2009

CHAPTER 11

SUPPORTING INNOVATIVE CONSTRUCTION OF EXPLANATORY SCIENTIFIC MODELS

WILL BRIDEWELL

STUART R. BORRETT

PAT LANGLEY

CONSIDER the following scenario with characteristics common to science. An ecologist is studying an aquatic ecosystem to learn how it functions. Data gathering has yielded weekly measurements for several variables, such as the concentrations of nitrogen, phosphorus, and phytoplankton. Daily measurements exist for water temperature, solar irradiance, wind speed, and wind direction. Finally, weekly reports of zooplankton abundance exist for the summer months. Hopefully, this information will lead to a mathematical model that accurately predicts the ecosystem's response to environmental management.

Few would deny that such model-construction tasks involve creativity. The scientist must assemble a new artifact that explains the observations in a consistent and coherent way. The space of possible models is quite large, making it impractical to simply consider each candidate in turn. The problem may not be as challenging as discovering the general theory of relativity, since it does not involve paradigm shifts, but even model creation within an established theoretical framework can stretch the cognitive abilities of experienced scientists.

Luckily, the situation is not hopeless. In addition to the observed data, the ecologist also has knowledge regarding mechanisms that might plausibly operate within an aquatic ecosystem. For example, the zooplankton likely eats the phytoplankton, but the rate of consumption, the regulating factors, and the overall effects of this grazing process are undetermined. The scientist can also use deeper theoretical knowledge to guide the construction of the final model. This knowledge can consist of reasonable bounds on rates, plausible causal links, and possible formulations of grazing, amongst other things. In many cases, the ecologist will even have an existing mathematical model (e.g., Moore et al., 2002; Benz et al., 2001) that is adaptable to the current ecosystem.

Nevertheless, this remains a challenging task that could benefit from computational assistance. Current approaches to ecosystem modeling range in scope from writing custom FORTRAN programs (Arrigo et al., 2003) to using graphical model-building tools such as STELLA (Richmond et al., 1987; Sage et al., 2003). These solutions vary in difficulty of use, but the end product for each is a simulation model that one can represent as a system of differential equations. There are two primary disadvantages to using such software. The first is that one must make simultaneous decisions about which biological processes to model and how to represent them. This aspect mixes theoretical knowledge about how ecosystems operate with problem-specific assumptions relevant only in a working context. As a result, the models' complexity increases while their comprehensibility decreases. The second disadvantage is that one must build each model by hand. This requirement creates undue conservatism by contributing to a general reluctance to explore and evaluate alternative models, which in turn decreases the chances of finding innovative solutions.

We believe that concepts and methods from artificial intelligence and cognitive science suggest a better approach to designing computational aids for scientific model creation. In the pages that follow, we describe PROMETHEUS, an interactive environment for constructing and revising process accounts of dynamic systems (Bridewell et al., 2006). To clarify the rationale behind the program's design, we must recount the challenges that the problem presents to intelligent assistants.

First, we should note that despite recent rhetoric in the data-mining literature scientific data relevant to discovery are often rare and difficult to obtain. The costs of collecting and preparing the data are non-trivial, and high rates or long periods of sampling may be impossible. As a result, the number of samples probably ranges in the low hundreds. Given the number of variables, parameters, and relationships in the target models, common methods for data mining are inappropriate, and we require new techniques.

Another challenge requires us to support model-revision in terms of both causal structure and parameters. Systems scientists like our ecologist come to a modeling task with prior knowledge of various sorts. At one level, this knowledge consists of the possible interactions among entities in a system and ways to formulate those relationships. For example, the ecologist knows that a process of phytoplankton growth exists and that it must be included in the final model. However, whether this growth can best be modeled as exponential, logistic, or something more complex may be unknown. At a different level, the ecologist may seed the discovery process with a prior model and search for revisions that explain the current data.

The third challenge is the need for *communicable* models. As we mentioned, ecologists often express their models in terms of differential and algebraic equations, but machine learning traditionally uses its own notations (e.g., decision trees, logical rules, Bayesian networks), which result in models that are not easily communicated to domain scientists. We need techniques for knowledge discovery that produce output that closely approximates the scientists' own modeling language.

In addition, scientists want models that move beyond description to provide *explanations* of their data. Regression-style techniques generate pithy summaries of the observations, but they fail to make contact with the underlying generating mechanisms. This desire poses the challenge of developing methods that construct explanatory models rather than purely descriptive ones.

These issues raise algorithmic challenges, but introspection suggests another problem. Many computational discovery systems strive to automate the activity of model construction, but few scientists want to be replaced. However, they may well accept computational tools that carry out tedious aspects of searching through the model space, provided we find ways they can participate in the model-building endeavor. Ideally the software would perform lower-level tasks and free the scientist to concentrate on higher-level goals. Thus, it behooves us to design interactive systems that support a creative partnership between software and scientific domain experts.

