the program is being studied; it mainly consists of a true expert system whose tasks are (1) the control of the work flow, (2) the analysis and the evaluation of intermediate and final results, and (3) the optimization of the application of the IAIA (initial approximation increasing accuracy) principle.[2] Here its first version is reported with particular attention to the analysis of the up-to-data achievements in the field of the characterization of its theoretical skeleton.

INTRODUCTION

When in 1986 we started the development of a new approach to computer assisted organic synthesis, we first discussed our basic philosphy in order to get, as the very last product, a reliable, general, and useful program. Three guidelines were born out, each one concerned with a special requirement of the synthetic analysis: (1) the approach should have been highly conditioned by the strategy; (2) the chemical reactivity management, everyday changing, should not have been a limitation; (3) the analysis should have been structured in such a way as to be in the position of correcting its own errors. Those three guidelines were satisfying the corresponding requirements of logical bias, generality, and reliability, therefore giving a useful product.

Since then, much effort was spent by a group of people, and we can proudly affirm that the first and the second objectives have been realized (at least concerning the 80% of the total solution). However, the last point presented some unpredictable difficulties. The greatest part of them concerns the need to evince the relative importance of errors in real practical synthesis; the remaining ones are, on the contrary, limits that we found in logical programming, i.e., in translating the logical mechanims of the approach into computer routines.

It is our intention to show part of the solution to this last problem in the present work.

BACKGROUND

Our program (Lilith) for CAOS is structured into three logical pieces: (1) strategy; (2) reactivity; (3) control mechanism.

The main strategy problem is to grasp the best synthesis plan inside that enormous box which is the solution space; our proposal is to focus on that part of the box which can assure the presence *also* of the best solution; i.e., we use the principles of simplification and convergence.[2,3]

As far as reactivity is concerned, we point to a different idea; the experimental and theoretical knowledge about reactivity constantly increases, and the ability of laboratory chemists in managing reactions becomes more and more impressive. Therefore it is absurd to build a system that is representative of the up-to-date reactions. On the contrary, the fundamental mechanisms of reactivity are fairly constant even if they are not completely understood yet. Our solution is an open model of reactivity representation which only uses the molecular characteristics, unbiased by the experimental data.[4]

We should emphasize the diversity of our solutions to the two problems: the first, concerning strategy, could be too rigid; the second, concerning reactivity, could be too general. Here is the point where the control mechanism must operate and where the program can change from naive to professional. However, here is also the point where difficulties come in.

Our first idea concerning the control mechanism was the application of a well-known principle of logical analysis which we called the IAIA principle (initial approximation increasing accuracy);[1] its task is very clear: at the beginning of the analysis the introduced approximations can be large; toward the end the program must be more and more careful, clear but not easy, and, overall, not error-free. Nobody can assure that the excluded solutions cannot become important when we have more data available. Our second idea was a direct consequence of that: at any time the need will present, the program should be able to loop back and either to recover the discarded solutions or even to change the analysis path.

This is the story which took us to the present work; now we will begin the description of the next installment.[5]

THE CONTROL SYSTEM

The difficulties presented by the analysis of the control system have already been pointed out. One of them, the programming limit present in Fortran, had found an easy solution as soon as we had discovered a shell for writing expert systems.[6] This product has relieved us of the problem, and we instantly found ourselves in the position of ignoring any programming problem.[7]

The second subset of difficulties strictly concerns the capability of choosing what can be defined erroneous in

® Abstract published in *Advance ACS Abstracts,* January 15, 1994.

0095-2338/94/1634-0120$04.50/0

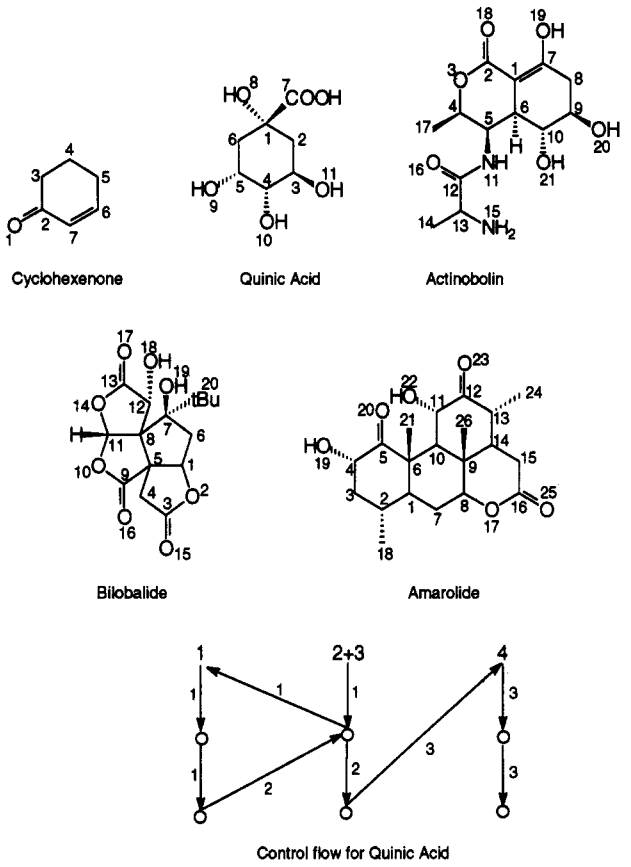

Figure 1. Structure used as a test set for ES. The logical flow of the control for a run on quinic acid is shown (node numbers refer to solutions; edge numbers refer to the rank of the solutions).

synthesis planning. For example if we consider the strategy, it is clear that we need a measure of the level of satisfaction of the two guiding principles: simplification and convergence. We choose a measuring system, and we could order solutions by strategical value and discard everything below a chosen threshold. This system works but it is not clever enough; in fact, what does happen if, at the end of the analysis, we have only two solution left? Or more than fifty? We next look at the answer proposed by our approach.

The control system should be able always to process the solution which is presently the best. This assumption ideally results in the growing of the best branch of the solution tree.

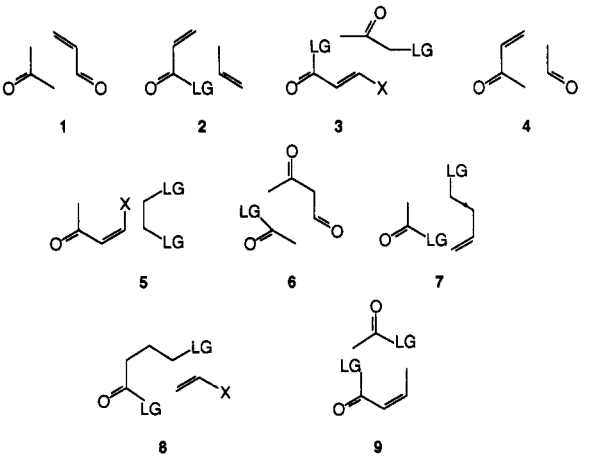

Figure 2. Solutions proposed by Lilith for cyclohexenone. The standard Robinson condensations (4 + 2, 3 + 3) are solutions 1 and 4; their ranks are respectively 4 and 7 (first run) and 5 and 4 (second run). These last ones reproduce also the bond breaking sequence of the true Robinson condensations.

