



# Augmented Cognition for Bioinformatics Problem Solving

*Olga Anna Kuchar and Jorge Reyes-Spindola*

Pacific Northwest National Laboratory  
P.O. Box 999, 902 Battelle Blvd.  
Richland, WA, 99352  
{olga.kuchar, jorge.reyes.spindola}@pnl.gov

*Michel Benaroch*

Center for Creation and Management of Digital  
Ventures  
Martin J. Whitman School of Management Syracuse  
University, Syracuse, NY, 13244  
mbenaroc@syr.edu

## Abstract

We describe a new computational cognitive model that has been developed for solving complex problems in bioinformatics. This model addresses bottlenecks in the information processing stages inherent in the bioinformatics domain due to the complex nature of both the volume and type of data and the knowledge required in solving these problems. There is a restriction on the amount of mental tasks a bioinformatician can handle when solving biology problems. Bioinformaticians are overwhelmed at the amount of fluctuating knowledge, data, and tools available in solving problems, but to create an intelligent system to aid in this problem-solving task is difficult because of the constant flux of data and knowledge; thus, bioinformatics poses challenges to intelligent systems and a new model needs to be created to handle such problem-solving issues. To create a problem-solving system for such an environment, one needs to consider the scientists and their environment, in order to determine how humans can function in such conditions. This paper describes our experiences in developing a complex cognitive system to aid biologists in knowledge discovery. We describe the problem domain, evolution of our cognitive model, the model itself, how this model relates to current literature, and summarize our ongoing research efforts.

## 1 Introduction

Bioinformatics is the application of computational techniques to understand and organize the information associated with biological macromolecules. The aims of bioinformatics are three-fold: gather/organize data; analyze data; and interpret data. Gathering and organizing data accomplishes two key criteria. First, it allows researchers to submit new data as it is produced. Secondly, it allows researchers to access existing information. Even though the task of gathering and organizing data is important, the information stored in these databases is relatively useless until analyzed. Thirdly, analyzing data is another aim of bioinformatics; to aid in this data analysis, many tools and resources have been developed that focus on particular aspects of the analysis process. The development of such tools requires extensive knowledge of biology and computational theory. Interpretation of this analysis in a biologically meaningful manner is key to uncovering common principles that apply across many systems, and highlight features that are unique to some. Traditionally, biological studies examined individual systems in detail and compared them with a few systems that are related. When dealing with today's bioinformatics, a need to conduct global analyses of all the available data is prudent. The reader is directed to Luscombe et al. for a detailed description of bioinformatics and its domain (Luscombe, Greenbaum, & Gerstein, 2001).

Over the last several years, advances in biology and the equipment available have resulted in massive amounts of data dealing in the terabytes range; but data does not equate to knowledge – data must be processed and fused in order to discover new knowledge. Humans are very adept at taking heterogeneous data and discovering new knowledge – but even humans cannot easily discover new information in the sea of data that they are facing on a daily basis. For example, biological data is increasing at an unprecedented rate (Reichhardt, 1999). As of February 15, 2004, GenBank contains 44,575,745,176 bases from 40,604,319 reported sequences (“Growth of GenBank”, 2005). The amount of information stored in databases around the world continues to grow; for example, the SWISS-PROT database increased its entries by 1% within two weeks to total 170,140 sequence entries (“SWISS-PROT Statistics”, 2005). Add to this raw data the knowledge that has been published from scientists' experiments and projects over the years, and a bioinformatician has an enormous quantity and variety of information to harvest and fuse.

The most profitable research in bioinformatics often results from integrating multiple sources of data and placing their pieces of information into context with each other. Unfortunately, it is not always straightforward to access and cross-reference these sources of information because of differences in nomenclature and file formats. The bioinformatician spends most of his/her time in data searching and retrieval, followed by tool applications to find the applicability of these disparate pieces of information to each other and their problem at hand. This is usually a long and tedious process, lasting anywhere between a few days to several months of work (depending on the problem complexity). Thus, the human-processing bottlenecks are: memory (where to go and search for information for its is all distributed); learning (how to relate the disparate pieces of information together and what is the bioinformatician still missing); comprehension (understanding how to use the databases, their outputs, and other tools for their trade); and decision making (what does all this information mean).