This chapter describes the application of ideas from artificial intelligence and cognitive science approaches to stimulate discovery in the systems sciences like ecology. As such, it introduces the above challenges and our response as embodied in PROMETHEUS, an environment that supports the creation of quantitative models of dynamic systems. The next section describes challenges in user interaction and our responses. We then discuss the challenges in developing a model discovery system, highlighting the integration of various threads of research to compose an intelligent assistant for scientific modeling. After this, we briefly discuss previous results from the use of PROMETHEUS and identify new challenges that have arisen during experimentation. Finally, we summarize our work and highlight unmet challenges that seem ripe for further research.

Addressing Challenges in Communication

One should address challenges of user interaction from the foundation upwards when building an intelligent system. *To meet the challenges of model comprehensibility and explanation,* PROMETHEUS *represents its knowledge in a language that builds on systems of equations.* Models expressed as differential and algebraic equations commonly appear in the ecosystems literature and pervade systems science as a whole. However, even in this familiar form, the explanatory content of the models is not easily accessible. Fortunately, we can turn to ecology for a solution. The models in this domain often portray mechanisms (e.g., Gaff et al., 2004; Sarmiento et al., 1998), which suggests that the language of entities and the processes in which they participate (Machamer et al., 2000) is appropriate. Forbus (1984) previously developed a formalism for qualitative process models, which takes this basic perspective, but our purposes, which include close contact with numeric data, suggest a need for *quantitative* process models.

Representing the models as mechanisms also addresses the challenge of a participatory system. Although systems of equations are the output of this task, scientists initially work at a conceptual level. For instance, Jørgensen and Bendoricchio (2001) recommend developing a conceptual structure of the studied system as the first step in ecological modeling. They suggest building this structure by listing the state-variables and then identifying the physical, chemical, and biological processes that link the variables to each other and to the environment. Afterwards, one uses mathematical formulations of the processes to produce an equivalent system of equations. We want to support this modeling style that gives scientists the creative freedom to design the larger-scale features of the modeled system before making low-level decisions about the nature of the processes.

Finally, the quantitative process representation also addresses a technical challenge. Unlike previous modeling environments, PROMETHEUS supports automated search through the space of models. The space of differential equations is far too large for unguided search, and it most certainly contains models that fit the observed data but lack plausibility. The processes used by PROMETHEUS contain meaningfully grouped chunks of equations that one can combine with others to form the model. For instance, a process describing predation between species would have one equation element that decreases the prey population and another that increases the predator population. Therefore, removing such a process would completely excise predation from the model and update the system of equations appropriately. By defining these processes, one can use knowledge from systems science to restrict PROMETHEUS's search to a space of plausible models.

Both the entities and the processes in quantitative process models have two forms: generic and instantiated. A generic entity, as shown in Table 11–1, declares the variables and parameters that store relevant properties. Parameters at both the process and entity levels are immutable, model-specific values that fall within a specified range. In contrast, the variable values can change over time. Variables themselves fall into one of three classes. An *exogenous* variable can only influence processes in the model, and its values must be read from a data source. An *observed* variable must be

TABLE 11–1 The generic entity for a primary producer contains a measure of its species' concentration, growth rate, and loss rate. Processes affecting the concentration will have additive influence, whereas the current growth rate will be the minimum of values produced by multiple processes. The loss rate must fall between zero and ten.

```
generic entity primary_producer:
  variables:
    conc {sum}
    growth_rate {min}
  parameters:
    loss_rate [0, 10]
```


explained by the model and must have associated data for purposes of comparison, and an *unobserved* variable needs only an initial value and a range in which this value should fall. All variables and parameters associated with an entity are passed along with that entity to any process in which it participates. One can instantiate a generic entity by specifying whether each variable is observed, unobserved, or exogenous; identifying necessary data sources; and assigning a numerical value to each parameter.

Generic processes contain entity and process roles, parameters, conditions, and equations. Entity roles consist of a local name for an entity along with the number and types of entities that can fill that role. For instance, the exponential loss process in Table 11–2 requires a single generic entity that has type "primary producer" or "grazer." A process role gives a process type and the list of entities to pass along to the selected subprocess. In addition, Boolean conditions control whether a process is active based on the current value of variables in the model, and equation elements define the quantitative behavior of the process. As a final feature, each generic process has a type that helps guide the search for plausible subprocesses. The instantiated form of a process requires one to specify the participating entities, any subprocesses, and local parameter values.

