ELSEVIER

Artificial Intelligence 91 (1997) 225–256

Artificial Intelligence

The computer revolution in science: steps towards the realization of computer-supported discovery environments

Hidde de Jong[a,*], Arie Rip[b,1]

[a] *Knowledge-Based Systems Group, Department of Computer Science, University of Twente, P.O. Box 217, 7500 AE Enschede, Netherlands*

[b] *School of Philosophy and Social Sciences, University of Twente, P.O. Box 217, 7500 AE Enschede, Netherlands*

Received December 1995; revised September 1996

Abstract

The tools that scientists use in their search processes together form so-called discovery environments. The promise of artificial intelligence and other branches of computer science is to radically transform conventional discovery environments by equipping scientists with a range of powerful computer tools including large-scale, shared knowledge bases and discovery programs. We will describe the future computer-supported discovery environments that may result, and illustrate by means of a realistic scenario how scientists come to new discoveries in these environments. In order to make the step from the current generation of discovery tools to computer-supported discovery environments like the one presented in the scenario, developers should realize that such environments are large-scale sociotechnical systems. They should not just focus on isolated computer programs, but also pay attention to the question how these programs will be used and maintained by scientists in research practices. In order to help developers of discovery programs in achieving the integration of their tools in discovery environments, we will formulate a set of guidelines that developers could follow. © 1997 Elsevier Science B.V.

Keywords: Scientific discovery; Computer-supported discovery environments; Social studies of science and technology

[*] Corresponding author. E-mail: hdejong@cs.utwente.nl.

[1] E-mail: a.rip@wmw.utwente.nl.

PII S0004-3702(97)00011-8

1. Introduction

Knowledge in a scientific domain is a heterogeneous collection of experimental findings, regularities and patterns, explanatory models, and theories, which continuously develops through the activities of scientists. Scientists are engaged in open-ended search processes, in which they construct new empirical phenomena, devise models accounting for these findings and propose theories relating a wide range of empirical phenomena. These search processes are local: they take place at laboratory benches, behind desks in research institutes, or in field work at distant locations. The generality of science is a consequence of the communication of the discovery claims generated in local search processes to relevant audiences within a scientific community. If these audiences are convinced of the value of the results, the claims are added to the shared body of knowledge in a domain [52]. The extension of the body of knowledge with new discoveries may lead to further research and further discoveries, thus giving rise to the characteristic dynamic of science.

In structuring the shared body of knowledge in their domain and performing discovery activities, scientists make use of the heuristics and rules of the research practice in which they work. These heuristics and rules are sometimes systematized into explicit methods and made available as tools to scientists grappling with similar problems. Philosophers of science have for instance described and evaluated how experimenters reconstruct their activities, and the skills these activities exemplify, into procedures that others can use in order to judge and replicate their findings [21,47]. The systematization of scientific work and knowledge may lead to the further material equipment of a practice, for instance when heuristics and rules are embodied into an apparatus performing a specific, self-contained subtask. The acidity of a liquid, its pH, is nowadays measured by putting electrodes in the liquid and reading off the pH from a scale. Within such a piece of lab equipment a number of electrochemical regularities are embodied.

The tools that have been thus constructed, both methods and procedures and material devices, together form a *discovery environment*, in which scientists can pursue their search processes [14]. Historians and philosophers of science have given detailed descriptions of discovery environments, such as the air-pump and related methods and instruments that Boyle used for his pneumatic experiments around 1660 [54], a modern biological lab focusing on neuroendocrinological research [37], and the complex installations for running experiments in high-energy physics [20].

The promise of AI, and of computer science in general, is to transform conventional discovery environments through widening the scope and increasing the depth of the systematization and material equipment of a research practice. Systematic methods for tasks like designing and planning experiments and finding regularities in large amounts of data are now being implemented in automatic discovery programs. Existing systematizations of bodies of knowledge, for instance in the form of handbooks or ordered files with experimental results in office drawers, are transformed into machine-readable databases and knowledge bases (Fig. 1). The resulting computer tools, integrated in existing scientific discovery environments, give rise to what we will call *computer-supported discovery environments*. This development has led some to speculate about

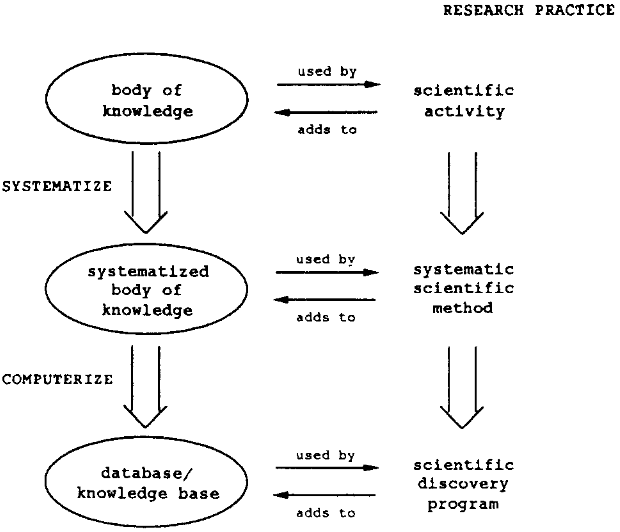

Fig. 1. Systematization and computerization of the body of knowledge and scientific activities in a research practice.

"radical, and perhaps surprising, transformations of the disciplinary structure of science" (Allen Newell, quoted in [6]). How present, well-tested AI techniques, supplemented with new and promising techniques, can help create these computer-supported discovery environments, and what this means for science, is the topic of this article.[2]

In Section 2, we will describe the computer-supported discovery environments of the future and the way these discovery environments are used by scientists in their search processes. To illustrate and reinforce our points, a realistic scenario of the discovery of a new genetic regulation mechanism in the *E. coli* bacterium will be introduced. In Section 3 we will then argue that computer-supported discovery environments cannot be adequately understood as a mere collection of computer tools, but that we have to view them as sociotechnical systems embedded in a research practice. This widening of perspective has implications for the design of discovery tools: we should not just focus on the task performance of isolated computer programs, but also pay attention to the way they are organized in a discovery environment. Section 4 reviews how far we have come toward the realization of computer-supported discovery environments and concludes that especially issues surrounding the integration of discovery tools in research practices have hardly been touched upon. In Section 5 we will therefore present a number of guidelines

[2] Amidst the enthusiasm about future computer-supported discovery environments, we should not forget, however, that the emergence of ideas on the large-scale systematization of scientific practices did not coincide with the availability of advanced computer techniques. In fact, philosophers from the past as well as from our age have shown much interest in systematizing scientific knowledge and activities. For example, Bacon and Leibniz, among others, propagated the construction of natural and experimental histories and the development of an *ars inveniendi* to derive new discoveries from these histories.

that developers could follow when they strive at embedding discovery programs and other computer tools in practices of scientists. Section 6 recapitulates the main points and offers some concluding remarks.

2. Search processes in computer-supported discovery environments

In the *computer-supported discovery environments* that are now envisaged, powerful discovery systems will find a place alongside other tools already available to scientists. Although we do not intend to propose a detailed technical architecture of discovery systems, one could think of them as being built around relatively passive resources like databases and knowledge bases, and active discovery programs like hypothesis generators, process simulators, and theory revision assistants (as suggested by Fig. 1). Telephone, e-mail, and other communication tools will also form a part of the discovery environments of the future. In addition we will meet with the conventional measuring and analytic equipment currently found in laboratories; computer-supported discovery environments gradually develop from existing discovery environments and retain elements from them.

A computer-supported discovery environment is not just a loose aggregate of tools, but an *integrated system.* The tools are mutually related and adjusted, so that for instance the results of a bioassay can be processed by an analysis program and subsequently sent forward to a distributed database, which stores experimental results obtained in millions of bioassays performed at laboratories across the world. The order and structure in a computer-supported discovery environment is not an inherent characteristic of the tools, but is imposed by the research practice in which the discovery environment is embedded. Through the way in which the tools are employed in the search processes of scientists, relationships between the components of a discovery environment emerge and become established. These relationships may be implicit, contained in the skills of scientists who know how to handle and combine the tools, but sometimes they are made explicit. Think, for example, of program manuals and operating and maintenance procedures; but also of relationships embodied in cables running through the building and other material connections.

This outline of computer-supported discovery environments gives only the bare bones of the potential and the promise of the emerging forms of computer support for scientists. In order to dress the bones of our outline with some flesh, we have to look for examples of computer-supported discovery environments. Since no such discovery environments exist to date, we have chosen to give substance to our argument by taking examples from a *scenario* of a future computer-supported discovery environment in molecular biology. The scenario describes a particular configuration of computer tools and gives an insight into the way they are used by scientists. Although the computer-supported discovery environment presented in the scenario does not exist, it is realistic in the sense that it is an extrapolation of current developments in the discovery environments of molecular biology. The plausibility of the scenario will be argued further in Section 4, when we make an inventory of existing discovery programs and other computer tools in scientific discovery environments.

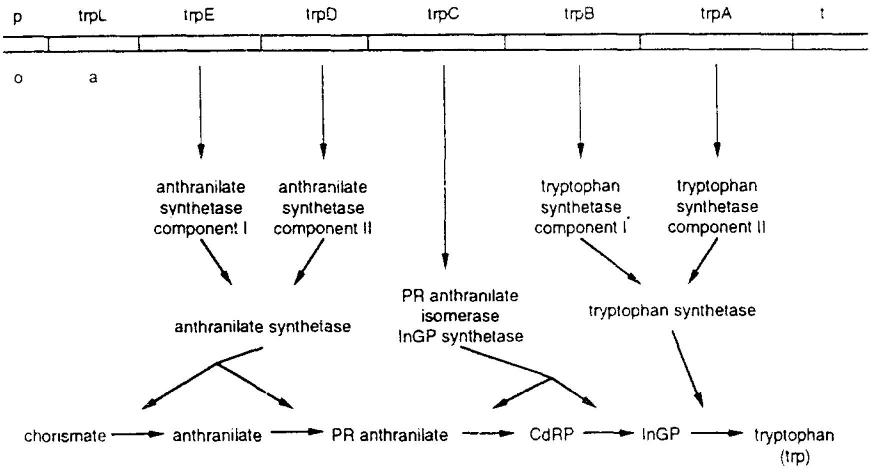
p
trpL
trpE
trpD
trpC
trpB
trpA
t
o
a
anthranilate synthetase component I
anthranilate synthetase component II
tryptophan synthetase component I
tryptophan synthetase component II
anthranilate synthetase
PR anthranilate isomerase InGP synthetase
tryptophan synthetase
chorismate
anthranilate
PR anthranilate
CdRP
InGP
tryptophan (trp)

Fig. 2. The trp operon in E. coli (figure reproduced from [26] with kind permission from Kluwer Academic Publishers). Five genes (trpE to trpA) code for five polypeptides which in turn form three enzymes. The enzymes catalyze the reactions that produce tryptophan (trp). Also visible in the figure are the genetic control regions called the promoter (p), operator (o), and terminator (t). RNA polymerase binds to the operon at the promoter region and transcribes the DNA until the terminator is reached. This transcription process is blocked when an active repressor molecule binds to the operator region. The function of the DNA region labeled "a" is elucidated in the search processes described in the scenario.