So, bioinformatics poses a challenge to intelligent systems because of its constant flux of data and knowledge. There are many different aspects to this problem:

- Solving complex problems involves both implicit and explicit knowledge.
- Not all information is known to the expert at the time of solving the problem. The expert needs to search for additional information in order to solve the higher problem. This is not an easy task based on the massive data that scientists need to sift through.
- Solving these complex problems involves solving sub-problems, each of these being complex in nature and solution.
- Being able to fuse information at a sub-problem level does not necessarily solve the higher problem – knowledge compression must occur as we progress up the hierarchy of solution space.

In our opinion, current cognitive architectures are not robust enough to handle such real environments. In bioinformatics, a scientist is faced with such problems on a daily basis. Many tools have been developed to help the scientists with parts of their task, but no architecture has been designed to help the scientists in knowledge discovery – having the scientist spend more cognitive time in understanding solutions than in digging for information. This paper describes our experiences in developing a new cognitive system to aid biologists in knowledge discovery. The remainder of this paper is divided as follows: description of the problem that we are using for our research; description of the new cognitive model; research issues in implementing this new model; how this new model relates to other cognitive architectures; and a summary of our current research and future work.

## **2 Bioinformatics**

Our observation of bioinformatics cognitive functions centred on the processes involved in analyzing a biological experiment that studied the genetic effects of an endotoxin on mice brains after an induced cerebral stroke. The data output of the experiment was in the form of gene expression data originating from an array of DNA snippets known as a gene chip array. These gene expression experiments quantify the expression levels of individual genes and measure the amount of mRNA or protein products that are produced by a cell. In this particular experiment, the expression levels of mice treated with the endotoxin were compared against a control group treated with a saline solution. DNA samples were then taken at different time points for both groups before and after the induced stroke. It was observed that the mice treated with the endotoxin were able to resist the aftereffects of the cerebral stroke compared to the mice treated with the saline solution. The aim of the experiment is then to investigate what were the genes involved in the stroke resistance and what role did they play in the physiology of the endotoxin-treated mice.

This problem is fairly typical of the high-throughput experimental approach that produces large amounts of results. These experiments provide a “snapshot” of cellular events that can eventually be integrated into a dynamic view of cellular processes through time. In the analysis of this data, gene expression patterns are typically clustered into groups that define different behaviours. At the highest level, biologists want to understand why different types of genes are grouped together and they want to find out what proteins get expressed or what chain reactions of gene-protein-gene they unleash. However, at the most basic level, it is a significant obstacle just to know the identities of proteins in a group, let alone why they are grouped together. This basic protein list is sometimes referred to as the “Molecular Parts List”. The process of identifying protein members is referred to as “annotation”. Annotation is an important problem for scientists and is the example area that we will use to describe the development of our cognitive model.

To explore the biological meaning of gene expression array experiments, biologists need to address several key sub-problems. First, biologists need to determine what kinds of expression patterns are within their experiment by applying clustering algorithms to the gene expression data. Clustering approaches have been widely applied to this type of analysis (Eisen, Spellman, Brown, & Botstein, 1998; Halkidi, Batistakis, & Vazirgiannis, 2001; Luo, Tang, & Khan, 2003); however, clustering algorithms do not directly provide statistical confidence for clustered expression patterns. As a result, some biologists may triage the genes based on their statistical significance in differential expressions and confirm consistent expression patterns within replicates (Ross et al., 2000; Scherf et al., 2000). Next, biologists may need to determine the aliases of the genes and proteins represented in the gene array. One of the problems is that there are multiple protein names and multiple protein database records that need to be searched and correlated. Once the actors have been identified, other features need to be determined for each one, such as role assignment, domain assignment, superfamily analysis, and ortholog analysis. To determine these, different tools are applied and several databases need to be accessed, such as Similarity Box tool (Sofia, Chen, Hetzler, Reyes-Spindola, & Miller, 2001), and Pfam or SMART databases. Then, key features need to be extracted for each gene. Some of these features are conserved operon patterns, promoter elements, associated transcription factors and metabolic pathway assignments. Once the information is gathered and correlated on each gene, a biologist will start analyzing each group of genes and what are their common attributes. Further data harvesting is required to determine the biochemistry and molecular biology of the involved genes and the associated proteins. This involves both data and literature mining. Finally, a biologist may analyze what network predictions can be made about these genes (Park, 2002). Research in information theory and Bayesian methods assists biologists in this analysis. Overall, biologists must mine and fuse enormous amounts of distributed data by using many different tools to aid them in knowledge discovery. Typically there are tens of thousands of genes in one microarray experiment. The reader is directed to more detailed information ("Dipping into DNA Chips", 1999).