Generic processes and entities address the challenges of incorporating prior knowledge and model discovery with few data. The generic components along with the constraints among them limit the model space to a subset of plausible structures, and this tight restriction helps offset the difficulties of knowledge discovery from small data sets. The structural constraints manifest in three ways. First, the use of generic entities along with entity roles constrains the viable participants in a

TABLE 11–2 The generic process for exponential loss has type "loss" and takes exactly one entity with type primary producer or grazer. The single equation in this process states that the first derivative of the concentration with respect to time is equal to a loss influenced by the species' loss rate.

```
generic process exponential_loss {loss}:
  entity_roles:
    S {primary_producer, grazer} <1 to 1>
  equations:
    d[S.conc, t, 1] = -1 * S.loss_rate * S.conc
```

process. Second, the bounds on parameter values help guide estimation tools, which we will discuss in the next section. Third, the hierarchy imposed by process types and subprocesses defines a modified AND/OR tree of possible structures. The subprocesses, which may be optional, specify the AND branching and specify which process types must occur along with the current generic process. These process types establish exclusive OR branches, specifying a set of generic processes that may satisfy a particular process role. To illustrate, the process type "growth" may have several forms (e.g., exponential, logistic, limited). In this case, suppose that a top-level process called "ecosystem" requires a growth process. This need constitutes an AND branch of the tree, whereas the multiple processes of the correct type compose the OR branch.

The creation of quantitative process models requires multiple steps. Initially, a scientist must develop a library of generic processes and entities. In our experience, one begins this task at an abstract level by identifying the entities and processes relevant to a chosen context (e.g., aquatic ecosystems). Next, one specifies the mathematical forms of the processes, selects the important properties of the entities, and determines the structural constraints for the hierarchy. Much of this work is straightforward. For instance, the process forms appear in the literature, and the generic entities relate directly to theoretical terms and the measurements one would typically make in the domain. However, the constraints encoded in the process hierarchy reflect implicit knowledge and are more difficult to elicit. In addition, the syntax of the constraint-specification language can influence the organization of equations into processes and properties into entities. As a result, assembling a library involves an iterative refinement of one's knowledge and increases in difficulty with the complexity of the process hierarchy. Fortunately, once completed, a single library describes the theoretical knowledge for a sizable range of problems. Therefore, one can build multiple models from a single library, make minor adaptations to fit similar domains, and borrow components for use in other problems.

To create a model from a domain-specific library, one selects the relevant entities to instantiate, the processes that link these entities, and the particular process alternatives that drive the observed dynamics. This step may constitute a stopping point, but it is more likely that the scientist will compare the model to some observations and adjust the model as necessary. We are developing PROMETHEUS to support as much of this procedure as possible.

ADDRESSING CHALLENGES IN LEARNING

PROMETHEUS consists of two major components—the user interface and the model-induction engine—each with its own set of challenges. Here we mainly discuss our approach to model construction and revision, but we first describe PROMETHEUS's wide range of interaction. At a basic level, one can create a model, view its causal flow as shown in Figure 11–1, and see the current system of equations. Additionally, the program supports model evaluation through the inclusion of a simulation engine and a means to compare the resulting trajectories with observed data. Moreover, one can manually revise models by altering parameters and adding or deleting both processes and entities. Thus, at its core, PROMETHEUS supports creativity through the ability to freely design and test quantitative process models.

At this level, PROMETHEUS operates much like other modeling packages (apart from its emphasis on mechanisms), but the integration of system identification and artificial intelligence components set it apart. These elements provide support for automated parameter estimation, model construction, and model revision. Todorovski et al. (2005) describe the underlying algorithm for these features.[1] This approach operates in two separable stages, the first of which defines the symbolic space of model structures. Beginning with the root process, PROMETHEUS satisfies the minimal set of constraints imposed by the hierarchy by including all required processes and no optional ones. This step produces a set of model structures that relate entities and processes but lack values for the parameters. At this level of the search, we predominantly draw on traditional, symbolic techniques from artificial intelligence. Specifically, the program performs a beam search through the AND/OR space defined by the background knowledge and guided by a quantitative measure of fit (i.e., sum of squared error or variance normalized mean-squared error).

For each structure, PROMETHEUS searches a second space defined by the numeric parameters. We use techniques from system identification to perform a gradient-descent search based on the quantitative measure of fit. The core algorithm, which was designed by Bunch et al. (1993), fits

[1] PROMETHEUS's current interface uses an earlier induction algorithm that lacks support for entities and process hierarchies. We are adapting the environment to use these structures.