The similarity of our choice with the result of a system controlled by a hypothetical reasoning strategy is evident; therefore the choice of this reasoning strategy is its natural consequence. As already mentioned, we have one part of the system (Lilith) which makes calculations and the other (ES) which (a) controls the value and the rank of each solution and (b) distributes activities. One of the possible approaches could be the full integration of the two parts, but two reasons are against it: the first concerns the level of development of Lilith which would require too many changes; the second concerns calculation speed which can be highly biased by a too heavy splitting of the work. Therefore we decided to structure the activities as follows.

The first phase is a complete run of Lilith from strategy[3a] to interference evaluation[4d] with simultaneous creation of the database containing all the useful information for ES. At this point the work structure changes toward a more integrated phase. ES analyzes the solution set for what concerns the strategy, chooses the best solution, and continues the control (only on the best solution) on the polarity. If ES finds a problem (e.g., contrasting polarities), the activity is switched to Lilith which takes a first-order correction which, in the case of multiple strategic bond break, consists of the reordering of the bond break sequence (with sequent recalculation of the

Table 1. First Run Solutions for Cyclohexenone[a]

soln no.	cmp	maxd	At1	At2	Pol1	Pol2	Sgi1	Sgi2	level	str val	pol val	Sgi val
1	4	1.000	6	7	10.78	−9.84	0.00	81.00	4	8.00	0.087	82
1	4	0.000	4	3	8.00	−8.69	0.00	1.00				
2	1	1.000	5	4	−7.43	7.00	16.00	0.00	1	2.00	0.133	32
2	1	0.000	7	2	−6.33	7.78	16.00	0.00				
3	4	1.000	6	5	−2.10	2.00	0.00	0.00	9	8.00	5.386	5
3	4	0.000	2	3	7.14	−3.16	4.00	1.00				
4	4	1.000	6	7	10.78	−9.84	0.00	81.00	7	8.00	5.046	81
4	4	0.000	4	5	−0.17	0.17	0.00	0.00				
5	4	1.000	6	5	−2.10	2.00	0.00	0.00	6	8.00	5.041	4
5	4	0.000	4	3	8.00	−8.69	3.00	1.00				
6	4	1.000	7	6	−9.84	10.78	81.00	0.00	5	8.00	0.459	146
6	4	0.000	3	2	−3.16	7.61	1.00	64.00				
7	1	1.000	7	2	−6.33	7.78	16.00	0.00	2	2.00	0.144	19
7	1	0.000	3	4	−8.69	8.00	0.00	3.00				
8	4	1.000	5	6	2.00	−2.10	0.00	0.00	8	8.00	5.103	1
8	4	0.000	7	2	−6.33	7.78	1.00	0.00				
9	1	1.000	2	3	7.14	−3.16	4.00	1.00	3	2.00	0.416	9
9	1	0.000	4	5	7.00	−7.43	3.00	1.00				

[a] Soln no. = solution number; cmp = molecular complexity; maxd = globularity index; At*n* = atom number; Pol*n* = effective polarity; Sgi*n* = interference level; level = solution rank; str val = strategy value; pol val = polarity value; Sgi val = interference value.

Table 2. Second Run Solutions for Cyclohexenone[a]

soln no.	cmp.	maxd	At1	At2	Pol1	Pol2	Sgi1	Sgi2	level	str val	pol val	Sgi val
1	4	1.000	4	3	8.00	−8.69	0.00	16.00	5	8.00	0.087	20
1	4	0.000	6	7	10.78	−9.84	3.00	1.00				
2	1	1.000	7	2	−6.33	7.78	16.00	0.00	1	2.00	0.133	35
2	1	0.000	5	4	−7.43	7.00	16.00	3.00				
3	4	1.000	2	3	7.14	−3.16	4.00	1.00	9	8.00	5.386	8
3	4	0.000	6	5	−2.10	2.00	0.00	3.00				
4	4	1.000	4	5	7.00	−7.43	0.00	16.00	4	8.00	0.075	20
4	4	0.000	6	7	10.78	−9.84	3.00	1.00				
5	4	1.000	4	3	8.00	−8.69	0.00	16.00	7	8.00	5.041	19
5	4	0.000	6	5	−2.10	2.00	0.00	3.00				
6	4	1.000	3	2	−3.16	7.14	1.00	4.00	6	8.00	0.432	8
6	4	0.000	7	6	−9.84	10.78	0.00	3.00				
7	1	1.000	3	4	−8.69	8.00	16.00	0.00	2	2.00	0.144	32
7	1	0.000	7	2	−6.33	7.78	16.00	0.00				
8	4	1.000	7	2	−6.33	7.78	16.00	0.00	8	8.00	5.103	20
8	4	0.000	5	6	2.00	−2.10	3.00	1.00				
9	1	1.000	4	5	7.00	−7.43	0.00	16.00	3	2.00	0.416	20
9	1	0.000	2	3	7.14	−3.16	4.00	0.00				

[a] Soln no. = solution number; cmp = molecular complexity; maxd = globularity index; At*n* = atom number; Pol*n* = effective polarity; Sgi*n* = interference level; level = solution rank; str val = strategy value; pol val = polarity value; Sgi val = interference value.

Table 3. First Run Solutions for Quinic Acid[a]

soln no.	cmp	maxd	At1	At2	Pol1	Pol2	Sgi1	Sgi2	level	str val	pol val	Sgi val
1	2	0.929	6	5	−1.40	8.93	81.00	64.00	1	4.216	1.458	485
1	2	0.125	2	3	−1.40	8.93	16.00	324.0				
2	2	0.929	6	5	−1.40	8.93	81.00	64.00	2	3.716	5.729	1170
2	2	0.000	2	1	−1.80	1.83	1.00	1024.0				
3	2	0.929	2	3	−1.40	8.93	81.00	64.00	2	3.716	5.729	1170
3	2	0.000	6	1	−1.80	1.83	1.00	1024.0				
4	6	1.000	6	1	−1.80	1.83	1.00	3.00	3	12.00	5.566	4
4	6	0.000	2	1	2.00	−7.22	0.00	0.00				

[a] Soln no. = solution number; cmp = molecular complexity; maxd = globularity index; At*n* = atom number; Pol*n* = effective polarity; Sgi*n* = interference level; level = solution rank; str val = strategy value; Pol val = polarity value; Sgi val = interference value.