With respect to annotation, proteins fall roughly into three categories: (1) approximately one in ten can be fully and completely annotated with a high level of certainty; (2) a small fraction of proteins are cryptic and cannot be reliably attached to any information; and (3) the majority can be associated with only partial information. Current annotation systems tend to assign misleading names that are partially right but just as equally wrong. The assignments are static and do not provide any basis for evaluating their meaning or quality without repeating multiple bioinformatic searches. Annotation is still an issue for structural data as well, although the biology community has attempted to form a consensus as to what annotation of a structure is currently required.

We worked closely with several biologists and bioinformaticians at Pacific Northwest National Laboratory to determine how they each solve gene expression problems. We interviewed each of our colleagues separately and elicited knowledge from them. During our elicitation process, we created decision trees so that we could capture the expert's strategies and knowledge about how they decompose the problem into manageable parts, and then how they solved those sub-parts. We then transformed our decision trees into verbal protocols. Upon examination of our elicited knowledge, the methodology for solving the gene expression problem can be abstracted to the following:

1. Obtain an initial gene classification by performing clustering.
2. Determine common features of each group.
3. Look for over-represented transcription factors.
4. Search for particular features of transcription factors.
5. Based on these searches, identify important features and gene linkages.
6. Determine particular features of linked genes.

Rather than building an expert system to solve this sub-problem, we built a system that can be used in a dynamic environment and that uses generic methods to solve many different problems. This led us to develop a cognitive model that mimics human higher-level problem solving for challenging problems in complex and dynamic domains.

### **3 Cognitive Model**

A cognitive architecture specifies the underlying structure of an intelligent system that is constant over time and domain. A review of the recent flow of research in cognitive architectures can be found in (Langley & Laird, 2002). Current computational cognitive architectures are not robust enough to handle such dynamic environments as described in section 2. In bioinformatics, a scientist deals with such problems on a daily basis, but solving these problems requires anywhere between a few days to several months. Many tools have been developed to help the

scientists with parts of their task, but no architecture has been designed to integrate the cognitive aspects of complex problem solving and associated tools to help the scientists in knowledge discovery *i.e.*, have the scientist spend more cognitive time in understanding solutions than in digging for information. This is the focus of our research work. A human solves problems using the five senses and the brain. The basic functions and operations of the brain remain constant, but the knowledge that is accessed changes based on the problem that is being solved. The brain accesses different knowledge stored within its capacity. In this section, we describe our new cognitive model that is based on this theory. Based on our observations of higher-level problem solving, humans address problem solving using some key knowledge components. We will explain these components in terms of our problem domain.

### **3.1 Goal Knowledge**

Biologists know something about their goal – what it means to have solved the current problem. For example, biologists know that for gene identification, a solution must include a gene name, its chromosome, position within the genome (if available), etc. We define this knowledge as *Goal Knowledge*. Goal knowledge represents a definition of what a solution to a particular problem may contain. For example, a solution to a gene expression problem may contain information about genes, proteins produced, transcription factors, etc. This sort of information can be captured as an entity-relationship diagram that depicts what a person knows about the goal that they are trying to solve. This needs to be in a form that can be expanded by a user (a non-programmer) as new information about solving such a problem is evolving. For example, in future gene expression problems, a biologist may know that the goal of solving this problem would involve a gene's position within the chromosome and they would need to add it to the goal knowledge. As humans, the way we solve certain problems today is probably not the way we solved those problems a decade ago. Since technology is growing and our knowledge and abilities are increasing, our methodologies are also changing and providing such a capability is important. We need to represent such dynamics in an easy fashion so that a user is not dependent on programmers to capture these changes.