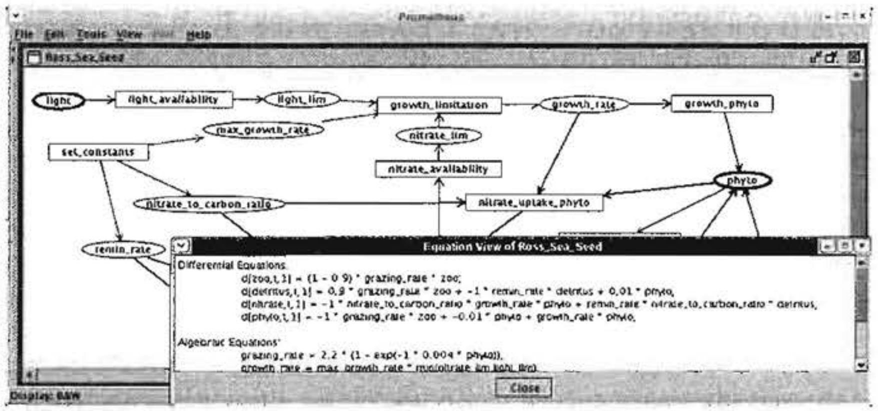

Figure 11–1 PROMETHEUS can display both a causal diagram of a model and the underlying equations. In the diagram, the ovals are variables and the rectangles are processes.

the parameters of dynamic, nonlinear systems of equations while ensuring that the resulting values fall within specified bounds. This algorithm performs a local search, so the system lets one specify a number of restarts that each explore the parameter space from a randomly selected point. In practice, we have found this approach to run slowly and to have high variance, which influences the selection of model structures. The FUSE algorithm (Bridewell et al., 2005) integrates research on ensemble methods to reduce overall variance, but we have yet to incorporate this solution into PROMETHEUS.

PROMETHEUS *meets the challenge of model revision by providing the scientist with several controls to influence semi-automated revision.* As input, the scientist provides an initial model along with three lists: (1) processes that may be removed, (2) generic processes that may be instantiated, and (3) processes and entities whose parameters may be changed. The structural search uses the iuitial model with all deletable processes removed to seed the search. From that point on, the algorithm tries both to add deleted processes back to the model and to add instantiations of the specified generic processes when possible. For the most part, revision operates just like induction from scratch, but the scientist's guidance further limits the possible moves in the search space. Upon completion, the program returns a list of the best models ranked by the chosen measure of quantitative fit. Each of these models can serve as a foundation for future revisions.

We can best describe the use of PROMETHEUS by example. Consider the ecologist described at the beginning of this chapter. This modeler begins by identifying a set of generic entities and processes expected to operate within the observed ecosystem. One could draw this knowledge from an earlier developed library, extract it from textbooks or articles, or create it a new. After developing this library, the ecologist can build an initial model in PROMETHEUS. The model may contain nothing more than a list of the entities, or it could be fully detailed, with all suspected relationships indicated with instantiated processes. For this example, we will assume the second case.

With a model structure in place, the ecologist can then fit the parameters using all available data and simulate the resulting model to compare the output with observations. Now, suppose that the scientist notices that the simulated phytoplankton population fails to decrease as expected. Examination of the model shows that nothing grazes on the phytoplankton, even though zooplankton exist in the region under study. The ecologist can either manually select and add the grazing process or have PROMETHEUS search the reduced space of models consisting of the initial structure plus all possible options for the inclusion of grazing. If the user opts for automated revision, the program will yield a ranked list of plausible models. The scientist may select, simulate, and evaluate each of the results, and if necessary, the revision process can continue.

Importantly, PROMETHEUS transforms the modeling task by automating lower-level tasks such as assembling equations, fitting parameters, and generating alternatives. Instead, the ecologist can concentrate on the types of processes likely to appear in an ecosystem, their alternative functional forms, and the constraints among the processes. More directly, the automated search tools in PROMETHEUS let one work closer to the theoretical structures and modeling assumptions that characterize plausible explanations. Given this information, the software explores the space of candidates, highlighting those few that both fit the background knowledge of the domain and match available observations.

INITIAL EXPERIENCES WITH PROMETHEUS

Researchers have evaluated PROMETHEUS's behavior in a variety of scientific domains. In this section, we summarize the nature of the tasks, the results obtained with the system, and some lessons suggested by those experiences. We focus on model induction in our description of two scientific tasks, and discuss an application of model revision to the Ross Sea domain. Detailed results appear in earlier papers, so here we present only the highlights.

Predator–Prey Interactions in Protists

Predator–Prey systems are among the simplest in ecology, which makes them a good starting point for evaluating PROMETHEUS. In earlier work (Asgharbeygi et al., 2006; Todorovski et al., 2005), we explored the protist system composed of the predator *Didinium nasutum* and the prey *Paramecium aurelia* using data from experiments originally reported by Veilleux (1979). Jost and Ellner (2000) report the observed values, which consist of population concentrations recorded in 12-hour intervals for three experimental conditions. The data, some of which appear in Figure 11–2, are fairly smooth and exhibit oscillatory behavior.