Table 4. Second Run Solutions for Quinic Acid[a]

soln no.	cmp	maxd	At1	At2	Pol1	Pol2	Sgi1	Sgi2	level	str val	Pol val	Sgi val
1	2	0.929	2	3	−1.40	8.93	81.00	64.00	1	4.216	1.458	485
1	2	0.125	6	5	−1.40	8.93	16.00	324.0				
2	2	1.000	2	1	−1.80	1.83	16.00	64.00	2	4.00	5.729	1105
2	2	0.000	6	5	−1.40	8.93	1.00	1024.0				
3	2	1.000	6	1	−1.80	1.83	16.00	64.00	2	4.00	5.729	1105
3	2	0.000	2	3	−1.40	8.93	1.00	1024.0				
4	6	1.000	2	1	−1.80	1.83	1.00	3.00	3	12.00	5.566	4
4	6	0.000	6	1	2.00	−7.22	0.00	0.00				

[a] Soln no. = solution number; cmp = molecular complexity; maxd = globularity index; At*n* = atom number; Pol*n* = effective polarity; Sgi*n* = interference level; level = solution rank; str val = strategy value; Pol val = polarity value; Sgi val = interference value.

solution). If the correction does not solve the problem, the analysis goes back, takes a new solution, and repeats. On the other hand, the analysis continues considering similar group interferencers (SGIs) and repeating a completely similar set of operations. This sequence is repeated on all the solutions of the first level. The result is the ordering of the solutions in accord with their value and the assignment of a status of acceptance.

At the second level there is a preliminary new run of Lilith with a new setting of the strategy variables therefore obtaining a new solution set. As a consequence there will be also a new run of ES, giving a result similar to the previous one. For the sake of clarity we describe, in the following, the structure of the expert system.

An expert system is made of two principal parts: the knowledge base (which resides in the objects) and the rules (which operate on the objects and represent the inference engine of the system). Both of them are introduced and discussed.

Objects. An object is a data structure used to define all data in an application. Each object consists of one or more attributes, which are the definition of individual data items. The object definition maps to one or more occurrences of the object, each of which contains the data about one instance of the real world entity it represents. An object can have different characteristics and properties.

(a) Multiple-Occurring and Singleton Objects. Typically the objects have several occurrences; for example, a molecule object would have one occurrence for each existing molecule. On the contrary, particular objects can have only one occurrence, and they are called singleton; for example, internal storage variable objects would be singletons.

(b) Temporary and Permanent Objects. Temporary objects are created during a consultation and deleted at the start of the next consultation; permanent objects, once created, must be explicitly deleted, or otherwise they remain in the knowledge base.

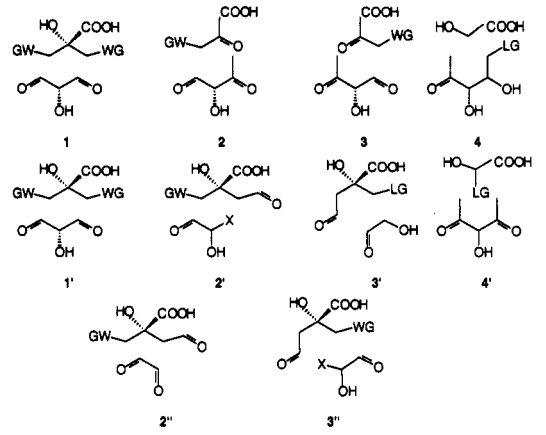

Figure 3. Solutions proposed by Lilith for quinic acid. In the first row the solutions of the first strategy setting are reported (they do not change in the two runs); in the second row the solutions of the second strategy setting are reported; in the third row the solutions of the second row which change in the second run are reported.

(c) Methods. It is possible to associate actions with an object through methods. They are actions which are initiated under particular situations, i.e., when a modification of the object occurs (when the object is created, deleted, or needed, or when an attribute value is assigned or needed).

In ES we can identify three types of objects: (1) objects representing the knowledge data; (2) objects used to manage variables; (3) objects which connect ES to Lilith. The most important objects are in the first group. They are four: MOLECULE, SOLUTION, VALUES, LEVEL; the last one is a singleton.

MOLECULE. It is the internal representation of a molecule; therefore it must contain information gathered from Lilith and needed by ES. Its attributes are STRAT ATOM1 and STRAT ATOM2, which correspond to the atoms spanning each strategic bonds; POLAR1 and POLAR2, which correspond to the effective polarity of each strategic atom; INTERFER1 and INTERFER2, which correspond to the similar group interference of each strategic atom; NUMBER, which uniquely identifies the Lilith solution; COMPL, which is the measure of the level of simplification and convergence of the solution; MAXDM, which corresponds to the level of globularity of each fragment of each solution.

There is one occurrence of MOLECULE for each strategic bond, i.e., more than one occurrence for each Lilith solution, but the strategic bonds are grouped together by their MOLECULE.NUMBER.

SOLUTION. It is the solution created by ES in correspondence to each solution proposed by Lilith; it contains the summary of the results elaborated in each ES session. Its attributes are NUMBER, which uniquely identifies the ES solution and connects SOLUTION to MOLECULE occurrence; DISCARDED, which respresents the solution status for the present cycle of the analysis; MAXD, which is the total strategy value of the Lilith solution (calculated using MOLECULE.MAXDM); COMPLEXITY, which directly corresponds to MOLECULE.COMPL; LVL, which is the analysis level reached by the solution; ACCEPTED, which represents the solution status at the end of one complete ES run; and STEP, which is the level reached by the solution at the end of each ES cycle. There is one occurrence of SOLUTION for each Lilith solution.

VALUES. It contains the data concerning the calculations made inside ES (those calculations made inside Lilith are contained in MOLECULE); its role is fundamental in choosing which node to process next. Its attributes are CURR VALUE, which is the current solution value; STR VALUE, which stores the strategy value only; POL VALUE, which stores both the strategy and the polarity values; SGI VALUE, which stores the interference value; VAL NO, which uniquely represents the ES solution and connects it to SOLUTION.NUMBER; and FLAG VAL, which is a flag used to save the calculation time.

LEVEL. It is a singleton object which contains data concerning the status of the consultation. Its attributes are PHASE NAME and PHASE LEVEL, which respectively identify the analysis phase with a conventional name and a number; LEVEL NO, which is the level of the current cycle; and STEP, which is the level inside the current cycle.

Table 5. First Run Solutions for Quinic Acid (Second Strategy)[a]

soln no.	cmp	maxd	At1	At2	Pol1	Pol2	Sgi1	Sgi2	level	str val	Pol val	Sgi val
1	6	0.929	5	6	8.93	−1.40	64.00	81.00	2	12.648	1.458	485
1	6	0.125	3	2	8.93	−1.40	324.0	16.00				
2	2	0.929	5	6	8.93	−1.40	64.00	81.00	1	4.116	0.765	470
2	2	0.100	3	4	7.52	−7.00	324.0	1.00				
3	6	0.929	5	4	2.52	2.54	64.00	64.00	3	12.348	5.529	129
3	6	0.100	3	2	−6.50	2.00	1.00	0.00				
4	6	1.000	1	6	1.83	−1.80	3.00	1.00	4	12.00	10.00	5
4	6	0.000	1	2	−3.65	−1.80	0.00	1.00				

[a] Soln no. = solution number; cmp = molecular complexity; maxd = globularity index; At*n* = atom number; Pol*n* = effective polarity; Sgi*n* = interference level; level = solution rank; str val = strategy value; pol val = polarity value; Sgi val = interference value.