### **3.2 Strategic Knowledge**

While solving a problem, biologists have different strategies or heuristics that they follow to get closer to a potential solution. These strategies are not always perfect or lead to the desired solution, but they do recognize a path towards solving the problem. We define this as Strategic Knowledge. Strategic knowledge is knowledge about strategies or problem-solving paths that we have learned during our life. Strategies can differ from person to person, based on their experiences, and some strategies are common. We need the ability to have a user update strategies in our model based on new paths that the user has learned over time. In order to do this, strategic knowledge needs to be separated from domain knowledge so that we can use strategies to a greater advantage. Strategic knowledge can be represented in formal computational logic and, in this form, can provide many advantages. First, strategic knowledge can be added by a user without re-writing all the strategies already in the strategic knowledge base. Secondly, logic engines known as formal theorem provers can provide a consistency check for the knowledge base and can identify any conflicts within it. Thirdly, research in natural languages and translators between English and Computational Logic are being developed (Pease & Murray, 2003). This would ease in having the user update the system and thus keeping it current with new strategies that the biologist may discover work well for their problems.

### **3.3 Domain Knowledge**

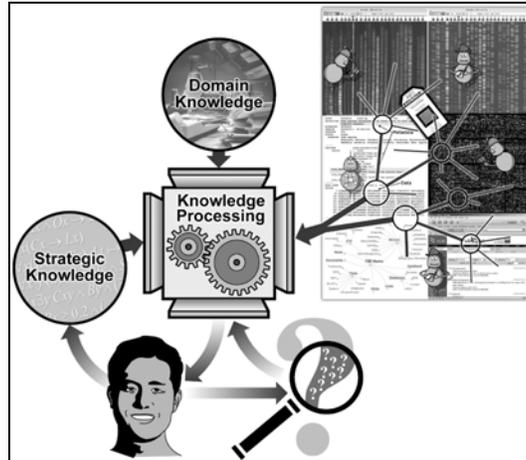
Of course, biologists have Domain Knowledge – knowledge about their field (this includes knowledge about tools, location of information, and knowledge deemed to be truthful). One of the key aspects dominant in this field is the understanding of missing knowledge from the Domain Knowledge. This is a key difference from other intelligent models – we assume that not all information is within the cognitive model during the time of processing a solution. As humans, when we do not have all the information in our brains, we know of ways to find the missing information and assimilate it into our working model. We call this New Knowledge – information we find and knowledge created while solving a problem.

Domain knowledge represents the knowledge about a particular problem area(s). Domain knowledge is contained in distributed knowledge bases. The usage of knowledge bases for problem solving has been growing over the past few years. In biology, it is being furthered not just by having vast amounts of data but by progress in ontologies for biology. This work will allow for a richer knowledge representation and manipulation. These techniques are

important for creating an infrastructure that is compatible with computational approaches and is also the key to adding new knowledge into the model.

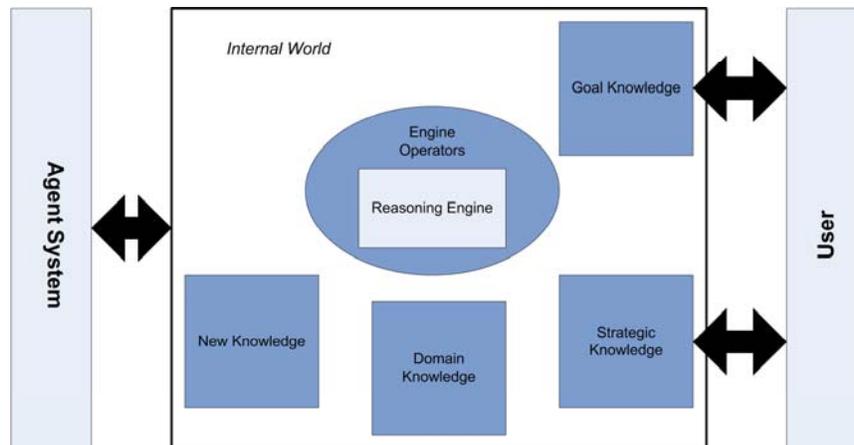
### 3.4 From Theory to Model

An overview of the interplay of these types of knowledge is depicted in Figure 1.



**Figure 1:** A conceptual overview of our cognitive system.

To create a computational cognitive model of Figure 1, we reverted to nature for guidance. A human brain contains billions of neurons, each possessing thousands of synapses connecting to other neurons. When a human is solving a problem, neurons and their pathways are triggered. If we could take a picture of the brain and highlight these neurons and paths that we “used” during problem solving, the result would be a connected graph. No matter what problem a human is solving, the final by-product is a mathematical graph (Weisstein 2002). Thus, the inner-workings of our cognitive model should function on connected graphs. This part of the model has one main requirement – it needs to be generic. Like a human brain, it needs to function no matter what the problem that it is trying to solve. This is represented as the “reasoning engine” denoted in Figure 2.