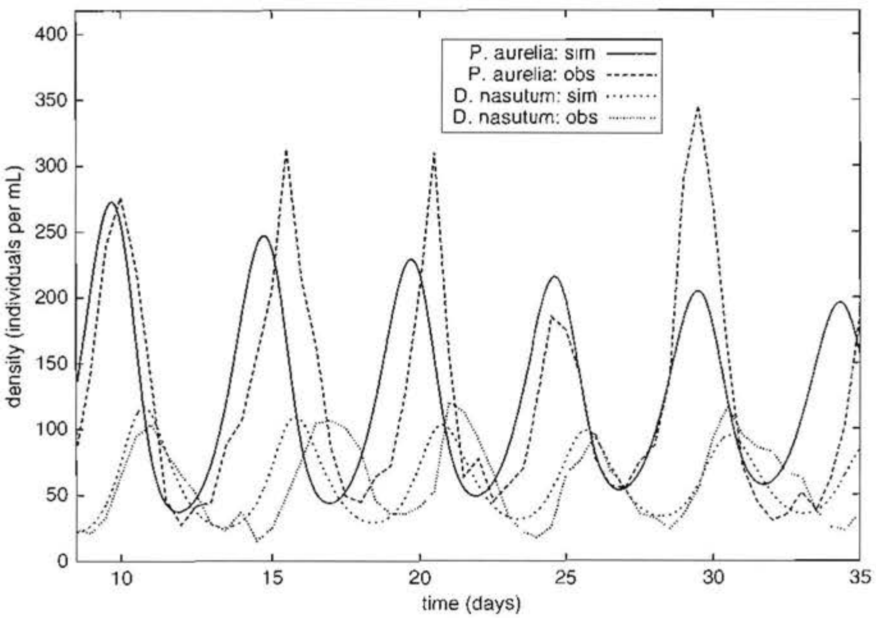

Figure 11–2 Population dynamics in a simple predator–prey ecosystem.

For this domain, we provided PROMETHEUS with generic processes for prey growth, predator decay, and predation, including alternative functional forms. When constrained by the process hierarchy, these defined a space of 24 distinct model structures that, with parameters specified, predict trajectories for the two species' concentrations from their initial values. The system's search of this space produced a plausible model that included processes for growth, predation, and decay. As shown in Figure 11–2, the simulated curves track the heights and timing of the observed trajectories reasonably well.

Notably, we encountered problems when we presented the system with the entire Jost and Ellner data set, and obtained these results only when we provided it with a selected subset. Measurements early in the time series had considerably lower peaks, which suggested a different regime was operating for unknown reasons. This result reveals an important ability that PROMETHEUS currently lacks: *When a scientific modeling system cannot explain an entire set of observations, it should consider ignoring some of the data.* This capability could help the system both identify separate regimes and minimize the effects of outliers during the early stages of modeling. Clearly, human scientists have this capacity, and future versions of PROMETHEUS would benefit from a solution that meets this challenge.

Population Dynamics in the Ross Sea

The Ross Sea in the Southern Ocean involves a somewhat more complex ecosystem. Here the phytoplankton, which may play an important role in the global carbon cycle (DiTullio et al., 2000), undergo repeated cycles of population increase and decrease. In this case (Asgharbeygi et al., 2006), we had access to two sets of 188 daily measurements for phytoplankton that spanned two successive years. Concurrent data were also available for nitrate concentrations and ice coverage; we used an algebraic equation to simulate the light dynamics.

Based on discussions with the team's biological oceanographer (Kevin Arrigo), we identified entities of interest and developed 25 generic processes that encoded how they might interact. In addition to phytoplankton and nitrate, the entities included detritus, which results from phytoplankton decay, and zooplankton, which feeds on phytoplankton. Because neither were measured, the researchers treated attributes of both as unobserved theoretical variables. In addition, they seeded PROMETHEUS with an initial model that substantially reduced the size of the structural search space.

PROMETHEUS produced a number of models that made sense ecologically and that fit the first year's data closely, but they generalized poorly to the second year's observations.

Inspection of the model suggested that ice differences across the years had little effect on phytoplankton growth, although this had originally seemed a likely explanation of differences between the two years. Discussion with the oceanographer led the group to include another generic process, which states that phytoplankton's absorption of nitrate depends on available light. Based on this information, PROMETHEUS found another model that fit the first year's data nearly as well as the earlier candidate but that, as Figure 11–3 shows, generalized much better to the second year. The implication is that the nitrogen-to-carbon ratio for phytoplankton varies as a function of light availability, which the oceanographer believes is an important ecological claim.[2] The original vision for PROMETHEUS was that it