Table 6. Second Run Solutions for Quinic Acid (Second Strategy)[a]

soln no.	cmp	maxd	At1	At2	Pol1	Pol2	Sgi1	Sgi2	level	str val	pol val	Sgi val
1	6	0.929	3	2	8.93	−1.40	64.00	81.00	3	12.648	1.458	485
1	6	0.125	5	6	8.93	−1.40	324.0	16.00				
2	2	0.929	3	4	2.52	2.54	64.00	64.00	1	4.116	5.729	1233
2	2	0.100	5	6	8.93	−1.40	1024.0	81.00				
3	6	0.929	3	2	8.93	−1.40	64.00	81.00	2	12.348	0.765	470
3	6	0.100	5	4	7.52	−7.00	324.0	1.00				
4	6	1.000	1	2	1.83	−1.80	3.00	1.00	4	12.00	10.00	5
4	6	0.000	1	6	−3.65	−1.80	0.00	1.00				

[a] Soln no. = solution number; cmp = molecular complexity; maxd = globularity index; At*n* = atom number; Pol*n* = effective polarity; Sgi*n* = interference level; level = solution rank; str val = strategy value; pol val = polarity value; Sgi val = interference value.

Table 7. First Run Solutions for Bilobalide[a]

soln no.	cmp	maxd	At1	At2	Pol1	Pol2	Sgi1	Sgi2	level	str val	Pol val	Sgi val
1	12	0.625	7	8	8.64	−1.23	1024	16.00	2	54.828	6.414	1122
1	12	0.625	5	4	−9.88	2.00	81.00	0.00				
1	12	0.273	5	1	−4.86	2.00	1.00	0.00				
2	8	0.450	10	11	−16.97	17.45	16.00	0.00	1	25.608	0.758	375
2	8	0.525	6	7	−1.64	9.06	16.00	324.0				
2	8	0.092	8	5	8.00	−8.85	3.00	16.00				
3	8	0.625	7	8	0.00	0.00	0.00	0.00				
3	8	0.650	9	5	0.00	0.00	0.00	0.00				
3	8	0.027	8	5	0.00	0.00	0.00	0.00				
4	12	0.525	7	6	0.00	0.00	0.00	0.00				
4	12	0.525	9	10	0.00	0.00	0.00	0.00				
4	12	0.092	8	5	0.00	0.00	0.00	0.00				
5	8	0.475	8	5	0.00	0.00	0.00	0.00				
5	8	0.575	10	11	0.00	0.00	0.00	0.00				
5	8	0.053	6	1	0.00	0.00	0.00	0.00				
6	20	0.625	8	7	0.00	0.00	0.00	0.00				
6	20	0.226	6	1	0.00	0.00	0.00	0.00				
7	12	0.625	7	8	0.00	0.00	0.00	0.00				
7	12	0.650	9	10	0.00	0.00	0.00	0.00				
7	12	0.011	8	5	0.00	0.00	0.00	0.00				
8	8	0.525	7	6	0.00	0.00	0.00	0.00				
8	8	0.525	9	5	0.00	0.00	0.00	0.00				
8	8	0.053	8	5	0.00	0.00	0.00	0.00				
9	12	0.475	8	5	0.00	0.00	0.00	0.00				
9	12	0.575	10	9	0.00	0.00	0.00	0.00				
9	12	0.053	6	1	0.00	0.00	0.00	0.00				
10	22	0.625	8	7	0.00	0.00	0.00	0.00				
10	22	0.233	6	7	0.00	0.00	0.00	0.00				
11	18	0.525	5	4	0.00	0.00	0.00	0.00				
11	18	0.575	5	1	0.00	0.00	0.00	0.00				
11	18	0.414	7	6	0.00	0.00	0.00	0.00				

[a] Soln no. = solution number; cmp = molecular complexity; maxd = globularity index; At*n* = atom number; Pol*n* = effective polarity; Sgi*n* = interference level; level = solution rank; str val = strategy value; pol val = polarity value; Sgi val = interference value.

Table 8. Second Run Solutions for Bilobalide[a]

soln no.	cmp	maxd	At1	At2	Pol1	Pol2	Sgi1	Sgi2	level	str val	Pol val	Sgi val
1	12	0.525	5	4	−9.74	2.00	16.00	0.00	2	49.428	6.410	1057
1	12	0.575	5	1	−4.86	2.00	16.00	0.00				
1	12	0.273	7	8	8.64	−1.23	1024.0	1.00				
2	8	0.450	10	11	−16.97	17.45	16.00	0.00	1	26.808	0.758	375
2	8	0.575	8	5	8.00	−8.85	3.00	16.00				
2	8	0.092	6	7	−1.64	9.06	16.00	324.0				

[a] Soln no. = solution number; cmp = molecular complexity; maxd = globularity index; At*n* = atom number; Pol*n* = effective polarity; Sgi*n* = interference level; level = solution rank; str val = strategy value; pol val = polarity value; Sgi val = interference value.

Table 9. Third Run Solutions for Bilobalide[a]

soln no.	cmp	maxd	At1	At2	Pol1	Pol2	Sgi1	Sgi2	level	str val	pol val	Sgi val
1	12	0.625	7	8	8.64	−1.23	1024	16.00	2	60.228	12.834	1123
1	12	0.775	5	1	−4.86	2.00	81.00	0.00				
1	12	0.273	5	4	−9.74	−7.33	1.00	1.00				
2	8	0.525	6	7	−1.64	9.06	81.00	1024.0	1	30.408	0.758	1361
2	8	0.650	8	5	8.00	−8.85	0.00	256.0				
2	8	0.092	10	11	−16.97	17.45	0.00	0.00				

[a] Soln no. = solution number; cmp = molecular complexity; maxd = globularity index; At*n* = atom number; Pol*n* = effective polarity; Sgi*n* = interference level; level = solution rank; str val = strategy value; pol val = polarity value; Sgi val = interference value.

We introduce only one of the objects belonging to the second group: THRSH, which contains temporary information concerning the active thresholds. As there are presently three distinct phases of the synthetic analysis (strategic bond location, polarity determination, and interference evaluation, being the fourth phase (reaction condition optimization) not done yet), we define three attributes of THRSH: TRSH1, which contains the strategy threshold; TRSH2, which contains the polarity plus the strategy threshold; TRSH3, which contains the interference threshold. The object is temporary and singleton and has a "when needed" condition; therefore it is calculated only when it is explicitly requested by ES. The threshold values change during the analysis, and they are equal to the corresponding solution values of the presently best solution.

Some brief comments follow on the objects of the third group. They are of two kinds: two of them are simply sequential files containing the data which ES and Lilith exchange; the third is an UDO (user defined object) whose task is to transfer the activity from ES to Lilith or vice versa. (This object is presently inactive.)

No other knowledge representation seems necessary; however, the system can be easily updated when needed.