**Figure 2:** Computational Cognitive Model.

Now, as humans solve problems, there are triggers that fire certain neurons. If we could time-step and take pictures of the brain as it is solving a problem, the connected graph grows in size. In our model, we require “engine operators” to manipulate the graph. These operators are rules for building graphs, such as adding a node or a sub-graph to the current graph. This layer in our model could be viewed as generic to any problem – engine operators

function independently of the problem being solved. The knowledge layer of our cognitive model is domain specific. Again, different parts of the brain are triggered for different domain problems, for these areas “store” information about that particular domain or sub-problem. A block-level view of our computational cognitive model is depicted in Figure 2. The different types of knowledge are more computationally complex. Our view of the computations for each of the knowledge is described in the remainder of this section.

New knowledge is knowledge that has been currently “found” to exist; it is not part of our domain knowledge at the time of solving a problem. Humans obtain new knowledge from the senses; since computers do not have senses, a multi-agent system must mine and fuse data to create new knowledge (see left side of Figure 2). The agent tasks are similar to human tasks of finding information and using the correct tools to extract knowledge that is required while solving a complex problem. Ongoing research in problem-solving methods, data mining, and distributed agents will aid in this aspect of the model. One of the key questions is “when does new knowledge become domain knowledge?” Computationally, how does one merge new knowledge into the existing domain knowledge structure? This is similar to humans – you may read words in a book, but to truly understand it, you must fit that information into your current knowledge structure. We are currently developing this aspect of the model.

This section has provided an overview of our new cognitive model. There are some research challenges in developing such a model, and we touch upon some of these challenges in the next section.

## 4 Research Issues

Even though the cognitive model seeks to address bottlenecks and limitations in cognition, achieving such a model computationally still challenges many fields in computer science. One of the foreseeable hurdles in the generalization of the cognitive model’s applicability is the lack of ontologies in most areas of knowledge. Typically, the systematization of science vocabularies and their internal relationships has not been hitherto a widespread activity. As an example, the authors were requested to investigate the applications of the model to problems in atmospheric science. Atmospheric science is another discipline that is currently hampered by problems similar to problems in biology: enormous and constantly-increasing amounts of data supplied by a wide array of instruments. In addition, the constantly changing nature of that field of study makes it difficult to replicate experiments. After interviews with several field practitioners, the authors were faced with the fact that while the physics fundamentals are well-established, there is a lack of meta-knowledge of the sub-field (atmospheric science). The same can be said for non-scientific disciplines such as intelligence analysis. Fortunately, the current drive for the automation of knowledge acquisition and processing is fostering the creation of ontologies. Because an important part of the implementation of our cognitive model is the existence of an adequate ontology, the biological sciences community again provides a significant advance in that matter through the creation of the Gene Ontologies (Gene Ontology Consortium, 2005) which helps establish the knowledge relationships amongst the objects of study.

Another hurdle will be the creation of a flexible and user-programmable reasoning engine. Up to this point, some of the problems of expert systems have been their lack of flexibility and difficulty in reconfiguring the inference engine components. Our reasoning engine is currently programmed using conventional logic programming formalisms. The authors believe that separating strategies, goals and domain knowledge was a step in the right direction. As it was mentioned before, strategies can be expressed in formal computational logic which provides many advantages. However, the point at which the user can input new strategies and rules using natural language is still quite a distance away.

To provide a working proof of the assumptions presented in this paper, the authors have been developing a software prototype of the cognitive model. This software system will take a goal-based description of our problem and through input of user-defined strategies, will attempt to provide a path towards a possible solution. The system, as it stands now, is focused on the solution of the gene expression problem insofar as the domain, goal, and strategic knowledge bases are concerned. Yet the internal workings of the reasoning engine will be knowledge-independent since they are graph-based.

The system is being implemented using a combination of the Java and Prolog languages. Java was chosen for its availability and ease of use as well as the existence of several open-source mathematical graph libraries and graph visualization libraries. The SWI-Prolog implementation (“SWI-Prolog”, 2005) was chosen because of its

robustness, size, and its inclusion of a well-developed Java-Prolog API library. Calls can be made directly to Prolog queries and if necessary, Prolog can call Java routines as well.