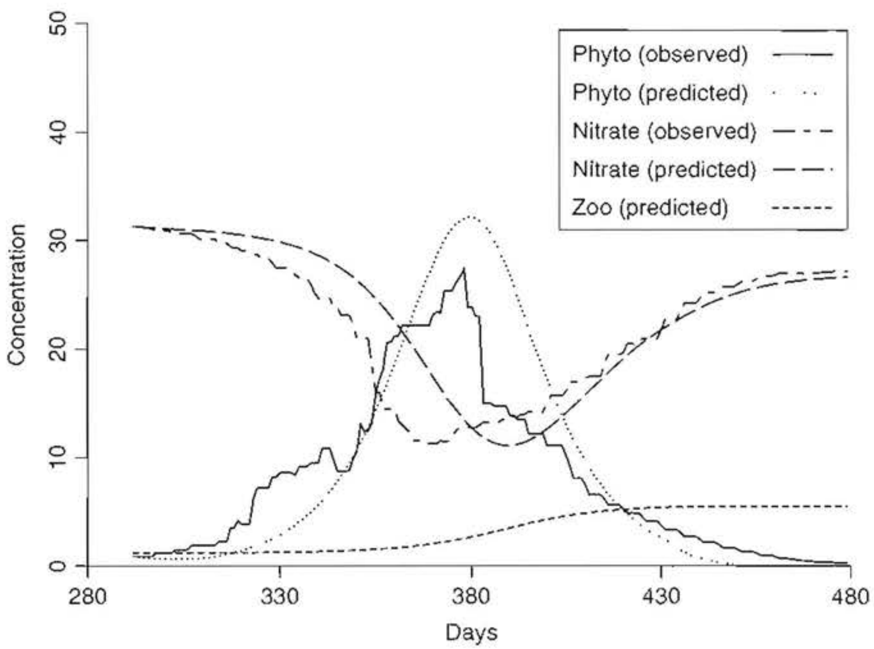

Figure 11–3 Performance on test data from the Ross Sea.

[2] This finding was made before support for entities and process hierarchies was complete.

should support the scientist's search for models in a well-defined space. However, our experience with the Ross Sea revealed another key ability that the system lacks: *When a scientific modeling system cannot account for observed differences, it should consider new mechanisms that expand its space of plausible models.* Human scientists prefer to explain phenomena in terms of familiar mechanisms, but they can consider new processes when necessary, presumably by falling back on more general knowledge. Adding such a capability to PROMETHEUS is another important direction for future work.

Biochemical Kinetics

We also applied PROMETHEUS to a problem from biochemical kinetics (Langley et al., 2006), which studies physiological changes in metabolites over time. Here we drew upon time-series data collected by Torralba et al. (2003) about the glycolysis pathway, which converts glucose into pyruvate and which plays an essential role in most life forms. Torralba's group used an impulse-response method that, given a biochemical system in steady state, briefly increases the inflow of one substance and measures its effects on others over time. We used 14 data points for six distinct glycolitic metabolites.

For this domain, we provided the system with five generic processes that encoded four types of metabolic reactions appearing in pathway models. These differ in how they affect positive and negative fluxes (i.e., flow into and out of a reaction pathway) of the substances involved. The researchers crafted four generic processes—*irreversible, reversible, inhibition,* and *activation* reactions—along with a fifth that stated a metabolite's concentration changes as a weighted sum of its positive and negative fluxes, with each flux term being multiplied by its respective rate.

When provided with the data and these generic processes, PROMETHEUS searched a space of 172 distinct models and estimated parameters for each candidate. Figure 11–4 shows both the observed trajectories and those predicted by the best-scoring model, which produces good fits in both qualitative and quantitative terms. However, the model structure differs from the generally accepted glycolysis pathway in that it lacks inhibition and activation processes. Presumably, this occurred because the system could not introduce unobserved entities to serve as inhibitors and activators, which suggests another limitation: *A scientific modeling system should consider introducing theoretical entities that augment those provided by the user.* PROMETHEUS can already generate