The scene of action in the scenario is a Boston university, where the molecular biologist Bruce Noyafsky and his colleagues investigate the regulation process of the trp operon of the *E. coli* bacterium (Fig. 2). At the start, only the repressor mechanism is known as a regulating factor in the expression of trp genes. By means of the computer tools in their discovery environment the researchers gradually develop ideas on a second regulation mechanism: attenuation.[3]

We have divided the scenario into five fragments that are introduced as illustrations at particular points of our argument. The main story line of the scenario is contained in Fragments 1, 2, and 5 (Fragments 3 and 4 are elaborations of episodes occurring in Fragments 1 and 2, respectively).

Fragment 1 (An introduction to Noyafsky's discovery environment). *In the quiet university building Bruce Noyafsky leaned back in his chair. It was late and he should have been home by now, but an annoying thought continued to nag in him. Their research on the regulation process of the trp operon had seemed to be clear and definitive, but lately a number of small problems had undermined the trust in their results. Also with him. They were on the right track, he was sure about that, but there was something that did not quite fit. This uncomfortable feeling had only been intensified*

[3] The scenario is based on a free adaptation of a real discovery, which was also used by Karp in his work on computational models of regulation processes [25,26]. For biological details on this discovery, see [70] and [56], especially Section 3.11d.

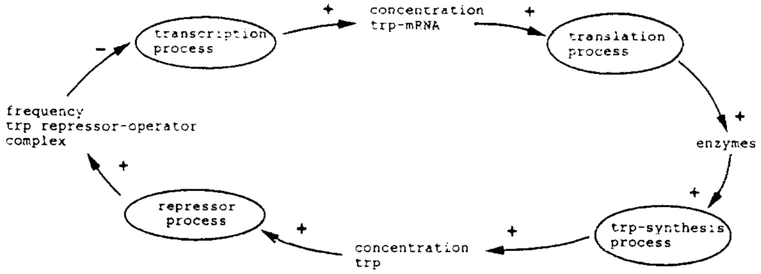

Fig. 3. Global qualitative model of the regulation of the trp operon before the discovery of the attenuation mechanism. An arrow represents a causal relation. A plus sign with an arrow from *A* to *B* means that an increase (decrease) of *A* implies an increase (decrease) of *B*. Conversely, a minus sign with an arrow from *A* to *B* means, that an increase (decrease) of A entails a decrease (increase) of *B*.

this morning, when he received an e-mail from Pierre in Paris to inform him about the latest laboratory tests performed over there, which seemed to indicate a new difficulty for the regulation model. Pierre had promised to forward the results today, at the same time when submitting them to the Regulation Process Database (RPD), *so that he would not have to wait until the Supervisory Committee had checked the data. Those bureaucrats were able to hold up important scientific findings for more than two weeks.*

To divert himself, Noyafsky browsed through the giant information network accessible from his workstation. As a matter of course, he selected the entry "The regulation of the trp operon" from the index of the large Genetic Regulation Processes (GRP) knowledge base *at Denver, Colorado. He remembered how proud he had been when a commission of peers decided to have his submission to the electronic newsgroup of the* Journal of Molecular Biology *modeled and included in the GRP. His name and the article in which he proposed the regulation model were mentioned under the schematic representation of the* qualitative model (*Fig.* 3), *as visible reminders of his contribution.*

It wouldn't harm to go through the reasoning steps in the regulation model once again, Noyafsky thought (*Fig.* 3). *He clicked with his mouse on the icon for the repressor process and the latest version of the model of this process was immediately transferred from Denver* (*Fig.* 4). *The model of the repressor process described how first trp and trp aporepressor merge into a trp-repressor molecule. This molecule can bind to the DNA strand at the trp operator site, which gives rise to a so-called repressor-operator complex. Because the operator and promoter regions overlap, the transcription process is blocked when a repressor-operator complex is formed on the DNA strand. RNA polymerase cannot bind to the promoter when the operator site is occupied by a trp-repressor molecule and consequently the transcription process cannot start.*

Once again the reasoning mechanism for the qualitative model set out to him the individual steps in the regulation of the trp production (*Fig.* 3): *when the trp concentration increases, the rate of the production of trp-repressor increases as well, resulting in a higher frequency that a trp-repressor complex is formed on the DNA. This implies that*

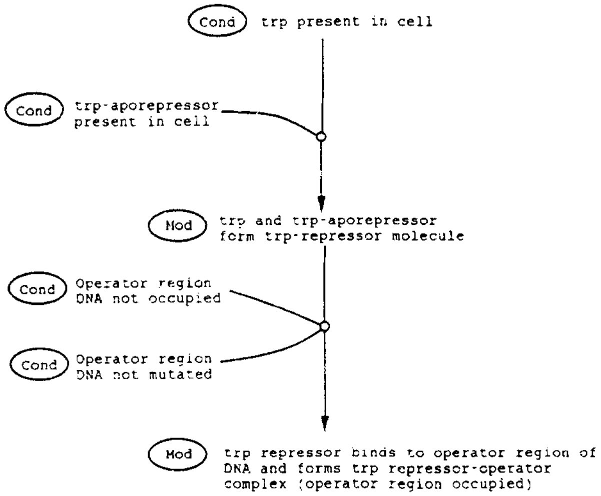
Cond
trp present in cell
Cond
trp-aporepressor present in cell
Mod
trp and trp-aporepressor form trp-repressor molecule
Cond
Operator region DNA not occupied
Cond
Operator region DNA not mutated
Mod
trp repressor binds to operator region of DNA and forms trp repressor-operator complex (operator region occupied)

Fig. 4. Model of the repressor process. The model consists of a number of conclusions of reasoning steps (indicated with Mod) and a number of conditions (Cond) that determine the applicability of a model to an experimental situation. The arrows represent reasoning steps.

the transcriptional activity decreases, leading to a lower mRNA concentration. Through the translation and synthesis processes this eventually results in a lower concentration of trp in the cell. Thus, one could conclude, the regulation process described in the model is essentially a negative feedback system: the production of trp varies inversely with the trp concentration in the cell.

Noyafsky directed his mouse to the transcription process in the qualitative regulation model (Fig. 3). A click, and a schematic structure summarizing the steps in the transcription process from DNA to mRNA appeared on the screen (Fig. 5). The model of the transcription process showed how RNA polymerase transcribed the genes of the operon, including the trpL gene for which no biological function was known, into an mRNA strand. Nothing is wrong with this, Noyafsky thought. It did not relieve his gloomy mood.

It would be possible to extend the model with details on RNA polymerase and the biochemical reactions of the transcription process. It was easy to accomplish, but he knew that once he would start to wander through the network, it would be seductive to delve into the mass of available data. In the beginning he would concentrate and focus on information that could be helpful in his research, but gradually his curiosity would lead him to completely different topics. Last week he ended with the pyramid of Cheops and the other day he had watched, fascinated, a three-dimensional simulation of a rocket launch. Tonight he did not want to work late and risk a quarrel at home. Besides, the e-mail message or telephone call from Pierre announcing that the results were available on the ftp site of the CRBM lab in Paris clearly would not arrive today. He stored the models obtained from Denver, intending to use the local copies for the daily group meeting tomorrow, and went home.

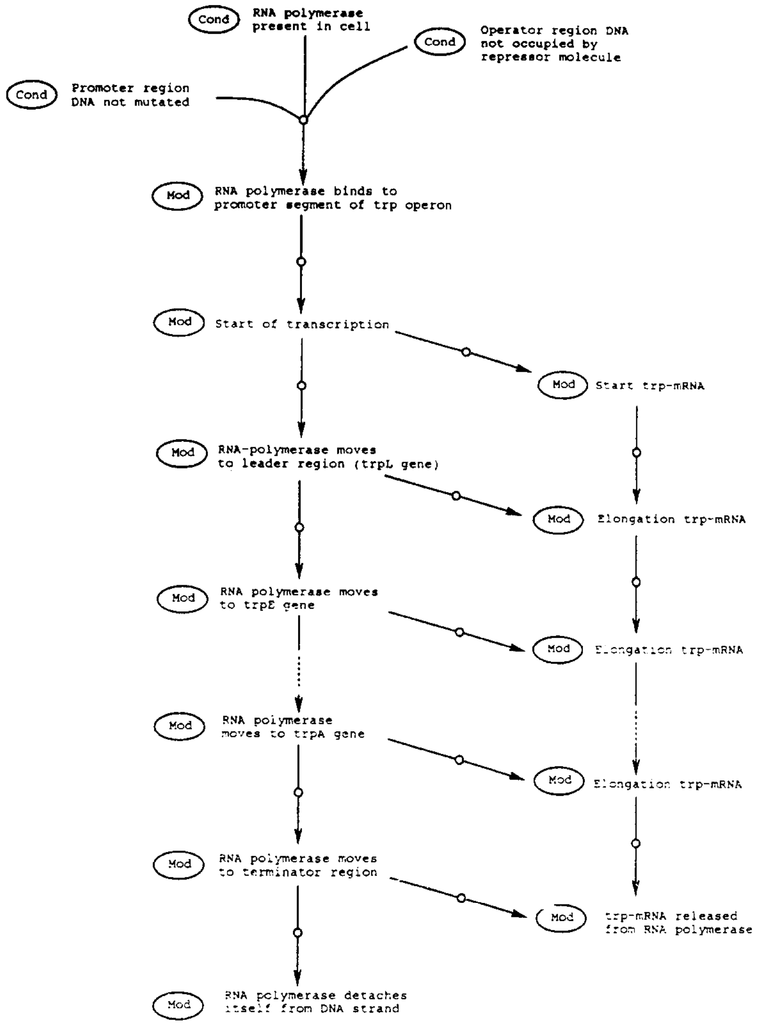
Cond
RNA polymerase present in cell
Cond
Operator region DNA not occupied by repressor molecule
Cond
Promoter region DNA not mutated
Mod
RNA polymerase binds to promoter segment of trp operon
Mod
Start of transcription
Mod
Start trp-mRNA
Mod
RNA-polymerase moves to leader region (trpL gene)
Mod
Elongation trp-mRNA
Mod
RNA polymerase moves to trpE gene
Mod
Elongation trp-mRNA
Mod
RNA polymerase moves to trpA gene
Mod
Elongation trp-mRNA
Mod
RNA polymerase moves to terminator region
Mod
trp-mRNA released from RNA polymerase
Mod
RNA polymerase detaches itself from DNA strand

Fig. 5. Model of the transcription process. See Fig. 4 for the notation used.

Noyafsky's tour of the computerized bodies of knowledge in molecular biology and other domains of inquiry introduced us to some of the powerful tools that may be found in a computer-supported discovery environment. Below we will meet other, even more impressive tools, but this initial fragment gives us a flavor of the kind of support that scientists can expect from computer-supported discovery environments. A few features that stand out are:

- The system of tools available to scientists is a *distributed* system. We can see in the scenario that the databases, knowledge bases, and supporting programs which Noyafsky uses are located at different, geographically dispersed sites: Boston, Denver, Paris.