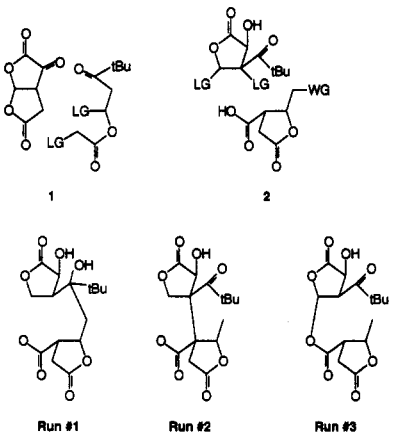

Figure 4. Some of the solutions proposed by Lilith for bliobalide. Solutions 1 and 2 of first run are shown in the first row in order to emphasize the difference in globularity. The last bond that breaks for solution 2 in all the runs is shown in the second row; the presence of the tBu group far from the last broken bond causes a greater weight of the globularity factor.

Rules. A rule is the encoding of a discrete piece of knowledge about the data being manipulated. The left-hand side of the rule specifies a "pattern", and the right-hand side specifies one or more "actions". The pattern describes data values and conditions that must be present in one or more object occurrences in order for the action(s) to be initiated; the actions are applied to each occurrence that matches the pattern. Rules are used in "packets" and "methods"; a packet is a collection of related rules which can execute upon entry to or exit from the packet; and methods have been already introduced in the object description.

Packets can include the following elements: priority function, a pattern which sets the relative priority of the packet when the packet is schedule for execution; agenda type, a keyword which defines the relative priority of packet families (they are three, standard, demon, or inactive; the first can be active or inactive, the second is always active, the third is always inactive); agenda condition, patterns that describe the conditions under which the packet is to be initiated; exit condition, a pattern that, if matched, will cause the exit from the packet before completion; entry and exit actions, any action to take at entry or exit time; rules. Packets ideally should take care of one precise task.

Excluding I/O packets, ES contains 18 packets (35 rules overall). Three of them are special packets which practically transfer data to and from Lilith; they are called "requests" and, even if they are important in order to assure perfect correspondence and timing between ES and Lilith, they do not give any special contribution to ES.

The remaining 15 packets can be further grouped:

(1) MAIN is the main packet which opens and closes each cycle and each run.

(2) READLIL, INSERT DATA, INIT TREE, and COMPLETE MOLECULE initalize the system.

(3) ACCESS LILITH and WRITE CONTR communicate with Lilith.

(4) EXPAND TREE, EVALUATE NODE, ASSIGN THRSH, DO VALUES, LAST CHECK, VERIFY THRSH, CURR TO ZERO, and ADJUST VALUE are the heart of the system.

MAIN. This packet controls the fundamental flow of the process; it contains only one rule which starts a new cycle of operations until the final result is obtained. Two other tasks (reading the input data and writing the output data) are automatically done as, respectively, entry and exit actions.

READLIL, INSERT DATA, INIT TREE, and COMPLETE MOLECULE. These four packets take care of the transformation of the Lilith data into ES form. They contain six rules.

ACCESS LILITH should activate Lilith when ES needs more data from it, but it is presently inactive.

WRITE CONTR. Its task is to write data on a file which is then read by Lilith and therefore controls its execution; moreover the rule CHECKING gives the last accepting quota to each solution by checking the level of correctness of polarities and interferences which depends on the analysis phase.

EXPAND TREE. It is the first packet directly involved in the ES manipulation of data; moreover it represents the heart of ES because it controls the growing of the decision tree by using both its rules and calling other packets. The first two rules generate the solution object occurrences, making the connections with the input and calling the DO VALUES packet which performs a similar task for the corresponding value object occurrences. The third rule calculates the solution values according to the phase level and step (calling the EVALUATE NODE packet). Then there are two rules, GROWING and STOPPING, which respectively grows a branch or stops it. The decision on which a branch should grow is taken according to the hypothetical reasoning strategy; therefore a provisional estimation of the "best at the moment" solution is done. (Many solutions can be selected at the same time.) The remaining rule of the packet (LAST TIME) gives temporary acceptance to the solutions which survived the ongoing cycle. Eventually some exit actions take care of the updating of the internal control variables.

EVALUATE NODE. This packet is strictly connected to the previous one, making the calculations needed in order to proceed in the analysis. It contains seven rules which fire only at the right moment and only for the active solutions (not discarded).

Table 10. First Run Solutions for Actinobolin[a]

soln no.	cmp	maxd	At1	At2	Pol1	Pol2	Sgi1	Sgi2	level	str val	pol val	Sgi val
1	4	0.633	3	2	−11.51	10.46	256.0	64.00	2	7.568	5.048	644
1	4	0.313	5	6	4.31	−4.67	324.0	0.00				
2	0	0.633	3	4	−16.12	3.30	256.0	0.00	1	1.892	5.660	580
2	0	0.313	5	6	4.31	−4.67	324.0	0.00				
3	24	0.391	5	11	0.00	0.00	0.00	0.00				
4	20	0.800	5	6	0.00	0.00	0.00	0.00				
4	20	0.250	5	4	0.00	0.00	0.00	0.00				
5	18	0.733	5	4	0.00	0.00	0.00	0.00				
5	18	0.269	3	2	0.00	0.00	0.00	0.00				
6	12	0.733	5	4	0.00	0.00	0.00	0.00				
6	12	0.269	3	4	0.00	0.00	0.00	0.00				

[a] Soln no. = solution number; cmp = molecular complexity; maxd = globularity index; At*n* = atom number; Pol*n* = effective polarity; Sgi*n* = interference level; level = solution rank; str val = strategy value; pol val = polarity value; Sgi val = interference value.

Table 11. Second Run Solutions for Actinobolin[a]

soln no.	cmp	maxd	At1	At2	Pol1	Pol2	Sgi1	Sgi2	level	str val	pol val	Sgi val
1	4	0.800	5	6	4.31	−4.67	324.0	0.00	1	8.904	5.048	648
1	4	0.313	3	2	−11.51	10.46	0.00	324.0				
2	0	0.800	5	6	4.31	−4.67	324.0	0.00	2	2.226	13.981	421
2	0	0.313	3	4	−15.52	−12.41	81.00	16.00				

[a] Soln no. = solution number; cmp = molecular complexity; maxd = globularity index; At*n* = atom number; Pol*n* = effective polarity; Sgi*n* = interference level; level = solution rank; str val = strategy value; pol val = polarity value; Sgi val = interference value.