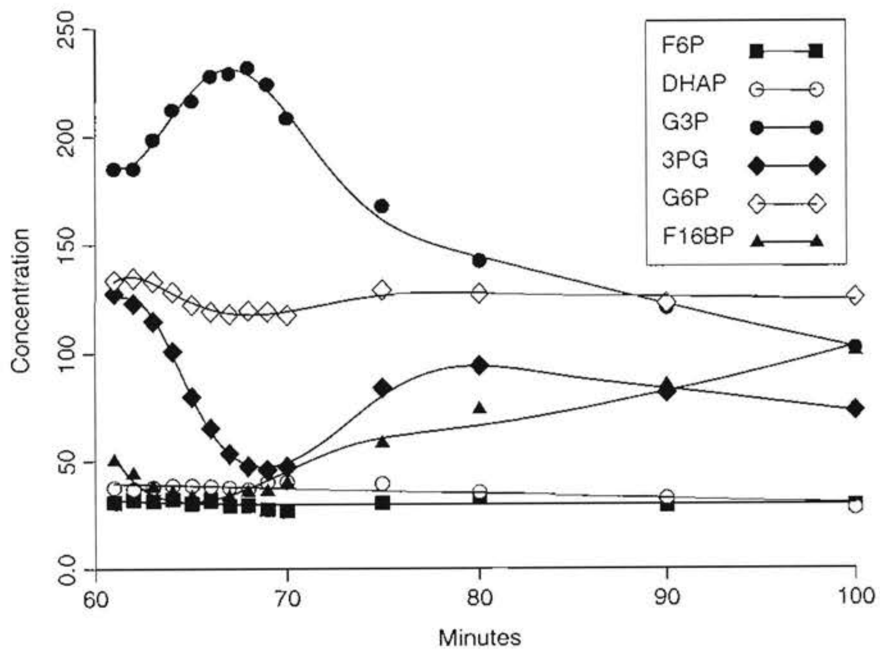

Figure 11-4 Observed (points) and predicted (lines) trajectories of chemical concentrations in the biochemical kinetics domain.

models with unobserved terms, but only when they are given as input. Introducing the ability to postulate new entities, as constrained by background knowledge, would extend the system's ability to generate plausible explanatory models.

Discussion

At the outset, we described five challenges that arise when building a tool to support the construction of scientific models. These included sparsity of relevant data, the presence of prior models and knowledge, a match between system output and the primary domain language, the production of explanatory models, and an emphasis on interactivity. We designed the formalism for quantitative process models and generic processes with these challenges in mind, and we integrated techniques from artificial intelligence and system identification in response.

The formalism for quantitative process models has some clear advantages. First, one can directly translate the models into a more familiar representation for scientists, thereby addressing the challenge of communication. Second, casting the domain knowledge as processes leads to mechanisms that explain the studied system's behavior. Finally, the processes mesh well with the conceptual stage of model-building, which eases the input of domain knowledge and prior models to the program.

To meet the challenges involved in model construction and revision, we borrowed from several research traditions. Heuristic search of AND/OR trees provides a means for navigating the space of model structures, while tools from system identification (e.g., Åström and Eykhoff, 1971) direct search through the parameter space. The use of prior knowledge helps constrain search to produce plausible models even without large data sets. Finally, theory revision techniques (e.g., Ourston and Mooney, 1990) support interactive search, letting the user gauge the scope and nature of revisions at each step in the modeling process.

Experiments with Prometheus identified several open challenges for the artificial intelligence community. First, we need a way to ignore connected sets of data, not just isolated outliers, that may stem from a different regime and keep a program from producing good models. In dynamic systems, assigning observations to different operating regimes will allow easier identification of the active mechanisms. Second, a program should be able to introduce new processes to its library. Third, model construction methods should introduce theoretical entities that are not specified explicitly by the user. These last two additions can increase the search space substantially, so we need more intelligent mechanisms to guide the structural search.

Perhaps the biggest surprise we encountered involved current software capabilities. In the early stages of our work, we believed that techniques for parameter estimation were ready for application. However, we found the tools available for nonlinear dynamical systems to be both unreliable and slow. Generally, parameter estimation techniques use very little knowledge, and we believe that ideas from artificial intelligence and knowledge-based reasoning could improve these systems on both fronts. One possibility is to incorporate scientists' knowledge of both the general shape that trajectories should take and the relationships among trajectories and parameters. Bradley et al. (2001) explored another possibility that used heuristics to avoid unnecessary parameter estimation. Capitalizing on this type of knowledge is the strength of artificial intelligence, and innovations in this area will have broad applicability.

In summary, we have seen that PROMETHEUS introduces a number of innovations that respond directly to the outlined challenges and support creative acts in science. These include a representation for models and background knowledge that supports communication with scientists, integration of domain knowledge to guide symbolic and numerical search, and incorporation of initial models and user input to guide revision. However, we have also seen that this combination of ideas does not exhaust the ways that we can support the creative activities of scientists as they develop models of dynamic systems. We need additional research that extends the power and flexibility of the modeling methods to better serve the needs of scientists.

ACKNOWLEDGMENTS

This research was supported by Grant Number IIS-0326059 from the National Science Foundation. We thank Kevin Arrigo for his expertise on the Ross Sea, and Nima Asgharbeygi, Oren Shiran, and Ljupčo Todorovski for contributions to the PROMETHEUS system.

REFERENCES

ARRIGO, K. R., WORTHEN, D. L., and ROBINSON, D. H. (2003). A coupled ocean-ecosystem model of the Ross Sea: 2. Iron regulation of phytoplankton taxonomic variability and primary production. *Journal of Geophysical Research-Oceans*, 108(C7), 3231, doi: 10.1029/2001JC000856.

ASGHARBEYGI, N., BAY, S., LANGLEY, P., and ARRIGO, K. (2006). Inductive revision of quantitative process models. *Ecological Modelling, 194*, 70–79.

ÅSTRÖM, K. J., and EYKHOFF, P. (1971). System identification—a survey. *Automatica, 7*, 123–167.

BENZ, J., HOCH, R., and LEGOVIC, T. (2001). ECOBAS: Modelling and documentation. *Ecological Modelling, 138*, 3–15.