- Although the components of a computer-supported discovery environment may be scattered throughout the world, they can be accessed and manipulated from a local workstation. The computer network by which the tools are connected abolishes geographical distances, at least in the eyes of working scientists like Noyafsky. Tools at far-away sites are *locally applicable* in the search processes of scientists.
- A computer-supported discovery environment is composed of *heterogeneous* elements. In the scenario we encountered such sophisticated tools as large-scale databases and knowledge bases, graphical browsers, integrated communication systems, simulation programs, but also tools like telephones.
- The heterogeneity of tools does not prevent them from being combined in the search process of scientists. Without problems, Noyafsky can apply a simulation program, obtained from a particular program library, to a regulation model stored in some far-away knowledge base. The computer-supported discovery environment is an *integrated* system, with the scientists coordinating the tools.
- Another conspicuous feature of the discovery environment in the scenario is the *shared* character of the tools. Noyafsky employs databases, knowledge bases, and program libraries, which because of their scale have to be set up and managed by special institutes or companies. The use and maintenance of these resources is shared among scientists: individual researchers have access to the electronic knowledge stores and contribute to their contents.
- The point of shared maintenance leads to another important feature of computer-supported discovery environments: they are in *continuous development.* A discovery environment is never finished; existing tools are modified and extended, and new tools become available all the time.

Fragment 1 showed the computer-supported discovery environment at rest. Noyafsky was idling away. In the next fragment, the discovery environment is put to work.

Fragment 2 (Finding an explanation for an experimental anomaly). *Pierre's results were indeed in conflict with the regulation model, that was rapidly agreed upon by the researchers in Noyafsky's group. Gathered in Noyafsky's room they listened to his report on the results from Paris. Pierre had performed a laboratory experiment with E. coli cells without functional trp-aporepressor proteins. Surprisingly, when the trp concentration was decreased, the cells nevertheless succeeded in increasing the production of mRNA, in plain contrast to the expectations from the model.*

John, the computer expert in Noyafsky's group, switched on the large screen in the corner of the room and started the programs, while Noyafsky continued his story. The qualitative model of the GRP knowledge base confirmed their surprise by predicting that under the conditions of the experiment the trp-mRNA production should remain unchanged. When the data from Paris were combined with the qualitative model, it was obvious that the repressor process could never start, because one of the initial conditions, the availability of functional trp-aporepressor molecules, was not satisfied (Fig. 4)*. If the repressor process was not active, then lowering the trp concentration could not influence the production of trp-repressor, which happened in the repressor process. The concentration of functional trp-repressor would remain constant, so that* *the frequency of the formation of repressor-operator complexes would not change. As a consequence, the system concluded after making a few additional reasoning steps, the concentration trp-mRNA should remain unchanged (Fig. 3). The qualitative reasoning module had established an inconsistency between the observations in Paris and the accepted regulation model.*

Now the thing was to act swiftly and stay ahead of their competitors. It was important to come up with at least a preliminary answer before the Paris results would find their way into the Regulation Process Database. The computer programs at other institutes, continuously interpreting new laboratory findings and comparing them with the library of regulation models in Denver, would not need much time to identify the inconsistency. The striking findings would immediately stir rival research groups into action. Bruce Noyafsky looked round the circle of colleagues: Who had a good idea?

Neil Sandorf, a biochemist, responded first: "I think the problem should be located somewhere in the transcription process. The transcription process seems to react to a change of trp concentration in a special way, differently from the usual repressor mechanism. But how? I don't know ...". The computer librarian Liza Bernstein broke the silence. "Why don't we have a look at what QualChem has to say?" QualChem *was a program which generated hypotheses on the basis of qualitative chemical models. Other groups had intimated it could come up with interesting suggestions, but Noyafsky and his group had never used it. Recently, the program was made available to members of the American Chemical Society, in a special program base. It remained to be seen whether QualChem was compatible with the way in which qualitative chemical processes were represented in the GRP knowledge base, but Liza would not need much time to find out. The others decided to use the graphical process simulator in the meantime, in order to check the transcription process of the trp operon once again.*

The graphical process simulator *made use of the underlying knowledge structure to visualize the sequence of steps in the transcription process. Symbols representing RNA polymerases slid over an abstract DNA strand dragging behind them a steadily growing tail of mRNA. Bruce Noyafsky saw how ribosomes translated the mRNA molecule into a number of enzymes which catalyzed the chemical reactions producing trp. While he looked at the RNA polymerases moving in succession on the screen, he wondered briefly about the strange fact that the first gene, trpL, did not make any direct contribution to the synthesis of trp. Its function was unknown. He did not have time to reflect on this, for Liza turned away from the workstation to tell them that QualChem was compatible with the regulation model. The graphical process simulator was interrupted and the group waited to see what QualChem would make of the regulation model and the conflicting data.*

QualChem consisted of a structured set of hypothesis operators which could change parts of the qualitative model. A qualitative model was viewed as a constellation of interacting processes, each composed of individual reasoning steps or subprocesses. The hypothesis operators could change conditions of processes, postulate new processes, modify effects of processes, dispute the connection between subprocesses, and adapt the initial conditions of models. The hypothesis operators were domain independent, but the user had the opportunity to indicate in which specialized knowledge bases background knowledge could be found. QualChem could thus generate relevant hypotheses for a

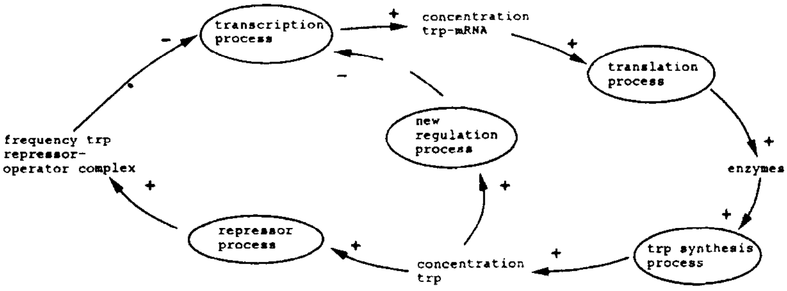

Fig. 6. Modified global qualitative model of the regulation process of the trp operon. Transcription is affected by the trp concentration through a new regulation process. For the notation, see Fig. 3.

problem presented to the program. Which hypothesis operators QualChem applied to the model was determined by its large store of strategic knowledge, gradually extended in the light of earlier results obtained with the program.

QualChem now generated the hypothesis that a new process should be added to the model, namely a regulation process by which the transcription of the trp operon was prematurely broken off at one of the six genes in the operon. Although the hypothesis did not stipulate anything about the biochemical reactions underlying the new process, it added that its activity, and thus the rate of abortion of the transcription process, was positively correlated with the trp concentration in the cell. After a click, the program elaborated the hypothesis into a modified global qualitative model (Fig. 6). Assuming that transcription was terminated at the trpL gene, it also showed how the aborted transcription process deviated from the "normal" transcription process (Fig. 7).

After these changes, the relation between the trp concentration and the production of mRNA was shaped both by the repressor mechanism and by another, thus far unknown mechanism which prematurely broke off transcription. In this situation, a decrease of the trp concentration in a cell with non-functional repressor proteins could indeed lead to an increase of the concentration trp-mRNA. After all, the rate of abortion of the transcription process would then decrease in favor of undisturbed transcription. The reasoning mechanism for qualitative regulation models could easily conclude this on the basis of the modified trp regulation model. This was interesting!

Noyafsky told his secretary to cancel all his, and his collaborators', appointments for the rest of the day.

This fragment shows how Noyafsky and his team are engaged in a search process to explain an anomaly with respect to their regulation model. They use a number of discovery tools in their attempts to modify the regulation model: a knowledge base with regulation models, a qualitative reasoning mechanism, a hypothesis generation program.

The question can be raised whether we could be more specific on what these search processes amount to and how the tools in a computer-supported discovery environment are used. A suitable entrance point is formed by the influential analysis of discovery

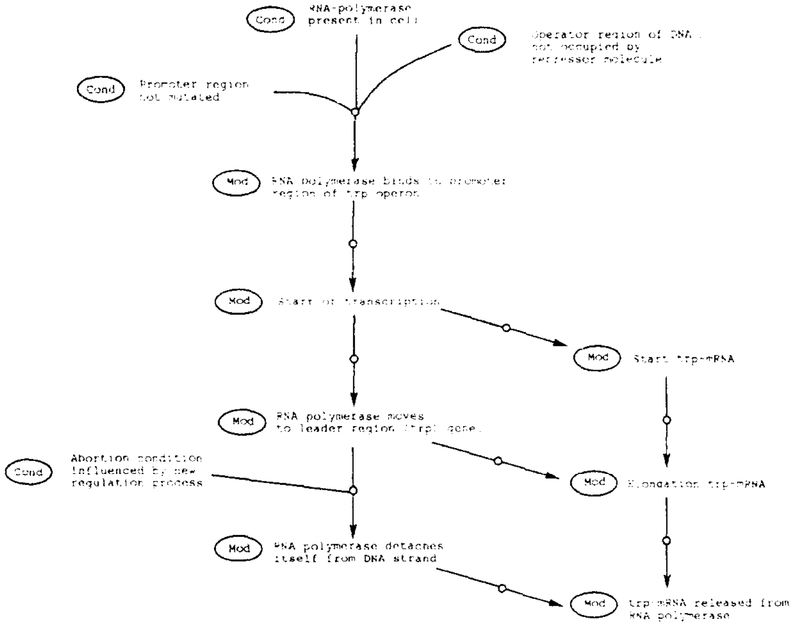
Cond
RNA-polymerase present in cell
Cond
Operator region of DNA not occupied by repressor molecule
Cond
Promoter region not mutated
Mod
RNA polymerase binds to promoter region of trp operon
Mod
Start of transcription
Mod
Start trp-mRNA
Mod
RNA polymerase moves to leader region (trp) gene
Cond
Abortion condition influenced by new regulation process
Mod
Elongation trp-mRNA
Mod
RNA polymerase detaches itself from DNA strand
Mod
trp-mRNA released from RNA polymerase

Fig. 7. Model of the aborted transcription process. For the notation see Fig. 4.

activities by Herbert Simon and his colleagues [36,55]. Simon and his colleagues view scientific discovery as a special case of the ubiquitous activity of problem solving. They claim that the way scientists set out to tackle a complicated scientific question is not qualitatively different from the way they approach an everyday problem. Scientists engage in the same kind of problem-solving behavior when they determine the structure of a molecule as when they determine the quickest route from their house to the supermarket.

The advantage of treating scientific discovery as an instance of ordinary problem solving is that one can draw on the large body of literature on the latter topic. Simon and his colleagues follow the *heuristic search approach* toward human problem solving [46] when they characterize the discovery activities of scientists. They consider problem solving as a complex form of information processing, in which a problem is represented in terms of a *problem space*. This problem space includes states of knowledge and operators to move between the states. A solution for the problem is found by a search in the problem space. Starting from an initial problem state, a sequence of operators is applied in such a way that a goal state representing a solution for the problem is reached. This search through the problem space is guided by heuristics, that is, knowledge about the problem definition and the states already explored in the problem space.