We expect that the cognitive model will help address the problem of mapping and navigating massive information landscapes through the use of our novel reasoning process together with the use of the autonomous agent system. The ultimate purpose of the computational model will be to aid the researcher in solving a complicated question by taking over the onerous tasks of retrieving the appropriate information, processing it and fusing it, thereby freeing the scientist to engage in more analytical work. The question could be raised of whether the authors are building another expert system, however, we believe that the inclusion of the New Knowledge aspect of the model separates this approach from the rest of the pack. By recognizing the fact that all information is dynamic and it should be updated frequently the model takes one step further into injecting intelligence in a well-understood process. The question will linger, though, of at what point the retrieved information becomes part of the established domain knowledge. That is something that will have to be addressed by the appropriate knowledge curators.

This section has provided some insight into the research challenges for building such a cognitive model. Even with these challenges, this research is extending the computer science field. The next section provides a literature overview of how our research relates to the field.

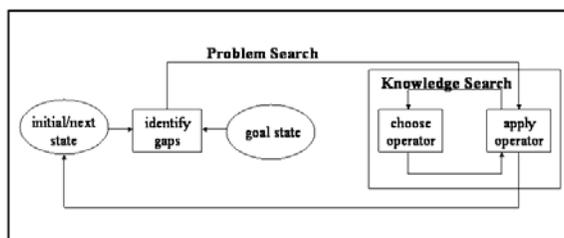
## 5 Literature Review

There is a great need for intelligent systems in biology. The recent data explosion in this field yielded massive and complex information that requires processing and analyzing it to determine new biological knowledge. As noted in (Altman 2001), certain key features of biological data make intelligent systems critical for their analysis:

- There is a need for robust analysis methods since biological data is normally collected with a relatively low signal-to-noise ratio.
- There is a need for statistical and probabilistic models since bioinformatics is still in its infancy.
- There is a need for complex knowledge representations since we know more about biology in a qualitative rather than in a quantitative sense.
- There is a need for cross-scale data integration methods since the data sources operate at multiple scales that are tightly linked.
- There is a need for data integration methods since biological data is distributed.

Reviewing the research work in biology and intelligent systems, we have seen how intelligence-driven applications have aided biologists, spanning the landscape from extracting weak trends in data to extracting high-quality information-level summary knowledge from scientific databases. The research solutions to some of the challenges facing biology have involved multiple disciplines including data mining, ontologies, knowledge management, mathematics, computer graphics, human-computer interaction, and artificial intelligence. What is missing from this landscape is a model that encapsulates how biologists think and work in their current environment to discover new knowledge. There is a need to support biologists from a cognitive perspective.

The reasoning process underlying our cognitive model relates to Newell's problem-space paradigm of intelligence (Newell 1990). As seen in Figure 3, this paradigm considers problem-solving to be a process that involves two searches – problem search and knowledge search.



**Figure 3:** Newell's problem-space paradigm of intelligence.

Problem search assumes the existence of a problem space for the task solved. This implies the availability of: a goal state  $g$  that describes what needs to be known at the end of problem solving; an initial state  $i$  that describes what is known at the start of problem solving; and a set of operators  $P$  that can be used to generate new states in the problem space. As the outer loop in Figure 3 indicates, starting from state  $i$ , problem search applies one operator in  $P$  at a time to generate a new state in the problem space, with the hope that this state will match (or at least be closer to) the goal state  $g$ . Thus, problem search uses operators in  $P$  to generate states in the problem space that form a path from state  $i$  to state  $g$ .

Knowledge search occurs in the inner loop of problem search, as seen in Figure 3. It guides problem search by using search control knowledge to select the operator in  $P$  that should be applied to generate the next state on the path from  $i$  to  $g$ . Knowledge search has usually associated with it a fixed space whose structure (connectivity and how paths through it are described) pre-exists. That is, while problem search occurs in a space that is constructed dynamically, knowledge search typically occurs in a fixed pre-existing structure.

We expand on Newell's paradigm to involve new information that can be gathered from the external world. Thus, we are proposing an extension or elaboration on the knowledge search to include information that can be gathered. Knowledge search still occurs in a pre-existing structure, but it is not fixed.