Table 12. First Run Solutions for Amarolide[a]

soln no.	cmp	maxd	At1	At2	Pol1	Pol2	Sgi1	Sgi2	level	str val	pol val	Sgi val
1	8	0.522	17	16	−11.51	7.96	16.00	64.00	4	27.21	5.237	598
1	8	0.565	10	6	8.00	−8.93	3.00	256.0				
1	8	0.043	8	9	2.00	−1.17	3.00	256.0				
2	2	0.500	8	7	−2.46	−1.56	625.0	0.00	1	2.248	5.055	641
2	2	0.062	10	6	8.00	−8.93	0.00	16.00				
3	2	0.565	10	6	8.00	−8.93	0.00	256.0	2	7.038	5.712	531
3	2	0.565	8	17	3.31	−15.99	0.00	16.00				
3	2	0.043	8	9	1.17	−1.17	3.00	256.0				
4	2	0.587	10	11	−1.37	1.00	0.00	324.0	3	8.082	15.00	677
4	2	0.630	10	9	−1.58	1.78	256.0	0.00				
4	2	0.130	8	7	−2.46	−1.56	16.00	81.00				
5	8	0.587	10	11	0.00	0.00	0.00	0.00				
5	8	0.630	10	9	0.00	0.00	0.00	0.00				
5	8	0.630	8	9	0.00	0.00	0.00	0.00				
5	8	0.173	17	8	0.00	0.00	0.00	0.00				
6	4	0.587	10	11	0.00	0.00	0.00	0.00				
6	4	0.587	17	16	0.00	0.00	0.00	0.00				
6	4	0.630	10	9	0.00	0.00	0.00	0.00				
6	4	0.173	8	9	0.00	0.00	0.00	0.00				

[a] Soln no. = solution number; cmp = molecular complexity; maxd = globularity index; At*n* = atom number; Pol*n* = effective polarity; Sgi*n* = interference level; level = solution rank; str val = strategy value; Pol val = polarity value; Sgi val = interference value.

Table 13. Second Run Solutions for Amarolide[a]

soln no.	cmp	maxd	At1	At2	Pol1	Pol2	Sgi1	sgi2	level	str val	pol val	Sgi val
1	8	0.522	17	16	−11.51	7.96	16.00	64.00	4	27.12	5.237	757
1	8	0.565	8	9	2.00	−1.17	3.00	625.0				
1	8	0.043	10	6	8.00	−8.93	48.00	1.00				
2	2	0.565	10	6	8.00	−8.93	0.00	256.0	1	2.508	5.055	353
2	2	0.062	8	7	−2.46	−1.56	81.00	16.00				
3	2	0.522	8	17	3.31	−15.99	3.00	16.00	2	6.780	5.712	2504
3	2	0.565	8	9	1.17	−1.17	3.00	2401.0				
3	2	0.043	10	6	8.00	−8.93	0.00	81.00				
4	2	0.500	8	7	−2.46	−1.56	625.0	0.00	3	7.434	15.00	1665
4	2	0.609	10	11	−1.37	1.00	0.00	1024.0				
4	2	0.130	10	9	−1.58	1.78	16.00	0.00				

[a] Soln no. = solution number; cmp = molecular complexity; maxd = globularity index; At*n* = atom number; Pol*n* = effective polarity; Sgi*n* = interference level; level = solution rank; str val = strategy value; pol val = polarity value; Sgi val = interference value.

ASSIGN THRSH and VERIFY THRSH. Two packets which assign the current thresholds and check the correlation with the analysis level.

LAST CHECK. It contains just one rule (ADD STEP) whose importance is, however, fundamental; its work begins when EXPAND TREE has finished. At this time that are "accepted" solutions, which must exit the process, and "discarded" solutions, which must be recovered in order to continue the analysis; the packet exactly selects the right set for each solution.

We have thus introduced objects and rules of ES, and we would like to point to their important characteristics: (1) objects contain the knowledge; (2) standard rules operate on the knowledge, change it, and create new knowledge; (3) special rules make possible the interaction between ES (the knowledge manager) and Lilith (the knowledge maker).

It is important to emphasize the principal differences between standard program and expert system operations. In a standard program the flow of control is basically top-down regulated; each statement is processed as soon as it is encountered, the actions found inside the statement are applied to all the data which satisfy certain conditions, and the control is passed to the following statement. In an expert system a special control mechanism is present: the "agenda". It contains all the packets (and the corresponding rules) that are currently active together with all the object occurrences that satisfy the required conditions. The presence in the agenda is dynamically controlled; therefore as soon as either a rule becomes eligible for firing (because of an explicit call or any control chaining) or an object occurrence enters or leaves the agenda (because of a change in its value), the agenda changes and the rule application is modified as well. The consequence is a complete simultaneous control on the entire knowledge base. In the perspective to effectively connect Lilith to the system, we can easily imagine the same kind of dynamic control also applied to Lilith, therefore fully implementing the IAIA principle.

Table 14. Third Run Solutions for Amarolide[a]

soln no.	cmp	maxd	At1	At2	Pol1	Pol2	Sgi1	Sgi2	level	str val	pol val	Sgi val
1	8	0.565	10	6	8.00	−8.93	0.00	256.0	4	29.712	5.237	1877
1	8	0.630	8	9	2.00	−1.17	0.00	1296.0				
1	8	0.043	17	16	−11.51	7.96	1.00	324.0				
2	2	0.500	8	7	−2.46	−1.56	625.0	0.00	1	2.248	5.055	641
2	2	0.062	10	6	8.00	−8.93	0.00	16.00				
3	2	0.565	10	6	8.00	−8.93	0.00	256.0	2	7.428	5.712	1559
3	2	0.630	8	9	2.00	−1.17	3.00	1296.0				
3	2	0.043	8	17	3.31	−15.99	3.00	1.00				
4	2	0.500	8	7	−2.46	−1.56	625.0	0.00	3	7.824	15.00	1665
4	2	0.674	10	9	−1.58	1.78	0.00	0.00				
4	2	0.130	10	11	−1.37	1.00	16.00	1024.0				

[a] Soln no. = solution number; cmp = molecular complexity; maxd = globularity index; At*n* = atom number; Pol*n* = effective polarity; Sgi*n* = interference level; level = solution rank; str val = strategy value; pol val = polarity value; Sgi val = interference value.

Table 15. Dump Performance by RULE[a]

fire-cnt	true-cnt	percent	rule	fire-cnt	true-cnt	percent	rule
45	54	22.53	growing	0	0	0.59	CALC VALUES22
9	18	21.92	ADDING LEVEL	9	9	0.58	CURR TOTAL
90	90	17.18	STOPPING	27	9	0.51	START CAOS
9	18	5.80	FIRST TIME	0	9	0.48	STOP CAOS
30	83	4.41	REF VAL2	9	9	0.39	LEVEL TO TRSH3
18	18	3.41	CALC VALUES3	9	9	0.39	LEVEL TO TRSH1
27	27	3.28	CHECK CALC	9	9	0.37	LEVEL TO TRSH2
13	75	2.72	REF VAL1	18	18	0.25	COMPLETE-IT
0	0	2.57	ADD FLAG	0	0	0.24	ZEROING FLAG VAL
18	18	2.03	CALC VALUES1	0	0	0.22	DO WRITE
36	36	1.79	ADD STEP	18	18	0.20	DO-IT
9	9	1.69	REF VAL3	9	9	0.19	TO CHECK
22	63	1.54	EVAL	0	0	0.19	CHANGE-IT
9	9	1.18	LAST TIME	0	0	0.19	CHANGE-IT2
18	18	0.79	CALC MAXD	0	0	0.18	CHECK-IT ORIGINAL
0	0	0.76	CHECKING	243	243	0.11	ZEROING CURR
14	14	0.69	CALC VALUES21	18	18	0.03	LILITH INPUT
4	4	0.61	CALC VALUES23	1	1	0.00	CREATE ROOT

[a] Fire-cnt is the number of times that a rule has actually fired during a consultation. True-cnt is the number of times that a rule has been evaluated to true and is eligible to fire, whether or not the rule has actually fired. Percent is the percentage of time spent evaluating the pattern in the left-hand side of the rule. Rule is the rule name.