Not every scientific discovery coincides with heuristic search in a problem space, as Simon and his colleagues admit. The heuristic search model takes the problem space for granted, whereas in many discoveries the construction of this very problem space was the major achievement. In addition, so-called serendipitous discoveries are characterized by a goal state that was not defined as such in advance. These objections are valid, but do not make the model useless. Even though the articulation of a problem space necessarily precedes search and the formulation of goals, methods, and strategies may be revised while underway, many subtasks in the discovery process can still be characterized as heuristic search in a problem space. However, the critical remarks are important enough and we have to take them into account when we characterize the search processes of scientists in computer-supported discovery environments.

By means of the heuristic search model we can rephrase what Noyafsky and his colleagues are doing when they work in their computer-supported discovery environments: with the help of computer tools they define and subsequently explore a problem space. The states in the problem space consist of regulation models; more specifically, the contested regulation model forms the initial state and a regulation model that succeeds in accounting for the anomalous observation is the desired goal state. The hypothesis operators of the QualChem program can be used to move from the initial state to the goal state by changing components of a regulation model. This search process is guided by heuristics deciding which operators to apply and heuristics rating the hypotheses constructed by the operators.

The scenario provides an insight into the way scientists use a computer-supported discovery environment which can be generalized. We may formulate the use of tools as: (1) *construct* a problem space and (2) *explore* this problem space. Scientists can alternate these tasks, for instance when the exploration of a problem space gives rise to its redefinition followed by a new phase of exploration. The knowledge bases, databases, and program libraries in the discovery environment provide symbolic structures defining the initial state, operators to move to other states, and heuristics to guide the search through the problem space. In many situations the problem space is not readily available, but has to be constructed from the heterogeneous set of (electronic) resources which scientists have at their disposal. The problem space on which Noyafsky and his group concentrate has been put together from regulation models in the GRP knowledge base, a hypothesis generation program in a program base of the American Chemical Society, and the data log of an experiment directly obtained from a colleague in Paris.

The strength of the computer tools in the envisioned discovery environments derives from two circumstances. In the first place, existing problem spaces that were formerly too large to be explored can now be searched for interesting results. An example in molecular biology are the sequences of DNA, RNA, and proteins that have been collected through the years. Together these sequences represent a rich store of information on how living systems have evolved and how they operate [15]. However, the sequences used to be scattered in articles, reports, and occasional databanks. In addition, manually analyzing the sequences is a hopeless task due to the bulk of information. Now that the sequences are stored in large databases and made available to scientists [3,4], and computer programs have been developed for analyzing the sequences [3,13,18], exploring these problem spaces has become a feasible task. Molecular biologists have come up with important findings using these computer tools and they hope to find many more [15]. The shared databases and the availability of sequence analysis programs gives these problem spaces the character of *public search spaces*, open to researchers in a scientific community (see Section 5).

A second reason for the strength of computer-supported discovery environments is that they enable scientists to construct *new* problem spaces. Tasks that were hitherto unthinkable can now be formulated and carried out by means of computer tools. By combining knowledge bases, databases, and discovery programs in their own domain and in other fields, scientists can experiment with problem spaces and methods to search them. The scenario provides a clear example of the new combinations that have become possible. A regulation model in molecular genetics combined with a hypothesis generation program from the field of chemistry allows scientists to experiment with alternative regulation models, which eventually led to a new discovery (see Fragment 5). And who knows which other combinations are possible, leading to further interesting problem spaces? As we will see in Section 4, computer scientists endeavor to develop tools which allow domain scientists to open up new problem spaces.

This analysis of the manner in which scientists employ the tools in a discovery environment in their search processes makes us realize how important it is to set up a problem space, that is, to bring structure in a puzzling situation. A computer-supported discovery environment provides the means to construct and explore interesting problem spaces, but the scientists working in such an environment should have the skills to use these resources. An important step in Fragment 2 was Liza's proposal to connect the hypothesis generator QualChem to the computational models of regulation processes, which resulted in a problem space that allowed the researchers to experiment with alternative regulation mechanisms. This step depended on Liza's ingenuity and her good memory, but in the future Noyafsky's discovery environment might be extended with new computer programs supporting the generic scientific task [66] of *problem space construction*. Such programs would:

(i) analyze a task description and establish the information requirements,
(ii) search through a catalog of knowledge bases, databases, and program libraries to find tools matching these requirements, and
(iii) combine the tools into a problem space consisting of an initial state, search operators, and heuristics to guide the search.

Operating the sophisticated tools in a computer-supported discovery environment requires new skills that can in turn be systematized and transferred to specialized programs. This gives rise to a dynamic of new tools-new tasks-new tools, which puts a discovery environment in a state of constant evolution.

The efforts of Noyafsky and his group begin (and end, as will be seen below) as a typical case of normal science; their work can be described as puzzle solving within a paradigm consisting of exemplary achievements, symbolic generalizations, and cognitive norms [33]. However, one could imagine a different continuation of the story, one in which the anomaly resists all attempts at resolution and no regulation model is found capable of explaining Pierre's data (and data flowing from new experiments). Eventually, this predicament might lead to a drastic revision of the way regulation processes are understood, a scientific revolution.

Computer-supported discovery environments can assist in signaling persistent anomalies and evaluate their importance. And although the computer tools do not bring about scientific revolutions by themselves, they may urge the need to change existing approaches and pinpoint the troubles that should be addressed. Of course the transformations in a practice accompanying a scientific revolution will have their impact on the evolution of a scientific discovery environment. The system of tools has to be reorganized to accommodate the new ways of looking at problems that scientists have found useful. One could try to partially anticipate and facilitate such reorganizations when constructing a computer-supported discovery environment, as will be argued in Section 5.

3. Computer-supported discovery environments as sociotechnical systems

The promise of computer-supported discovery environments is that they may increase and improve the output of scientific discoveries, that is, enhance the *competent performance* in a research practice. So far, we have described the enhancement of competent performance as the delegation of particular discovery tasks to technical systems: databases, knowledge bases, and a variety of discovery programs. Although we met human scientists, their role in the discovery process appeared to be that of waiting until the machines delivered their findings for interpretation. However, enhancing the competence to make scientific discoveries does not reduce to the construction of new computer tools. We will argue in this section that for the enhancement of competent performance one also has to pay attention to questions concerning the organization of the tools in a computer-supported discovery environment embedded in a practice. As a prelude to the elaboration of this point, we present three further scenario fragments.

Fragment 3 (The cooperative design and maintenance of tools in a discovery environment). *Noyafsky knew that it would be possible to extend the model with details on RNA polymerase and the biochemical reactions of the transcription process. Although it would not take much effort, he did not care to do that. Those details were not so exciting, because he knew that the reliability of this knowledge, available from a number of specialized knowledge bases at West Coast institutes, was beyond doubt. The knowledge bases had been constructed by teams including recognized specialists on these topics and the assertions that were added to the knowledge bases had all passed severe validation tests. Not so long ago, there had emerged some conceptual confusion about the relations between the Polymerase knowledge base and the Genetic Regulation Processes (GRP) knowledge base, but after a meeting at a workshop in Canada the overlapping parts of the ontologies of both knowledge bases were brought into close agreement.*

Fragment 4 (Focusing the discovery process). *Noyafsky and his collaborators decided to apply the hypothesis generation program QualChem to the regulation model, in order to explain the anomaly. QualChem started with the assumption that the measurements of the trp concentration were not precise enough and that the concentration actually remained constant rather than changing as the measurements appeared to indicate. This was not a bad hypothesis, since the regulation model was based on a principle that explained a tremendous amount of empirical data. One should hesitate before tampering with a successful model. But the second hypothesis appearing on the screen did precisely* *this. It postulated a new regulation process, by which the transcription of the trp operon was prematurely broken off at one of the six genes in the operon. The rate of abortion of transcription was positively correlated with the trp concentration in the cell.*

Although QualChem did not have any preference for the gene at which the transcription process could terminate, Noyafsky suddenly realized that this could be the function of the trpL gene. The postulation of an additional link between the trp concentration and the eventual production of trp-mRNA was an interesting idea. If this link could somehow bring in the seemingly functionless trpL gene, it would be even better. Feeling a bit excited, he asked for an elaboration of the hypothesis. The program showed a modified global qualitative model (Fig. 6) *and, after a click on the mouse button, the aborted transcription process deviating from the "normal" transcription process (Fig.* 7).

Fragment 5 (Making a good idea from a promising idea). *The idea about the new regulation mechanism seemed to be promising, but was it really a good idea? Noyafsky knew that the requirements for publication were becoming stiffer, because the editors of scientific journals were flooded with off-hand ideas that researchers let their discovery programs generate. They would have to find more indications for the existence of this presumed new regulation mechanism in the trp operon before their hypothesis would be acceptable.*

Liza installed the experiment generator in order to create interesting new experiments with the help of the modified qualitative model (Fig. 6). *The* experiment generator *was a computer program that had been developed at Noyafsky's institute, based on an initial prototype constructed by a group of researchers at MIT. Liza had found the prototype somewhere in a backwater program base. The experiment generator had been tailored to the capabilities of the apparatus in the* automated laboratory *at Noyafsky's institute. The program contained detailed knowledge on the equipment, copied from the suppliers' databases, and strategic knowledge for setting up an experiment. Especially the latter kind of knowledge was a strong property of the program; colleagues called the capability to rapidly generate high-quality experiments the gold mine of Noyafsky's institute.*

Noyafsky ordered the experiment generator to investigate the process of terminated transcription. The program analyzed the knowledge structure and tried, with the help of the reasoning mechanism, to think of situations in which the role of the trpL gene could be elucidated. A series of possible experiments appeared on the screen and after some discussion Noyafsky decided to send the second option to the laboratory. They would compare two E. coli strains. The first strain had a non-functional repressor protein, whereas the second strain had both a non-functional repressor and a mutation in the trpL gene. The modified regulation model of Fig. 6 *predicted that at a given trp concentration the trp-mRNA concentration in the double mutant E. coli strain would be higher as a result of disabling both the common repressor mechanism* and *the additional repressor mechanism, allowing transcription to continue undisturbed across the trpL gene (Fig.* 8(a) *and* (b)). *In the single mutant E. coli strain, on the other hand, the latter mechanism was still operational, suppressing trp-mRNA production to a certain extent.*