Research in cognitive models has also been growing. This model relates well to current thoughts on cognitive models of the brain. For example (Wang & Wang, 2002), the authors attempt to develop functional and cognitive models of the brain. The functional model relates to our model in two ways. First, the functional model contains NI-OS (the thinking engine) and NI-App (a set of acquired life applications). Secondly, the functional model has input sensors to gather new knowledge about the external world. As the reader can determine, these functional model items are related to our thinking engine, strategic knowledge, and multi-agent system aspects.

During the last three decades, research on cognitive architectures has been growing and has provided a variety of architectural classes that make different assumptions about the representation, organization, utilization, and acquisition of knowledge. We have surveyed this landscape to find a cognitive architecture that addresses several key features:

- A cognitive architecture that can be updated by a user (non-programmer) to keep the system current in both problem-solving techniques and tools.
- A cognitive architecture that can function in both "fuzzy" defined and dynamic environments.
- A cognitive architecture that is robust to solve many different problems without the need to reprogram the entire system.

Current architectures were not able to handle all of these requirements. This lead us to develop a different cognitive model to base our cognitive architecture upon. For example, Soar (Laird, Newell, & Rosenbloom, 1987) is a cognitive architecture in which all long-term knowledge takes the form of production rules, which are in turn organized in terms of operators associated with problem spaces. Our approach differs – we explicitly define our goal knowledge by describing the structure that our model seeks to produce as an answer. This allows the user to update the goal knowledge (or how should a solution look) and thus allows for expansion of the system as the user finds new ways of solving a problem. Another example is ICARUS (Shapiro & Langley, 1999). ICARUS focuses on reactive execution of existing skills rather than on problem-space search. We feel that the problem-space search is more robust in how humans solve their problems and will lead to a model (and thus architecture) that can be applied to many domains. Another example is ACT-R (Anderson & Lebiere, 1998). ACT-R has two distinct memories: a declarative memory that encodes knowledge about facts and events; and another memory that stores procedural knowledge in the form of production rules. ACT-R mixes strategic and domain knowledge together into its production rules. We feel that strategic knowledge needs to be independent of domain knowledge in order to allow the system to expand as the user discovers new paths for finding a solution to a problem. If domain knowledge is mixed with strategic knowledge, then updating the knowledge in a system is a tedious programming task that involves domain experts and programmers, with usually a massive revamping of code. Overall, intelligent systems have been considered as "black boxes" by users. Users feel that they do not understand how these intelligent systems find answers, and users feel powerless about what information (or intelligence) an intelligent

system is using. Biologists have a need for a system that they can manipulate and understand. We want to address this need through the robustness of our cognitive architecture.

## 6 Conclusions

In this paper, we provided an overview of our new cognitive model to aid biologists in knowledge discovery. Bioinformaticians are overwhelmed at the amount of fluctuating knowledge, data, and tools available in solving their problems. Problem-solving some tasks in biology can take anywhere between a few hours to several months; therefore, there is a need for a system to overcome several computational bottlenecks and thus, augment the biologists performance. We propose a new cognitive model to address this need by providing many key features: the model incorporates any domain since the knowledge bases are separated from the thinking engine; the thinking engine creates a directed graph that gradually grows links between nodes depicting domain objects and relations, but needs no understanding of the domain that it is acting upon; current knowledge is added to the model using a multi-agent system to populate the new knowledge base, thus leading the model to be more dynamic in information content; and new strategies on solving the higher-level problem can be incorporated easily, since domain knowledge and strategic knowledge are not incorporated as one knowledge base within the system.

There are many challenges to information and knowledge processing in biology. In our opinion, current computational cognitive architectures are not robust enough to handle such dynamic environments as biology. Many tools have been developed, but no architecture has been designed to integrate current technology, data, and knowledge into a complex problem-solving environment that can aid biologists in knowledge discovery.

## 7 Acknowledgements

The research described in this paper was conducted under the LDRD Program at the Pacific Northwest National Laboratory, a multi-program national laboratory operated by Battelle for the U.S. Department of Energy under Contract DE-AC06-76RL01830.