Everything could appear straightforward, but we still have some troubles. First, Lilith has internal evaluation routines which, at each phase, temporarily discard solutions; second, the number of accepted solution can be very small, particularly because SGIs are very seldom absent; third, the solution order is too dependent on strategy. The first problem is easy to solve (recovering some of the discarded solutions), but we must choose when solutions should be recovered. The decision is still open; presently we do not recover solutions. The second problem partly depends on the previous one, but it is mainly a chemical problem. In fact, if we have few solutions because the molecule is simple, we do not need to continue the analysis; if the reason is due to the particular structure of the target (with unevenly distributed complexity), a different strategy approach could solve the problem; if the reason is due to the reactivity of the molecule (SGIs), the solution is located in a future development of Lilith (i.e. the reaction condition block). In all the cases the structure of ES remains unchanged. The third problem, on the contrary, has been already considered and solved. In fact, when ES considers a solution, it automatically calculates its future cost (i.e. the polarity cost) in order to choose the best solution; it is therefore quite immediate to combine the actual and future costs and to compare them since the beginning. In this way both the strategy and the polarity values participate to the choice. Concerning the SGI value, it is clear that its weight cannot have an importance similar to the others.

At this point we have chosen the problem, i.e. the control system; the main solution, i.e. an expert system (ES); and the reasoning strategy, i.e. the hypothetical reasoning. We can now discuss some examples of application.

RESULTS AND DISCUSSION

The structures of the molecules used to test the ES part of our program are reported in Figure 1.[8] They present increasing structural complexity and different planning histories, different numbers of bond breaks, and different functionalities. First we will briefly comment on the synthetic solutions, and then we will use the rest of the section to show the work of ES.

Lilith's strategy is strongly directed toward highly convergent synthetic pathways; therefore its solutions are very often quite unexpected; moreover, the different weight given to strategy and reactivity can suggest solutions with combinations of functionalities with many interferences. These problems are to be solved by our modification of the program; in this sense we would like to a priori suggest a reserve of goodwill to the reader when looking at the results. The first consideration we can do concerns the number of proposed solutions. Because the procedure combinatorially considers the bond breakds around the complexity center, the number of solutions depends on the number of atoms at the molecular center and therefore molecules with different levels of complexity can given similar numbers of solutions. Moreover, when the second evaluation phase of Lilith comes at work, it can happen that many solutions are discarded in the most complex cases. For example looking at Tables 1 and 7 we can observe that more solutions survive for cyclohexenone than

Table 16. Dump Performance by PACKET[a]

fire-cnt	true-cnt	percent	calls	depth	packet
193	252	65.15	27	1	EXPAND TREE
90	90	11.19	22	1	EVALUATE NODE
52	167	8.81	38	1	ASSIGN THRSH
9	18	5.80	9	1	DO VALUES
0	0	2.57	0	0	ADJUST VALUE
36	36	1.79	9	1	LAST CHECK
27	27	1.15	38	1	VERIFY THRSH
27	18	0.98	1	1	MAIN
0	0	0.98	0	0	WRITE CONTR
18	18	0.76	18	1	INSERT DATA
252	252	0.54	27	1	CURR TO ZERO
18	18	0.25	18	1	COMPLETE MOLECULE
18	18	0.03	18	1	READLIL
1	1	0.00	1	1	INIT TREE

for overall dump performance:[b] total activity, 32 143; run time, 35.92 s

[a] Fire-cnt is the number of times that rules in the packet has actually fired during a consultation. True-cnt is the number of times that rules in the packet have been evaluated to true and are eligible to fire, whether or not the rules have actually fired. Percent is the percentage of time spent evaluating the pattern in the left-hand side of all rules in the packet. Calls is the total number of times that the packet has been invoked. Depth indicates how deeply the packets have been recursively called. Packet is the packet name. [b] For Tables 15 and 16.

for Bilobalide (at the first stage Lilith creates more solutions for Bilobalide as shown in Table 7). It is clear that ES must recover some of the discarded solutions; for example there are solutions with equal or even greater strategical value than those surviving (solutions 3, 5, 7, and 8). The second remarks concern the interference weight. We already pointed to this problem, but we think that it is important to discuss it a bit deeper. If we look at Table 1, last column, we can observe that solution 8 has a very low SGI value even if it ranks only fourth; but looking at the precursor structures (Figure 2), the difference is less clear-cut. In fact the difference in SGI between solutions 8 and, for example, 4 is really small and very difficult to find. Moreover the lack of any care for the reaction conditions makes it impossible to separate the solvable from the insoluble interferences.

Concerning the improvements made by ES, we will consider each molecule.

Tables 1 and 2 show the ES results for cyclohexenone with two different bond breaking sequences. The most exciting result concerns solution 4 which changes rank from 7 to 4. In fact in the second ordering a previously undefined bond polarity becomes well defined, changing the POL value from 5.046 to 0.075. If we look at the precursor structure, the result is evident; breaking bond 4–5 after bond 6–7 defines the polarity of atom 5 (α-carbonyl position) and, as a consequence, also the polarity of atom 4. The other clear difference concerns the SGI values which generally improve.

Tables 3 and 4 concern the solutions for quinic acid (Figure 3). The only evident result is the automatic selection of the molecular symmetry in solutions 2 and 3, symmetry which is not explicitly searched by Lilith. For this molecule we ran Lilith a second time with a different strategy set-up (Tables 5 and 6); the solutions change and are altogether worse than the previous ones, but one of them ranks as the best (solution 2, first run). But, it is overcome by solution 1 (Table 3) in the second run where a defined polarity becomes a contrasting polarity (atoms with the same polarity sign). This last effect, however, does not change the rank inside the second analysis. It is evident that the use of ES allows a deeper analysis of the synthetic tree.

Tables 7–9 show the solutions for bilobalide. In Table 7 we report all the solutions proposed by Lilith in order to emphasize those being discarded. It is easy to note that some solutions which have been discarded have low strategy values; their analysis is not a priority, but the possibility of their recovering is fundamental. Because of the presence of three bond breaks we obtain three result sets. In this case the final rank remains constant but it is interesting to note how great is the change in the values, even concerning the strategy. Looking at numbers with greater accuracy, it is possible to point out the reasons for the ranking of the solutions, in particular the strategy value; the difference between solutions 1 and 2 is more related to the differences in the globularity indices than to those in the complexity values (columns 2 and 3). Looking at the structure, this result shows an evident connection with the strategic bond sets which for solution 1 separate a globular piece from a straight piece, while for solution 2 the globularity level of the two pieces is very similar (see Figure 4). Moreover the change in the bond break sequences causes a corresponding change in the globularity indices particularly in the third run for solution 2 (see Figure 4).