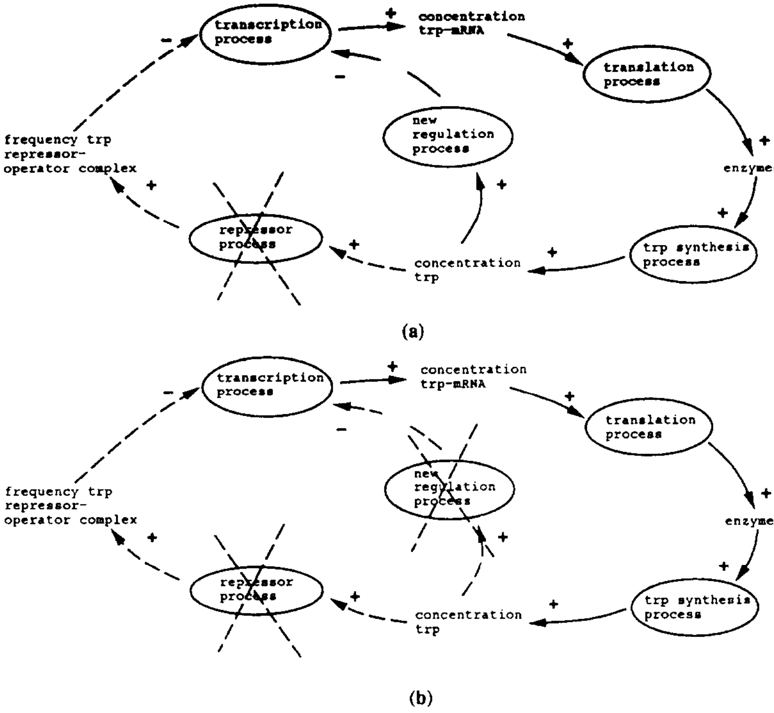

Fig. 8. Global qualitative model of the regulation process of the trp operon modified according to the experimental conditions. In (a) only the repressor process has been disabled, whereas in (b) the repressor process as well as the new regulation process have been disabled. For the notation, see Fig. 3. The dotted arrows indicate that these relations no longer matter, because the processes they connect have been disabled.

Noyafsky was delighted when four hours later the laboratory tests were finished and a positive result was reported. As a matter of fact, the strain in which both regulation mechanisms had been disabled did produce considerably more trp-mRNA than the strain with an intact trpL gene. This was an important step forward.

The next question Noyafsky put to himself was whether the additional regulation mechanism proposed by them occurred in other organisms as well. That would make their results even more plausible. Could they find examples, somewhere in that enormous Regulation Processes Database containing millions of experiments, of organisms with a regulation mechanism deviating from the usual repressor mechanism? It would have been like looking for a needle in a haystack, even with their sophisticated search engines, *if Neil Sandorf had not remarked pensively that he had once read something about problems with the regulation of the operon of S. typhimurium. Liza immediately selected the data on these experiments from the RPD and let the modified regulation model give a qualitative interpretation. The results were encouraging, in any case better than those obtained with the old regulation model.*

It looked fine and this feeling was strengthened by the evaluation program, *which had been developed by the* Journal of Molecular Biology *in order to rate submissions. The evaluation program rated theories on regulation processes on a number of carefully* *chosen evaluation criteria. The criteria were specified by members of a committee of molecular biologists and computer scientists which was installed by the American Association of Molecular Biologists, as a guarantee that the criteria were subscribed to by the majority of the community of molecular biologists. The criteria were in the form of explicitly formalized and implemented procedures operating upon the knowledge structure.*

On three important criteria the new model obtained a score considerably higher than the old model: a larger number of observations was successfully explained, the revision of the model was remarkably simple, and judging from the long list of suggested new experiments it seemed to offer opportunities for fruitful further research. A high score on these criteria increased the chance that their results would be published in a major journal, so there was every reason to be satisfied: the idea had turned out to be a good idea.

Now others had to be convinced that it was a good idea. Noyafsky glanced at his watch. If they would work hard, the article could be on the desk of the editor of Nature *tomorrow morning, so that it could be included in tomorrow's edition. They quickly outlined the structure of the article and allocated parts of the writing to each of the group members. They had to find a suitable name for the new regulation mechanism: "trpL mechanism" was too specific and "transcription termination mechanism" broke their tongues. Liza's proposal was more elegant and appropriate: attenuation mechanism. The mechanism did not completely block gene expression, like the usual repressor mechanism did, but attenuated the expression depending on the trp concentration in the cell.*

Fortunately, they could write the article rather quickly, because recently an automated research logfile *was made available, in which all calls to and results of computer programs were registered. Making a table with laboratory results was now simplified to retrieving data from the research logfile. Moreover, the section on research methods could be quickly composed by downloading from the on-line logfile the general characteristics and parameter values of the discovery programs and the knowledge bases and databases used. The article would be on its way to* Nature *by the end of the day.*

Scenario Fragment 3 illustrates some infrastructural requirements of competent performance in the computer-supported discovery environments of the future. The task of maintaining an infrastructure to facilitate the operation of computer tools is less straightforward than it might seem at first sight. For the proper management of a large-scale, shared database, for example, one needs not only hardware, software, and technical and scientific staff, but also an elaborated network of scientists who use and contribute to the database, scientific advisors, and financiers. These participants to the shared database have to meet regularly in order to coordinate their efforts and they have to come to agreements in the presence of sometimes conflicting viewpoints on how to structure and release the data. In the scenario we see how scientists organize a special workshop to harmonize the overlapping parts of the ontologies underlying the Genetic Regulation Processes knowledge base and the Polymerase knowledge base. Institutional arrangements like these have already become standard practice in the case of a number of real-life, shared scientific tools, such as large sequence databases [4] or taxonomic databases [24] in biology.

The importance of organizing and maintaining an infrastructure can be recognized if one realizes how many people are involved in a discovery produced in a computer-supported discovery environment. If the SYNGEN program discussed in [23] designs a new, efficient synthesis route to a particular compound, then quite a few people have contributed to this result. Apart from the scientists using the tool, think of the people who built the program, the local technical staff installing and maintaining the program, the management and staff of the reaction databases from which SYNGEN retrieves its information, the chemists contributing reactions to these databases. Discovery programs may relieve human scientists of some discovery tasks, but at the same time they create new tasks and new divisions of labor.

Scenario Fragment 4 highlights another aspect of competent performance in the discovery process. We see how Noyafsky intervenes in the operation of QualChem and decides to focus on the unknown function of the trpL gene. He guides the construction of alternative regulation models by acting on the basis of a hunch about the importance of a particular functionless component in the original regulation model. In the scenario this intervention turns out to be a profitable move, since it directly leads to an interesting hypothesis that deserves further testing (see Fragment 5).

More generally speaking, the performance achieved by means of the tools in a computer-supported discovery environment is determined by the competence of scientists who decide how to use the tools in the search process. The scientists decide which problems merit attention, they coordinate and adjust the operation of several programs, each exploring the search space belonging to a subproblem of the original problem, and they interpret the outcomes of the search process. In addition, they are on the alert to spot unusual intermediary results, which may lead them to ask new questions and change the course of their investigation. Such creative shifts require a scientist to be "prepared", to have the competence to see when an interesting deviation occurs, otherwise serendipity will be lost [2].

The longer Fragment 5 shows a social aspect of competent performance in a computer-supported discovery environment. Noyafsky realizes that his fellow scientists have to be convinced that the idea to extend the model with a new regulation mechanism is a good idea. The knowledge claims generated in a computer-supported discovery environment have to find their way to wider audiences and become accepted there to deserve the predicate "discovery" [7]. Presupposed in all the actions of scientists in a discovery environment is the wider research practice in which they work and where they have to justify and defend the knowledge claims generated with the help of their computer tools. The computer tools enhance competent performance, but do not replace the practice in which the competence of the performance is determined.

These examples illustrate an important point: if we want to enhance competent performance through the development of computer-supported discovery environments, we have to realize that the successful use of databases, knowledge bases, and discovery programs presupposes a research practice with a division of labor and competent scientists who handle the tools and evaluate the outcomes of the search process. As a consequence, we cannot understand a computer-supported discovery environment as a technical system only; we have to pay attention to how it is embedded in a research practice. Borrowing a term from organization theory [62], we can characterize a computer-supported discovery environment as a *sociotechnical system*. When a system is analyzed as a sociotechnical system, emphasis is laid on the interdependencies between technical equipment and the (groups of) people using this equipment. An advantage of this concept is that it allows recognition that the technical and social aspects of a discovery environment cannot always be clearly distinguished. For instance, the geographical structure of a distributed knowledge base is a technical characteristic of the discovery environment, but it also represents a social organization of work in the sense that different research groups are responsible for parts of the knowledge base.

More important for our present purposes is the notion, carried by the concept of sociotechnical system, that in order to enhance competent performance developers of discovery tools should not only be concerned with such technical aspects as the construction of suitable algorithms, but should also pay attention to other aspects of the design, in particular how the tools in a computer-supported discovery environment are to be organized in a research practice. If these aspects are neglected, discovery programs and other supporting tools will fall short of the promise voiced by Allen Newell and others, a promise which has been brought to life in the scenario.

These considerations have immediate implications for the construction of computer-supported discovery environments. In Section 5 we will articulate these implications in the form of a number of guidelines that developers can follow to integrate their tools in scientific practices. But in order to motivate these guidelines, we must first review what progress computer scientists and scientists in other disciplines have made in the direction of discovery environments like the one envisioned in the scenario.

4. Computer-supported discovery environments: state-of-the-art

For many of the tools encountered in the scenario, research prototypes have already been developed and some are even used by scientists on a regular basis. In this section, we review progress towards computer-supported discovery environments. First we describe discovery tools, roughly divided into those that arise from a systematization of the activities of scientists and those that arise from a systematization of the domain knowledge in a research practice (Fig. 1). Then we pay attention to the integration of the tools in scientific discovery environments.

Our focus is on computer tools that perform more sophisticated and knowledge-intensive types of reasoning, like hypothesis construction, theory revision, law induction, and concept formation. These systems are often labeled *discovery programs* or *discovery systems*, and are primarily found in the AI literature and the literature on advanced computer applications in various branches of (natural) science. Thus, we deliberately leave out a discussion of other efforts at automating scientific activities: statistical analysis, combinatorial optimization, numerical simulation, etc. This is not to say, of course, that computer support for these activities could not be extended into discovery programs. A nice example of work in this direction is the AIDE system, which assists human statistical analysts by planning, executing, and controlling exploratory data analysis tasks [58].

Existing discovery programs fall into two categories [14].[4] First, there are programs that *simulate* historical discovery processes, often (but not necessarily) focusing on the cognitive processes of scientists. Well-known examples of simulation programs can be found in the BACON family, consisting of programs for finding quantitative and qualitative relations in experimental data [36,53]. Other examples of simulation programs are AM [38], PI [60], ECHO [48,61], BR3 [32], KEKADA [34,35], COAST [50], HYPGENE [25,26], and ReTAX [1].[5]

The computational simulation of scientific discoveries is an interesting goal in itself, and one which may raise deep philosophical questions (see [57] and the reactions in the same journal issue). However, for (computer) scientists its main interest lies in the eventual application of the principles and strategies embodied in the programs to *new* problems in *current* research practices. By carefully simulating historical discoveries one may hope to explicate sufficiently general methods of inquiry which can then be transferred to existing problems. This program is for instance clearly stated in [25].