## 8 References

- Altman, R.B. (2001) Challenges for Intelligent Systems in Biology, *IEEE Intelligent Systems*, 16 (6), 14-18.
- Anderson, J.R., and Lebiere, C. (1998) *The Atomic Components of Thought*, Lawrence Erlbaum Associates, Mahwah, NJ.
- Eisen, M.B., Spellman, P.T., Brown, P.O., & Botstein, D. (1998) Cluster analysis and display of genome-wide expression patterns, *Proceedings of the National Academy of Sciences*, 95(25), 14863-14868.
- Gene Ontology Consortium (2005) Gene Ontology. Retrieved March 4, 2005 from <http://www.geneontology.org>
- Halkidi, M., Batistakis, Y. & Vazirgiannis, M. (2001) Clustering Algorithms and Validity Measures, *Thirteenth International Conference on Scientific and Statistical Database Management*, 3-22.
- Laird, J.E., Newell, A., & Rosenbloom, P.S. (1987) Soar: An Architecture for General Intelligence, *Artificial Intelligence*, 33, 1-64.
- Langley, P., & Laird, J.E. (2002) *Cognitive Architectures: Research Issues and Challenges*, (Technical Report) Institute for the Study of Learning and Expertise, Palo Alto, CA.
- Luo, F., Tang, K., & Khan, L. (2003) Hierarchical Clustering of Gene Expression Data, *Third IEEE Symposium on Bioinformatics and BioEngineering*, 328-335.
- Luscombe, N.M., Greenbaum, D., & Gerstein, M. (2001) "What is bioinformatics? An Introduction and Overview", <http://bioinfo.mbb.yale.edu/~nick/bioinformatics>
- Newell, A. (1990) *Unified Theories of Cognition*, Harvard Press, Boston, MA.
- Park, J.H. (2002) Network Biology: Data Mining Biological Networks, *IEEE Intelligent Systems*, 17 (3), 68-70.

- Pease, A., & Murray, W. (2003) An English to Logic Translator for Ontology-based Knowledge Representation Languages, *IEEE International Conference on Natural Language Processing and Knowledge Engineering*, 777-783.
- Reichhardt, T. (1999) "It's sink or swim as a tidal wave of data approaches." *Nature*, 399 (6736), pp. 517-520.
- Ross, D.T., Scherf, U., Eisen, M.B., Perou, C.M., Rees, C., Spellman, P., Iyer, V., Jeffrey, S.S., Van de Rijn, M., Waltham, M., Pergamenschikov, A., Lee, J.C., Lashkari, D., Shalon, D., Myers, T.G., Weinstein, J.N., Botstein, D., & Brown, P.O. (2000) Systematic variation in gene expression patterns in human cancer cell lines, *Nature Genetics*, 24(3), 227-235.
- Scherf, U., Ross, D.T., Waltham, M., Smith, L.H., Lee, J.K., Kohn, K.W., Reinhold, W.C., Myers, T.G., Andrews, D.T., Scudiero, D.A., Eisen, M.B., Sausville, E.A., Pommier, Y., Botstein, D., Brown, P.O., & Weinstein, J.N. (2000) A cDNA microarray gene expression database for the molecular pharmacology of cancer, *Nature Genetics*, 24 (3), 236-244.
- Shapiro, D., & Langley, P. (1999) Controlling Physical Agents Through Reactive Logic Programming, *Proceedings of the Third international Conference on Autonomous Agents*, 386-387.
- Sofia, H., Chen, G., Hetzler, B.G., Reyes-Spindola, J.F., & Miller, N.E. (2001) Radical SAM, a novel protein superfamily linking unresolved steps in familiar biosynthetic pathways with radical mechanisms: functional characterization using new analysis and information visualization methods, *Nucleic Acids Research*, 29(5), 1097-1106.
- unknown, (1999) Dipping into DNA Chips, NetWatch, *Science*, August 6, 285 (5429), 799d.
- unknown, (2005) Growth of GenBank. Retrieved February 28, 2005 from <http://www.ncbi.nih.gov/Genbank/genbankstats.html>
- unknown, (2005) SWI-Prolog. Retrieved March 1, 2005 from <http://www.swi-prolog.org>
- unknown, (2005) SWISS-PROT Protein Knowledgebase Release 46.1 Statistics. Retrieved February 28, 2005 from <http://www.expasy.org/sprot/relnotes/relstat.html>
- Wang, Y., & Wang, Y. (2002) Cognitive Models of the Brain, *IEEE International Conference on Cognitive Informatics*, 259-269.
- Weisstein. E.W. (2002) Connected Graph, From MathWorld--A Wolfram Web Resource. Retrieved March 2, 2005 from <http://mathworld.wolfram.com/ConnectedGraph.html>