Tables 10 and 11 report the solutions for Actinobolin. It is possible to note again the change in the rank order due to the change in the polarity value for solution 2, which in the first run has a good level of definition of the polarities while in the second run shows a bad case of contrasting polarities. The second interesting result is the diminution of the interference level in the second run, thus confirming our impression that SGIs require a deeper analysis in order to become fully integrated with the rest of the program. Looking at some of the real synthesis for this molecule,[9] we can note that the use of the Diels–Alder reaction made the skeleton of the molecule by constructing the ring which is untouched by Lilith. In this case a second analysis with automatic strategy change could suggest some interesting results (as we experimentally verified).

Tables 12–14 shows the results for amarolide. We would like to point out two considerations: (1) the positions of solutions 2 and 4 which respectively score better or worse than expected because solution 2 breaks only two bonds and solution 4 has a bad polarity setting in all runs; (2) the two solutions discarded by Lilith (in this case the main reason is the number of the strategic bonds (4) which is too high). Even if this result is in agreement with the program philosophy and also with the specific synthetic problem, we know different cases where the simplification introduced by many bond breaks is really important, therefore making appealing an otherwise not too good solution.

An example of the performance of the ES section is reported in Tables 15 and 16, where both the rule and the packet analysis is shown. It is evident that the rules governing the construction of the solution tree are the most used and the most time consuming, as expected; on the contrary, we have no chance to compare ES and Lilith efficiency because they still run on different machines. The structure of the system is quite clear with packets dedicated to special purposes and without too much recursive calls; however, it has not been fully optimized yet.

As a general comment we would like to emphasize the great amount of new information brought into the system by the use of ES even in this preliminary phase of realization. Moreover, for those unfamiliar with true expert systems, we also wish to clarify ES working ability. In fact, using its basic features (object definition, rules, agenda conditions, backward–forward activation), it is possible to process new solutions as soon as the system characteristics are varied by whatever cause

without requiring an explicit new control phase.

CONCLUSIONS

The results of the present analyses show the capability of the system to realize a first part of the final objective. In fact it is clear that the ES can control the decision flow with good accuracy, and, moreover, it can complete the application of the IAIA principle. We feel to be on the right way to reach the final solution. For the sake of honesty we must emphasize a couple of still unresolved problems: (1) the ES writing shell is not working in the same area of Lilith and therefore the communication between the two is still hard; (2) we are considering the opportunity to split more of the control activity in order to get the most from ES (considering, for example, each bond break as a self-standing solution).

ACKNOWLEDGMENT

Partial financial support by the Consiglio Nazionale delle Richerche and by the Ministero dell'Universitá e della Ricerca Scientifica e Tecnologica is gratefully acknowledged.

REFERENCES AND NOTES

(1) For a review on Computer Assisted Organic Synthesis see: *Computer aids to chemistry*; Vernin, G., Chanon, M., Eds.; Ellis Horwood: Chichester, U.K., 1986. See also the series of papers in: *Anal. Chim. Acta* **1990**, *235*, 1–239.

(2) Baumer, L.; Sala, G.; Sello, G. Organic Synthesis Planning: An Algorithm for Selecting Strategic Bond Forming Sequences. *Tetrahedron* **1989**, *45*, 2665–2676.

(3) (a) Baumer, L.; Sala, G.; Sello, G. Organic Synthesis Planning: A New Algorithm for Strategic Bond Perception. *Tetrahedron* **1988**, *44*, 1195–1206. (b) Baumer, L.; Sala, G.; Sello, G. Organic Synthesis Planning: Fundamental Principles in a New Approach (Lilith). *Anal. Chim. Acta* **1990**, *235*, 209–214. (c) Baumer, L.; Campagnari, I.; Sala, G.; Sello, G. Erythronolide B: the analysis of the synthesis proposed by the Lilith program. *Rec. Trav. Chim. Pays-Bas* **1992**, *111*, 297.

(4) (a) Orsini, F.; Sello, G. Geometric Requirements for Reactivity: The Simulation of Access to Reaction Centers and the Influence of Atomic Deformation on It. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 451–457. (b) Baumer, L.; Sala, G.; Sello, G. Computer Assisted Organic Synthesis Planning Effective Bond Polarity as a Guideline to Reactivity. *J. Am. Chem. Soc.* **1991**, *113*, 2994–2500. (c) Baumer, L.; Sell, G. A New Method for the Calculation of Natural Bond Polarity Using Molecular Electronic Energy. *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 125–130. (d) Baumer, L.; Sala, G.; Sello, G. Similar Group Interferences. A general approach to the location of interfering functionalities. *Tetrahedron* **1993**, *49*, 3367–3386.

(5) (a) For a review on artificial intelligence applied to chemistry see: *Artificial intelligence applications in chemistry*; Pierce, T. H., Hohne, B. A., Eds.; ACS Symposium Series 408, American Chemical Society: Washington, D.C., 1989. (b) Gunasingham, H.; Wong, M. L. Deterministic Control in Knowledge-Based Systems: Application to the Development of a Cybernetic Analytic Instrument. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 211–215. (c) Eggert, A. A.; Middlecamp, C. An Oxidation Number Assignment Expert for CHEMPROF. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 181–187. (d) Huixiao, H.; Xinquan, X. ESSESA: An Expert System for Elucidation of Structures from Spectra. 1. Knowledge Base of Infrared Spectra and Analysis and Interpretation Programs. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 203–210. (e) Leach, A. R.; Dolata, D. P.; Prout, K. Automated Conformational Analysis and Structure Generation: Algorithms for Molecular Perception. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 316–324.

(6) KBMS for Microsoft Windows, AICorp, Inc., 138 Technology Dr., Waltham, MA. Release 405, Version E01, 1992.

(7) The ES shell we are presently using is a DOS version that we cannot use in direct connection with Lilith (Unix version); therefore the communication between the two parts of the program is still difficult. We are looking forward to the RISC version of the ES shell expected for the end of the year.

(8) The molecules used as examples, excluding cyclohexenone, have been randomly taken from: *The Logic of Chemical Synthesis*; Corey, E. J.; Cheng, X., Eds.; Wiley Interscience: New York, 1989.

(9) (a) Yoshioka, M.; Nakai, H.; Ohno, M. Stereocontrolled Total Synthesis of (+)-Actinobolin by an Intramolecular Diels–Alder Reaction of a Chiral Z Diene Derived from L-Threonine. *J. Am. Chem. Soc.* **1984**, *106*, 1133–1136. (b) Garigipati, R. S.; Tschaen, D. M.; Weinreb, S. M. Total Synthesis of (+)-Actinobolin. *J. Am. Chem. Soc.* **1985**, *107*, 7790–7794. (c) Kozikowski, A. P.; Nieduzak, T. R.; Konoike, T.; Springer, J. P. π-Facial Selection in Intermolecular Diels–Alder Reactions: Total Syntheses of (+)-Actinobolin and (+)-5,6,10-triepi-Actinobolin. *J. Am. Chem. Soc.* **1987**, *109*, 5167–5171.