Second, there are discovery programs primarily concerned with providing useful support to scientists in their work. The developers, often computationally oriented domain scientists, take an *engineering* perspective towards the tasks they have to automate. They analyze a problem that scientists in a particular domain are confronted with and then draw suitable methods and techniques from their toolbox to design a computer program which performs the task or helps making the task manageable. A few examples of discovery programs with this orientation are (Meta-)DENDRAL [8,39], MECHEM [63,67], RX [5], SYNGEN [23], GOLEM [30,59], 49er [43], and IMEM [11,12].

Whichever of the two approaches is followed, in the end the acid test for any discovery program is of course its assimilation in the discovery environment of scientists. The important question to ask is: do scientists use the program to generate new knowledge claims that are recognized by a scientific community as discoveries? Given the amount of work invested in the development of discovery programs, the results to date are somewhat disappointing. Although we have quite a few simulations, extensions of the toolbox of AI techniques, and working prototypes, programs that form a self-evident component of scientific discovery environments are rare. Only a few programs have helped make a real contribution to the body of knowledge in a particular domain, a contribution worthy of being published in a refereed journal in that domain. Those exceptions include Meta-DENDRAL in mass spectroscopy [9], MECHEM in chemistry [64] and PAULI in particle physics [68], and GOLEM in pharmaceutical chemistry [31]. Although the current discovery environments of molecular biologists contain tools that incorporate AI techniques, the core business of research, making discoveries, is seldom covered.

The main reason that the breakthrough of discovery programs has not yet occurred must be sought in the difficulty of constructing interesting problem spaces that can be

[4] See [27] for a similar classification. Our distinction between the simulation and engineering approach towards discovery programs owes much to the distinction in [40] between AI programs that try to emulate human problem-solving processes and programs that perform complex tasks irrespective of the way human beings perform them.

[5] The program COAST differs from the others in the list, in that it does not model a well-known historical achievement, but a piece of common-sense scientific problem solving.

explored by the programs. The construction of a problem space in a computer-supported discovery environment requires large amounts of data and knowledge to be available in electronic form. Think only of the computerized data and knowledge sources that Noyafsky and his team drew from when they formulated a problem space for finding an alternative regulation mechanism: laboratory data from Paris, the GRP knowledge base, the QualChem program made available by the American Chemical Society, etc.[6] The computerization of knowledge in scientific domains is however still in its infancy. By far the greatest parts of the bodies of knowledge have not yet been formalized and included in databases and knowledge bases, and the parts that have are often difficult to access and integrate. As a consequence, the flexible construction of problem spaces—assumed in the scenario—is not possible in current scientific discovery environments. This rules out the routine use of discovery programs and, even more important, it limits the further development of systematic methods of discovery.

The difficulty of constructing search spaces will be alleviated when the current efforts in building large-scale databases and knowledge bases are carried through successfully. The discipline of molecular biology has witnessed a dramatic increase in the number of scientific databases and knowledge bases, and according to the latest estimates more than a hundred now exist [27]. Databases predominate (such as the GenBank sequence database that we mentioned above [4]), but from a long-term perspective the development of specialized, richly structured knowledge bases is even more important for computer-supported discovery. One example is the ColiGene knowledge base [51], which contains knowledge about genetic regulation mechanisms in the *E. coli* bacterium. Other examples of scientific knowledge bases are EcoCyc, which gathers information about the genes and metabolic pathways in the *E. coli* bacterium [28,29], and GeneSys, which contains information about structure-function relationships in gene expression [49].

In addition to the domain-oriented programs, computer scientists have attempted to formulate more general principles for building large-scale (scientific) knowledge bases (see [19,42] for overviews). In particular, they have propagated the separate development of domain ontologies: logical theories which give an explicit, partial account of a conceptualization of domain knowledge [22]. The logical theories partially define the contents of knowledge bases in a formal and implementation-independent way, facilitating the construction and maintenance of a knowledge base, the sharing and reuse of knowledge [41,45], and the comparison of information provided by different knowledge bases.

The increasing availability of databases and knowledge bases enhances the capability of scientists to construct problem spaces and explore them with their discovery programs. The development of more databases and knowledge bases alone is not enough, however. In order to create computer-supported discovery environments, we not only have to

[6] Once a problem space has been constructed, the amount of knowledge necessary to produce an interesting finding may seem modest, as for example noted by Valdés-Pérez [65]. However, as discoveries often result from making *unexpected* combinations of heterogeneous resources, one cannot tell beforehand which are needed. As a consequence, a whole range of computerized data and knowledge sources should be available, from which maybe only a few elements are picked in actually constructing a new problem space.

install more technical components, but we also need to pay attention to such questions as the organization of the cooperative development of shared databases, the compatibility of different knowledge bases, and procedures to guarantee the validity of discovery programs. In other words, we have to take into account that a computer-supported discovery environment is a sociotechnical system.

Unfortunately, the awareness that individual computer tools have to be combined into a discovery environment involving a host of technical and organizational complications, is not often found in the efforts discussed above. With some exceptions (e.g., [39,67]), developers of discovery programs do not show much interest in how their programs might be integrated into the discovery environment of scientists and how they could include this prospective use in the design of the programs. Developers of large-scale scientific databases and knowledge bases have recognized the importance of issues surrounding the cooperative development and use of shared resources (e.g., [4,10,16,51] in molecular biology), but this has not led to clear, domain-independent guidelines which developers can follow in making their tools an integrated part of computer-supported discovery environments.

In the next section, we will formulate a set of such guidelines, based on our analysis of how computer tools function in the context of discovery environments in research practices.

5. Guidelines for embedding computer tools in scientific discovery environments

It is important to emphasize that the guidelines to be presented are not cook-book recipes for building or improving program *x*; rather, they point at issues that should be addressed when integrating discovery programs and other computer tools in the everyday work of scientists. The issues bring a number of interwoven questions to the fore, some of which one would call technical, others social. Straightforward, general answers to these questions are hard to give, since both the questions and the answers are to a large extent determined by the contingencies of the practices in which the discovery tools have to find a place. However, examples taken from existing scientific discovery environments and the discovery environment sketched in the scenario will suggest how the guidelines could be put to work.

Guideline 1. *Devise technical and social measures to make discovery tools accessible to scientists, so as to prevent exclusion of potentially interested researchers and infringement of intellectual property rights.*

Noyafsky and his team gained access to a wide range of (shared) resources and employed these tools to build a problem space which they explored in search of a satisfying regulation mechanism. The possibility to construct and explore problem spaces reflects a powerful capability of computer-supported discovery environments. It also presupposes something which seems so trivial that it is easily overlooked, namely that the computer tools in a distributed discovery environment be *accessible* to scientists. The databases, knowledge bases, and discovery programs should not only be available to researchers who wish to use them in tackling a problem, but also to their peers who need to be able to check and evaluate a result obtained by means of a discovery program. In order to ensure that interactions transcending local practices remain possible and productive, entrance to computer-supported discovery environments must be open to interested members of the relevant research community.

Technically, the accessibility of tools in a computer-supported discovery environment is easy to accomplish. One could simply connect them to the Internet, so that they can be manipulated from a local workstation. Examples of such tools are the sequence database GenBank [4] and the sequence analysis server Grail (referred to in [13]), which receive and answer requests by e-mail. However, the social aspects of making discovery tools accessible to scientists may be more difficult to arrange. Special organizations or branches of existing organizations may need to be created, like the GRP in the scenario, who maintain and control the tools and guarantee that they are (against nominal charges) accessible to the researchers in a scientific community.[7] These organizations should not only further the release of tools for use and inspection; they should also set up social mechanisms for the protection of the developers of tools. A discovery program can be viewed as an important finding in its own right, a new scientific technique embodied in an executable mathematical structure. The phrase in the scenario "... colleagues called the capability to rapidly generate high-quality experiments the gold mine of Noyafsky's institute" illustrates how the availability of powerful discovery programs provides a competitive advantage to scientists. If they feel that their claims to originality are insufficiently protected, scientists will be reluctant to release their tools.[8]

Guideline 2. *Design computer-supported discovery environments and individual discovery tools in a modular fashion with components that can be developed, used, and maintained independently from other components.*

Successful work in computer-supported discovery environments is often based on new and original combinations of discovery tools: applying a particular data mining algorithm to one of several databases, for example, or guiding search in a well-known hypothesis space by heuristics imported from another domain. In Fragment 2 we encountered the successful application of the hypothesis generation program QualChem to the regulation models in the GRP knowledge base. Such new combinations cannot be foreseen by the developers of knowledge bases, hypothesis generation programs, and other discovery tools, so they should design their applications as much as possible as self-contained modules that can be connected in the discovery process.

The emerging computer-supported discovery environments in molecular biology show a *modular organization* of loosely coupled research tasks and knowledge sources that can be considered a systematization of an already existing, implicit or explicit modularity in the body of knowledge and search practices. The organization of discovery

[7] A real-life example of such an organization is the National Center for Biotechnology Information (NCBI), which is responsible for producing and distributing the GenBank sequence database.

[8] Compare [69], which describes how inventors and industrialists in 19th century Germany were reluctant to release the chemical structure of new synthetic dyes, because their inventions were insufficiently protected by patent law.

tools is grafted upon established subdivisions of biological entities (organisms, proteins, nucleotides) and research tasks (gene recognition, protein structure determination). The machine learning applications discussed by Craven and Shavlik [13] function as separate modules, coupled to the sequence databases from which they derive their input.

Giving a computer-supported discovery environment a modular structure has the additional advantage that it facilitates the division of development and maintenance tasks. Different modules can each be built and maintained by specialists in the domain of expertise captured by the module. The design principle of modular structure can profitably be applied to individual tools as well. The development of a large knowledge base with qualitative models of regulation mechanisms could, for example, be distributed over research groups with specializations in different classes of regulation mechanisms. The cooperative development of tools in a discovery environment, possibly involving the scientific community as a whole, may be the only answer to the problem observed by the makers of DENDRAL that "[a] surprisingly large amount of specialized knowledge is necessary to achieve expertise in even a very circumscribed field" [39, p. 251].

Guideline 3. *In developing discovery tools try to adhere to existing standards or try to become involved in efforts directed at the establishment of standards.*

Modularity is not enough to enable researchers to integrate the tools in a computer-supported discovery environment. If Noyafsky had wanted to investigate the binding of an active repressor to the operator region in the trp regulation process in more detail, he should have been able to add a module with detailed knowledge about the molecular structure of both the repressor and the operator region and a module with knowledge about chemical bonding. *Standardization* of the modules is necessary to ensure their compatibility on a conceptual and an implementational level. The standards concern for example the interfaces between different modules and the representation of data and knowledge in the modules. Standardization is a hot issue in the field of DNA sequence databases, because researchers would like to be able to share and integrate the information contained in different databases.

Achieving standards that are actually followed by tool developers in discovery environments is a process that involves more than finding solutions for technical problems. Proposals for standards have to be made by developers and users from the scientific communities sharing the tools in a discovery environment. These standards have to be discussed and sanctioned by a representative body of researchers, then they must be circulated, adapted and occasionally adjusted. In scenario Fragment 3 we saw how scientists organized a special workshop to this end, in which they discussed the connection between two knowledge bases. This procedure is not uncommon among developers of molecular biological databases (see [27]).

