Title:
Supporting the Construction of Dynamic Scientific Models

Track: Emerging Applications

Authors:
Will Bridewell (willb@csli.stanford.edu)
Stuart R. Borrett (sborrett@stanford.edu)
Pat Langley (langley@csli.stanford.edu)
Computational Learning Laboratory,
Center for the Study of Language and Information,
Cordura Hall
210 Panama Street
Stanford University, Stanford, CA 94305 USA
phone: (650) 723-1684
fax: (650) 725-2166

Application Domain:
Scientific Model Discovery, Ecological Modeling

AI Techniques:
Heuristic Search, Knowledge Representation, Qualitative Physics, Theory Revision, Machine Learning

Application Status: Operational Prototype

Abstract:
In this paper, we report some challenges encountered in developing Prometheus, a software environment that supports the construction and revision of explanatory scientific models. Our responses to these challenges include the use of quantitative processes, to encode models and background knowledge, and the combination of AND/OR search through a space of model structures with gradient descent to estimate parameters. We report our experiences with Prometheus on three scientific modeling tasks and lessons learned from those efforts. We conclude by noting additional challenges that were not apparent at the outset of our work.

Supporting the Construction of Dynamic Scientific Models

Will Bridewell and **Stuart R. Borrett** and **Pat Langley**
Computational Learning Laboratory
Center for the Study of Language and Information
Stanford University, Stanford, CA 94305 USA
willb@csli.stanford.edu, sborrett@stanford.edu, langley@csli.stanford.edu

Abstract

In this paper, we report some challenges encountered in developing Prometheus, a software environment that supports the construction and revision of explanatory scientific models. Our responses to these challenges include the use of quantitative processes, to encode models and background knowledge, and the combination of AND/OR search through a space of model structures with gradient descent to estimate parameters. We report our experiences with Prometheus on three scientific modeling tasks and lessons learned from those efforts. We conclude by noting additional challenges that were not apparent at the outset of our work.

Introduction

Consider a problem scenario common to science. An ecologist is studying an aquatic ecosystem with the intent to learn how it functions. Hopefully, this knowledge will lead to a mathematical model that accurately predicts the ecosystem's response to environmental management. 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 only.

In addition to these data, the ecologist also has knowledge regarding the mechanisms operating 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. Luckily the situation is not hopeless. The scientist can 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) that is adaptable to the current ecosystem.

The described problem presents several challenges to artificial intelligence. To begin with, scientific data are often rare and difficult to obtain. The costs of collecting and preparing the data are nontrivial, 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 incremental model revision in terms of both causal structure and system 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 relationships between 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 bring it into contact with the current data.

The next challenge revolves around the need for communicable models. 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 results in models that are not easily communicated to domain scientists. We need to develop techniques for knowledge discovery where the output closely approximates the their 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 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: few scientists want to be replaced.[1] While many automated discovery systems strive to do just that, most researchers would prefer to participate in the model-building process. Thus, it behooves us to concentrate on establishing a creative partnership between computational methods and domain experts.

[1] For empirical evidence, ask one of your colleagues.

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

Figure 1: The entity type for a primary producer contains a measure of its species' concentration, growth rate, and maximum growth rate. Processes affecting the concentration will have additive influence, whereas the current growth limitation will always be the minimum produced by multiple processes. The loss rate must fall between zero and ten.

This paper 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 the knowledge representation behind the program. We then discuss the application itself, where we highlight the integration of various threads of research to compose an intelligent application. After that, 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.

Quantitative Process Models

We designed Prometheus's knowledge representation to address many of the above-mentioned challenges. In particular, the underlying language should ease communication between the scientist and the program. In this case, the software must produce models that are systems of equations and must support a style of modeling that is familiar to the users. In ecology, the resulting models often portray mechanisms, which suggests that the language of entities and the processes in which they participate (Machamer, Darden, & Craver 2000) would be 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.

We have eschewed the possibility of operating directly on the equations for two reasons. First, although systems of equations are the output of this modeling process, scientists initially work at a conceptual level. For instance, Jorgensen and Bendoricchio (2001) recommend developing a conceptual structure of the studied system as the first step in ecological modeling. They build this structure by listing the state-variables and then identifying the physical, chemical, and biological processes that link the variables to each other and the environment. They then use mathematical formulations of the processes to produce an equivalent system of equations. We want to support this modeling style that allows scientists to design the larger-scale features of the modeled system before making low-level decisions about the nature of the processes.

```
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
```

Figure 2: The generic process for exponential loss has type "loss" and takes 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.

Our second reason for adding a layer of representation on top of the equations is one of practicality. 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 most certainly contains models that fit the observed data but lack any sense of plausibility. The processes that we use contain meaningfully grouped chunks of equations that can be combined with each other to form the model. For instance, a process describing predation between species would have one equation 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, we can restrict Prometheus's search to a space of plausible models using the same type of knowledge used by systems scientists.