BRADLEY, E., EASLEY, M., and STOLLE, R. (2001). Reasoning about nonlinear system identification. *Artificial Intelligence, 133*, 139–188.

BRIDEWELL, W., BANI ASADI, N., LANGLEY, P., and TODOROVSKI, L. (2005). Reducing overfitting in process model induction. *Proceedings of the Twenty-Second International Conference on Machine Learning* (pp. 81–88). Bonn, Germany.

BRIDEWELL, W., SÁNCHEZ, J. N., LANGLEY, P., and BILLMAN, D. (2006). An interactive environment for the modeling and discovery of scientific knowledge. *International Journal of Human–Computer Studies, 64*, 1099–1114.

BUNCH, D. S., GAY, D. M., and WELSCH, R. E. (1993). Algorithm 717: Subroutines for maximum likelihood and quasi-likelihood estimation of parameters in nonlinear regression models. *ACM Transactions on Mathematical Software, 19*, 109–130.

DITULLIO, G. R., GREBMEIER, J. M., ARRIGO, K. R., LIZOTTE, M. P., ROBINSON, D. H., LEVENTER, A., BARRY, J. B., VAN WOERT, M. L., and DUNBAR, R. B. (2000). Rapid and early export of *Phaeocystis antarctica* blooms in the Ross Sea, Antarctica. *Nature, 404*, 595–598.

FORBUS, K. (1984). Qualitative process theory. *Artificial Intelligence, 24*, 85–168.

GAFF, H., CHICK, J., TREXLER, J., DEANGELIS, D., GROSS, L., and SALINAS, R. (2004). Evaluation of and insights from ALFISH: A spatially explicit, landscape-level simulation of fish populations in the Everglades. *Hydrobiologia, 520*, 73–87.

JØRGENSEN, S. E., and BENDORICCHIO, G. (2001). *Fundamentals of ecological modelling*. New York: Elsevier.

JOST, C., and ELLNER, S. (2000). Testing for predator dependence in predator–prey dynamics: A non-parametric approach. *Proceedings of the Royal Society of London B, 267*, 1611–1620.

LANGLEY, P., SHIRAN, O., SHRAGER, J., TODOROVSKI, L., and POHORILLE, A. (2006). Constructing explanatory process models from biological data and knowledge. *Artificial Intelligence in Medicine, 37*, 191–201.

MACHAMER, P. K., DARDEN, L., and CRAVER, C. F. (2000). Thinking about mechanisms. *Philosophy of Science, 67*, 1–25.

MOONEY, R. J., and OURSTON, D. (1994). A multistrategy approach to theory refinement. In R. S. Michalski and G. Teccuci (Eds.), *Machine learning: A multistrategy approach* (pp. 141–164). San Mateo, Calif.: Morgan Kaufmann.

MOORE, J. K., DONEY, S. C., KLEYPAS, J. A., GLOVER, D. M., and FUNG, I. Y. (2002). An intermediate complexity marine ecosystem model for the global domain. *Deep-Sea Research II, 49*, 403–462.

OURSTON, D., and MOONEY, R. J. (1990). Changing the rules: A comprehensive approach to theory refinement. *Proceedings of the Eighth National Conference on Artificial Intelligence* (pp. 815–820). Detroit, Mich.

RICHMOND, B., PETERSON, S., and VESCUSO, P. (1987). *An academic user's guide to STELLA*. Lyme, N.H.: High Performance Systems.

SAGE, R. W., PATTEN, B. C., and SALMON, P. A. (2003). Institutionalized model-making and ecosystem-based managemeut of exploited resource populations: A comparison with instrument flight. *Ecological Modelling, 170*, 107–128.

SARMIENTO, J. L., HUGHES, T. M. C., STOUFFER, R. J., and MANABE, S. (1998). Simulated response of the ocean carbon cycle to anthropogenic climate warming. *Nature, 393*, 245–249.

TODOROVSKI, L., BRIDEWELL, W., SHIRAN, O., and LANGLEY, P. (2005). Inducing hierarchical process models in dynamic domains. *Proceedings of the Twentieth National Conference on Artificial Intelligence* (pp. 892–897). Pittsburgh, Pa.

TORRALBA, A. S., YU, K., SHEN, P., OEFNER, P. J., and ROSS, J. (2003). Experimental test of a method for determining causal connectivities of species in reactions. *Proceedings of the National Academy of Sciences, 100*, 1494–1498.

VEILLEUX, B. G. (1979). An analysis of predatory interactiou between *Paramecium* and *Didinium*. *The Journal of Animal Ecology, 48*, 787–803.

TOOLS FOR INNOVATION

Edited by

ARTHUR B. MARKMAN
KRISTIN L. WOOD

OXFORD
UNIVERSITY PRESS

2009