Guideline 4. *Make discovery tools interactive by building in decision points that can be influenced by users through an advanced user-interface.*

Competent performance, leading to new and accepted discoveries, is located in a research practice of scientists who use the tools in their discovery environments. They decide which problems to tackle, they choose discovery tools that fit the problem, and they exercise control over the automated search process. This implies that discovery tools should be made *interactive* by equipping them with intelligent, user-friendly interfaces that enable the scientist to intervene at particular decision points (see also [39]). It was the graphical presentation of the simulation of the trp regulation process that suggested to Noyafsky to focus on the trpL gene and the user-interface subsequently allowed him to guide the program in that direction (Fragment 4). The active role of the user in the discovery process, and the control and presentation issues flowing from it, do not usually receive much attention in the published accounts of discovery programs.

Guideline 5. *Develop discovery tools that are flexible, in the sense that they can be easily adapted by the users of the tools.*

Work in a computer-supported discovery environment consists in the construction of a problem space and the traversal of this space in search of an interesting result. At a particular moment in the search process, scientists may feel the need to jump to another, related problem space or to explore the existing problem space in a completely different way. In order to switch to another problem space, scientists have to change the configuration of tools involved in formulating the problem space or even change the tools themselves. Flexibility of the tools in a computer-supported discovery environment becomes even more important if one remembers the occasional occurrence of scientific revolutions in which search practices and the conceptual order in a discipline radically change. If it is possible to modify the tools in a discovery environment without much effort, we will say that they are *flexible*.

An example of a flexible tool is a biological knowledge base that can be easily extended by refining the granularity of its ontology, so that at a later stage the chemical composition of biological entities can be added without affecting the existing properties and relations among the entities. Such a change in granularity is visible in MECHEM, where the addition of structural information on the reactants and products in a chemical reaction supplies further constraints for reducing the number of conjectured reaction mechanisms [67]. Another example is a hypothesis generation program in which the order of subactivities or the search strategy can be modified. In the scenario, we encountered flexible tools when the opportunity to adapt copies of centrally maintained regulation models to local circumstances was referred to (Fragment 2). Developers of molecular databases try to build flexibility into their data models by defining general concepts and relations which cross a wide range of domains, and which are, they expect, relatively stable [44]. Taking such a unifying conceptual framework as a starting point, appropriate specializations of the model can be formulated for each domain.

The required flexibility of discovery tools makes high demands on the design and the implementation of the tools. Developers could support flexibility by making a clear design, using a powerful and widely used programming language, and providing extensive documentation with the tools, features exemplified in [44].

Guideline 6. *Make the discovery tools transparent to others by clearly and unambiguously specifying how they have been designed and used in the search process.*

A precondition for the flexibility of a tool in a computer-supported discovery environment is its clear design, i.e., the *transparency* of the tool. If a theory revision program and an experiment generator are used by a scientist to generate a discovery claim, then her colleagues should be able to understand how the programs arrived at their conclusions in order to assess the validity of the claim. They should be able to inform themselves of the decisions that have been taken in designing the computer tools and the way in which these decisions influence the construction and exploration of problem spaces. If a data mining system unexpectedly concludes that there is a statistically significant relation between two rare illnesses, this claim will meet with a priori skepticism when no details are given about the database that was used. After all, the relation may be spurious due to the fact that the entries in the database are heterogeneous or even unreliable (for these and other problems, see [17]).

The transparency of the components of a computer-supported discovery environment can be increased by describing the contents of databases and knowledge bases and the activities of discovery programs on a conceptual level. The decisions made by developers and users of the tools should not remain hidden in the program code, but explicated in flow charts or ontologies, so that others can assess the discovery claims produced by means of these tools.

Guideline 7. *Promote the critical discussion of the design and use of discovery tools as part of the regular quality control mechanisms in a scientific domain, or help create new fora to this end.*

The actual evaluation of discovery claims by a scientific community requires that the programs be *subjects of discussion*. A forum should be created for discussing, criticizing, and certifying the discovery tools and the manner in which they have been used.

In local research settings, joint review of decisions that have been taken in developing and using discovery programs should become part of the normal team discussions in a research project. On the level of a scientific community, critical examination of discovery tools should be included in quality control mechanisms like peer review for scientific journals and (plenary) discussions during conferences. Alternatively, new fora could be established, such as special sessions during conferences or workshops, or a scientific organization controlling and maintaining large knowledge bases and libraries of discovery routines. The scientists participating in these fora are charged with the task of evaluating the quality of knowledge claims and discovery programs. Their judgment will decide whether the use of a program is deemed acceptable in a particular context and whether the new findings thus obtained are admitted to a shared knowledge base. In the scenario, we encountered such institutions as a special meeting at a workshop to bring the contents of two knowledge bases in agreement and the Supervisory Committee of the RPD which irritated Noyafsky with its bureaucratic style (clearly the scenario is not a utopia).

Guideline 8. *Contribute to the development of quality control criteria for the evaluation of discovery tools and their use in search processes.*

A critical evaluation of discovery programs and the claims generated by them presupposes the availability of standards and criteria to guide scientists in passing their judgment. In considering Noyafsky's claim they could for instance ask questions like: Are your qualitative regulation models detailed enough to capture all relevant factors in this situation? Are your hypothesis operators complete in the sense that minor modifications that may also solve the anomaly are not overlooked? Is your qualitative reasoning mechanism sound, so that testing the adequacy of proposed changes to the models, or proposing useful experiments on the basis of these changes, does not lead to invalid results?

Quality control criteria helping scientists to scrutinize knowledge claims generated in computer-supported discovery environments have not yet been formulated. In the scenario, however, they play an important role. Noyafsky and his colleagues realize that the regulation model put forward will be critically analyzed and evaluated. They anticipate on the reviewing process by including details on the discovery tools in their article and by having their regulation model rated by a special evaluation program of the *Journal of Molecular Biology* which considers, among other things, its consilience, simplicity, and fruitfulness.

6. Conclusions

The purpose of the guidelines is to assist developers of discovery tools in integrating their programs in the search practices of working scientists. The guidelines identify issues that have to be addressed if the current generation of discovery programs is to grow into the powerful computer-supported environments that we have described in the scenario. What is essential for the realization of such computer-supported discovery environments is the recognition that we are building large-scale sociotechnical systems. We do not only have to solve technical problems that concern the task performance of individual tools, but we also have to address a range of social and institutional problems that concern the organization of these tools in a research practice. The guidelines single out some of these problems and suggest developers of discovery tools how they could anticipate them in the design of their programs. In solving these problems, system developers have to enter into close cooperation with the users of their discovery tools, a point that has been made before in the field of discovery systems [39,67].

By locating discovery programs and other tools in computer-supported discovery environments, we are able to see how the systematization and computerization of research practices may dramatically change the aspect of science. If we restrict our focus to new tools deriving from the application of techniques in AI and other branches of computer science, we are apt to miss the import of these changes. New methods, criteria, and skills for doing research need to accompany the inclusion of the sophisticated computer tools in scientific discovery environments. This may result in a far-reaching transformation of science, ranging from the means enabling the individual scientist to make discoveries to the organization of scientific practices and communities. The computer-supported discovery environments effecting these transformations cannot be designed in a straightforward way. But one can develop pictures of what they could be like and how they would be integrated in research practices—as we did in our scenario. And one can take the emergence of new skills, standards, organizations, and institutions into account when developing tools—as we indicated in our guidelines.

Acknowledgements

The authors would like to thank P. Ahrweiler, B. de Jong, N. Mars, J. Nijhuis, W. ter Stal, G. van der Linden, P. van der Vet, and three anonymous reviewers for their contributions to this article.

References

[1] E. Alberdi and D. Sleeman, Taxonomy revision in botany: a simulation of historical data, in: R.E. Valdés-Pérez, ed., *Systematic Methods of Scientific Discovery, Papers from the 1995 AAAI Symposium* (AAAI Press, Menlo Park, CA, 1995) 99-104.

[2] B. Barber and R.C. Fox, The case of the floppy-eared rabbits: an instance of serendipity gained and serendipity lost, in: B. Barber and W. Hirsch, eds., *The Sociology of Science* (Free Press of Glencoe, New York, 1962) 525-538.

[3] G.I. Bell and T.G. Marr, *Computers and DNA* (Addison-Wesley, Redwood City, CA, 1990).

[4] D.A. Benson, M. Boguski, D.J. Lipman and J. Ostell, GenBank, *Nucleic Acids Res.* **22** (1994) 3441-3444.

[5] R.L. Blum, *Discovery and Representation of Causal Relationships from a Large Time-Oriented Clinical Database: The RX Project* (Springer, Berlin, 1982).

[6] D.G. Bobrow and P.J. Hayes, Artificial Intelligence—where are we?, *Artif. Intell.* **25** (1985) 375-415.

[7] A. Brannigan, *The Social Basis of Scientific Discoveries* (Cambridge University Press, Cambridge, 1981).

[8] B.G. Buchanan and E.A. Feigenbaum, Dendral and Meta-Dendral: their applications dimension, *Artif. Intell.* **11** (1978) 5-24.

[9] B.G. Buchanan, D.H. Smith, W.C. White, R. Gritter, E.A. Feigenbaum, J. Lederberg and C. Djerassi, Applications of artificial intelligence for chemical inference, XXII: automatic rule formation in mass spectroscopy by means of the Meta-Dendral program, *J. Am. Chem. Soc.* **96** (1976).

[10] C. Burks, The flow of nucleotide sequence data into data banks: role and impact of large-scale sequencing projects, in: G.I. Bell and T.G. Marr, eds., *Computers and DNA* (Addison-Wesley, Redwood City, CA, 1990) 35-45.

[11] D. Conklin, S. Fortier and J. Glasgow, Knowledge discovery in molecular databases, *IEEE Trans. Knowl. Data Eng.* **5** (1993) 985-987.

[12] D. Conklin, S. Fortier and J. Glasgow, Knowledge discovery of multilevel protein motifs, in: R.B. Altman, D.L. Brutlag, P.D. Karp, R. Lathrop and D. Searls, eds., *Proceedings Second International Conference on Intelligent Systems for Molecular Biology, ISMB-94* (AAAI Press, Menlo Park, CA, 1994) 96-102.

[13] M.W. Craven and J.W. Shavlik, Machine learning approaches to gene recognition, *IEEE Expert* **8** (1994) 2-10.

[14] H. de Jong, Ontdekkingssystemen in de wetenschap: een exploratie van computer-ondersteunde ontdekkingsomgevingen, Master thesis, Philosophy of Science, Technology, and Society, University of Twente, Enschede (1994).

[15] R.F. Doolittle, What we have learned and will learn from sequence databases, in: G.I. Bell and T.G. Marr, eds., *Computers and DNA* (Addison-Wesley, Redwood City, CA, 1990) 21-31.

[16] J. Euzenat, A protocol and distributed architecture for building consensual knowledge bases, in: N.J.I. Mars, ed., *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing 1995* (IOS Press, Amsterdam, 1995) 143-155.

[17] W.J. Frawley, G. Piatetsky-Shapiro and C.J. Matheus, Knowledge discovery in databases: an overview, *AI Magazine* **3** (1992) 57-70.

[18] P. Friedland and L.H. Kedes, Discovering the secrets of DNA, *Commun. ACM* **28** (1985) 1164-1186.

[19] K. Fuchi and T. Yokoi, *Knowledge Building and Knowledge Sharing* (IOS Press, Amsterdam, 1994).

[20] P. Galison, *How Experiments End* (University of Chicago Press, Chicago, 1987).

[21] D. Gooding, The procedural turn, or: why do thought experiments work, in: R.N. Giere, ed., *Cognitive Models of Science*, Minnesota Studies in the Philosophy of Science **15** (University of Minnesota Press, Minneapolis, MN, 1992) 45-76.

[22] N. Guarini and P. Giaretta, Ontologies and knowledge bases: towards a terminological clarification, in: N.J.I. Mars, ed., *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing 1995* (IOS Press, Amsterdam, 1995) 25-32.

[23] J.B. Hendrickson, Systematic synthesis design: the SYNGEN program, in: R.E. Valdés-Pérez, ed., *Systematic Methods of Scientific Discovery, Papers from the 1995 AAAI Symposium* (AAAI Press, Menlo Park, CA, 1995) 13-17.

[24] C. Hine, Representations of information technology in disciplinary development: disappearing plants and invisible networks, *Sci. Technol. Human Values* **20** (1995) 65-85.

[25] P.D. Karp, Hypothesis formation as design, in: J. Shrager and P. Langley, eds., *Computational Models of Scientific Discovery and Theory Formation* (Morgan Kaufmann, San Mateo, CA, 1991) 275-317.

[26] P.D. Karp, Design methods for scientific hypothesis formation and their application to molecular biology, *Mach. Learn.* **12** (1993) 89-116.

[27] P.D. Karp, New directions for scientific discovery research: applications, in: R.E. Valdés-Pérez, ed., *Systematic Methods of Scientific Discovery, Papers from the 1995 AAAI Symposium* (AAAI Press, Menlo Park, CA, 1995) 94-96.

[28] P.D. Karp and M.L. Mavrovouniotis, Representing, analyzing, and synthesizing biochemical pathways, *IEEE Expert* **8** (1994).

[29] P.D. Karp and M. Riley, Representations of metabolic knowledge, in: L. Hunter, D. Searls and J. Shavlik, eds., *Proceedings First International Conference on Intelligent Systems for Molecular Biology* (AAAI Press, Menlo Park, CA/MIT Press, Cambridge, MA, 1993) 207-215.

[30] R.D. King, D.A. Clark, J. Shirazi and M.J.E. Sternberg, Inductive logic programming used to discover topological constraints in protein structures, in: R.B. Altman, D.L. Brutlag, P.D. Karp, R. Lathrop and D. Searls, eds., *Proceedings Second International Conference on Intelligent Systems for Molecular Biology, ISMB-94* (AAAI Press, Menlo Park, CA, 1994) 219-226.

[31] R.D. King, S. Muggleton, R.A. Lewis and M.J.E. Sternberg, Drug design by machine learning: the use of inductive logic programming to model the structure-activity relationships of trimethoprim analogues binding to dihydrofolate reductase, *Proc. Nat. Acad. Sci.* **89** (1992) 11322-11326.

[32] S. Kocabas, Conflict resolution as discovery in particle physics, *Mach. Learn.* **6** (1991) 277-309.

[33] T.S. Kuhn, *The Structure of Scientific Revolutions* (The University of Chicago Press, Chicago, IL, 2nd ed., 1970).

[34] D. Kulkarni and H.A. Simon, The processes of scientific discovery: the strategy of experimentation, *Cognit. Sci.* **12** (1988) 139-175.

[35] D. Kulkarni and H.A. Simon, Experimentation in machine discovery, in: J. Shrager and P. Langley, eds., *Computational Models of Scientific Discovery and Theory Formation* (Morgan Kaufmann, San Mateo, CA, 1991) 255-273.

[36] P. Langley, H.A. Simon, G.L. Bradshaw and J.M. Żytkow, *Scientific Discovery: Computational Explorations of the Creative Processes* (MIT Press, Cambridge, MA, 1987).

[37] B. Latour and S. Woolgar, *Laboratory Life, The Construction of Scientific Facts* (Princeton University Press, Princeton, NJ, 2nd ed., 1986).

[38] D.B. Lenat, On automated scientific theory formation: a case study using the AM program, in: J.E. Hayes, D. Michie and L.I. Mikulich, eds., *Machine Intelligence* **9** (Ellis Horwood, Chichester, 1979) 251-283.

[39] R.K. Lindsay, B.G. Buchanan, E.A. Feigenbaum and J. Lederberg, DENDRAL: a case study of the first expert system for scientific hypothesis formation, *Artif. Intell.* **61** (1993) 209-261.

[40] N.J.I. Mars, *Inleiding Kennistechnologie* (Academic Service, Schoonhoven, 1991).

[41] N.J.I. Mars, *Proceedings ECAI Workshop on Knowledge Sharing and Reuse: Ways and Means*, Vienna (1992).

[42] N.J.I. Mars, *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing 1995* (IOS Press, Amsterdam, 1995).

[43] J.R. Moraczewski, R. Zembowicz and J.M. Żytkow, Geobotanic database exploration, in: R.E. Valdés-Pérez, ed., *Systematic Methods of Scientific Discovery, Papers from the 1995 AAAI Symposium* (AAAI Press, Menlo Park, CA, 1995) 76-80.

[44] *NCBI Software Development ToolKit, Version 1.9* (National Center for Biotechnology Information (NCBI), Bethesda, MD, 1994).

[45] R. Neches, R. Fikes, T. Finin, T. Gruber, R. Patil, T. Senator and W.R. Swartout, Enabling technology for knowledge sharing, *AI Magazine* **12** (3) (1991) 37-55.

[46] A. Newell and H.A. Simon, *Human Problem Solving* (Prentice-Hall, Englewood Cliffs, NJ, 1972).

[47] T. Nickles, Justification and experiment, in: S. Schaffer, D. Gooding and T. Pinch, eds., *The Uses of Experiment: Studies in the Natural Sciences* (Cambridge University Press, Cambridge, 1989) 299-333.

[48] G. Nowak and P. Thagard, Copernicus, Ptolemy, and explanatory coherence, in: R.N. Giere, ed., *Cognitive Models of Science*, Minnesota Studies in the Philosophy of Science **15** (University of Minnesota Press, Minneapolis, MN, 1992) 274-309.

[49] G.C. Overton, K. Koile and J.A. Pastor, GeneSys: a knowledge management system for molecular biology, in: G.I. Bell and T.G. Marr, eds., *Computers and DNA* (Addison-Wesley, Redwood City, CA, 1990) 213-239.

[50] S.A. Rajamoney, A computational approach to theory revision, in: J. Shrager and P. Langley, eds., *Computational Models of Scientific Discovery and Theory Formation* (Morgan Kaufmann, San Mateo, CA, 1991) 225-253.

[51] F. Rechenmann, Building and sharing large knowledge bases in molecular genetics, in: *Proceedings International Conference on Building and Sharing of Very Large-Scale Knowledge Bases (KB&KS'93)*, Tokio (1993) 289-301.

[52] A. Rip, New combinations, *Eur. Rev.* **3** (1995) 83-92.

[53] D. Rose and P. Langley, Chemical discovery as belief revision, *Mach. Learn.* **1** (1986) 423-451.

[54] S. Shapin and S. Schaffer, *Leviathan and the Air-Pump: Hobbes, Boyle, and the Experimental Life* (Princeton University Press, Princeton, NJ, 1985).

[55] H.A. Simon, P.W. Langley and G.L. Bradshaw, Scientific discovery as problem solving, *Synthese* **47** (1981) 1-27.

[56] M. Singer and P. Berg, *Genes & Genomes, A Changing Perspective* (University Science Books, Mill Valley, CA, 1991).

[57] P. Slezak, Scientific discovery by computer as empirical refutation of the strong programme, *Soc. Stud. Sci.* **19** (1989) 563-600.

[58] R. St. Amant and P.R. Cohen, Preliminary system design for an EDA assistant, in: *Proceedings Fifth International Workshop on Artificial Intelligence and Statistics* (1995) 502-512.

[59] M.J.E. Sternberg, R.A. Lewis, R.D. King and S. Muggleton, Machine learning and biomolecular modelling, in: K. Furukawa, D. Michie and S. Muggleton, eds., *Machine Intelligence* **13** (Clarendon Press, Oxford, 1994).

[60] P. Thagard, *Computational Philosophy of Science* (MIT Press, Cambridge, MA, 1988).

[61] P. Thagard and G. Nowak, The conceptual structure of the geological revolution, in: J. Shrager and P. Langley, eds., *Computational Models of Scientific Discovery and Theory Formation* (Morgan Kaufmann, San Mateo, CA, 1991) 27-72.

[62] E.L. Trist, The sociotechnical perspective: the evolution of sociotechnical systems as a conceptual framework and as an action research program, in: A.H. Van De Ven and W.F. Joyce, eds., *Perspectives on Organisation Design and Behavior* (Wiley, New York, 1981) 19-87.

[63] R.E. Valdés-Pérez, Conjecturing hidden entities by means of simplicity and conservation laws: machine discovery in chemistry, *Artif. Intell.* **65** (1994) 247-280.

[64] R.E. Valdés-Pérez, Human-computer interactive elucidation of reaction mechanisms: application to catalyzed hydrogenolysis of ethane, *Catalysis Lett.* **28** (1994) 79-87.

[65] R.E. Valdés-Pérez, Some recent human-computer discoveries and what accounts for them, *AI Magazine* **16** (3) (1995) 37-44.

[66] R.E. Valdés-Pérez, Generic tasks of scientific discovery, in: R.E. Valdés-Pérez, ed., *Systematic Methods of Scientific Discovery, Papers from the 1995 AAAI Symposium* (AAAI Press, Menlo Park, CA, 1995) 23-28.

[67] R.E. Valdés-Pérez, Machine discovery in chemistry: new results, *Artif. Intell.* **74** (1995) 191-201.

[68] R.E. Valdés-Pérez and M. Erdmann, Systematic induction and parsimony of phenomenological conservation laws, *Comput. Phys. Commun.* **82** (1994) 171-180.

[69] H. van den Belt and A. Rip, The Nelson-Winter-Dosi model and synthetic dye chemistry, in: W.E. Bijker, T.P. Hughes and T.J. Pinch, eds., *The Social Construction of Technological Systems* (MIT Press, Cambridge, MA, 1987) 135-158.

[70] C. Yanofsky, Attenuation in the control of expression of bacterial operons, *Nature* **289** (1981) 751-758.