Both the entities and the processes in quantitative process models have two forms: generic and instantiated. A generic entity, or entity type, declares the variables and parameters that store relevant properties. We consider parameters at both the process and entity levels to be 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 explained by the model and must have associated data for purposes of comparison. And an *unobserved* variable must be given an initial value and at the generic level has a range of 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. A generic entity, such as that in Figure 1 can be instantiated by providing full definitions of the variables, depending on whether they are observed, unobserved, or exogenous, and by assigning values to the parameters.

The generic form of a quantitative process, and hence the instantiated form, was heavily influenced by qualitative process theory (Forbus 1984). For example, each generic process contains both entity roles and process roles, both of which may be optional, that are analogous to the *Individuals* field of a qualitative process. Entity roles consist of a local name for an entity along with the number and types of entities that can fill that role. A process role gives a process type and the entities to pass along to the selected subprocess. Additionally, each generic process can contain conditions similar to the *QuantityConditions* in a qualititive process that control when that process is active. Finally, the *Influences* in a qualititive process correspond to the equations in a quantitative process. As a final component, each generic process has a type that helps guide the search for plausible subprocesses. The instantiated form of a process requires the user to specify the participating entities, the subprocesses, and the parameter values. Figure 2 shows an example of a generic process.

Several aspects of the generic processes and generic entities allow us to guide search through the model space. First, the use of entity types along with entity roles constrains the viable participants in a process. Second, the bounds on parameter values helps guide estimation tools, which we will discuss later. Third, the hierarchy imposed by process roles defines an AND/OR tree of possible structures. As mentioned previously, each process has a type associated with it, and all processes of the same type can fill the same process roles during search. To illustrate, there may be a process type "growth" which has several forms (*e.g.,* exponential, logistic, limited). In this case, suppose that a top-level process called "ecosystem" has a required process role for growth. The need for such a processs constitutes an AND branch of the tree, whereas the multiple processes of the correct type flesh out the OR branch.

The creation of quantitative process models requires multiple steps. Initially, a user must develop a library of generic processes and entities. This library can be used for the creation of multiple models and updated as the need for more domain knowledge arises. Next, the user should instantiate entities and processes to form an initial model. This step may constitute a stopping point, but more likely the user 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.

Model Construction and Revision

Prometheus consists of two major components, the user interface and the model-induction engine. While each presents its own set of challenges, here we mainly discuss our approach to those of model construction and revision. However, for context we mention that Prometheus supports a wide range of interaction. At a basic level, the user can view the causal flow of a model as shown in Figure 3 or see the current state of the system of equations. The program supports model evaluation by including the ability both to simulate a model given initial conditions and to compare the trajectories to observed data. The user can also manually revise models by altering parameters as well as adding or deleting processes and entities.

The artificial intelligence behind Prometheus supports automated model construction and revision. Todorovski and colleagues (2005) describe the underlying algorithm, which operates in two distinct search spaces. The first of these involves a search through the symbolic space of model structures. Beginning with the root process, Prometheus satisfies the minimal set of constraints imposed by the hierarchy. That is, all required processes are included, and no optional processes are considered. The product of this first step is 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 model structures and guided by the sum of 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 model's sum of squared error. The core algorithm, which was designed by Bunch and colleagues (Bunch, Gay, & Welsch 1993), fits the parameters of dynamic, nonlinear systems of equations while ensuring that the resulting values fall within specified bounds. In practice, we have found this approach to be unreliable and very slow.

One of the challenges that we mentioned earlier was that scientists often begin with an initial model that they want to revise. Prometheus provides the user with several controls to influence semi-automated revision. As input, the scientist provides the 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.[2] The structural search uses the initial 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 possible moves in the search space are limited by the scientist's guidance. Upon completion, the user is given a list of the best models ranked in terms of the sum of squared error. Each of these models can be used as jumping off points for further discoveries.

Use of Prometheus can best be described by example. Consider the ecologist described at the beginning of the paper. This modeler's first step is to identify a set of generic entities and processes expected to operate within the observed ecosystem. This knowledge could be taken from an earlier developed library or created from scratch. Once this library is developed, 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 model structure in place, the ecologist can then fit the parameters using all available data. The resulting model can be simulated and the results can be compared to the observations. Now, suppose that the scientist notices that the simulated phytoplankton population fails to decrease as expected. Examination of the model shows that nothing is grazing on the phytoplankton, even though some zooplankton were observed in the area. The ecologist either can manually add the processes associated with grazing, selecting

[2]The current version of Prometheus only supports changing the parameters of processes, but development is under way that will add support for entity-level parameters.

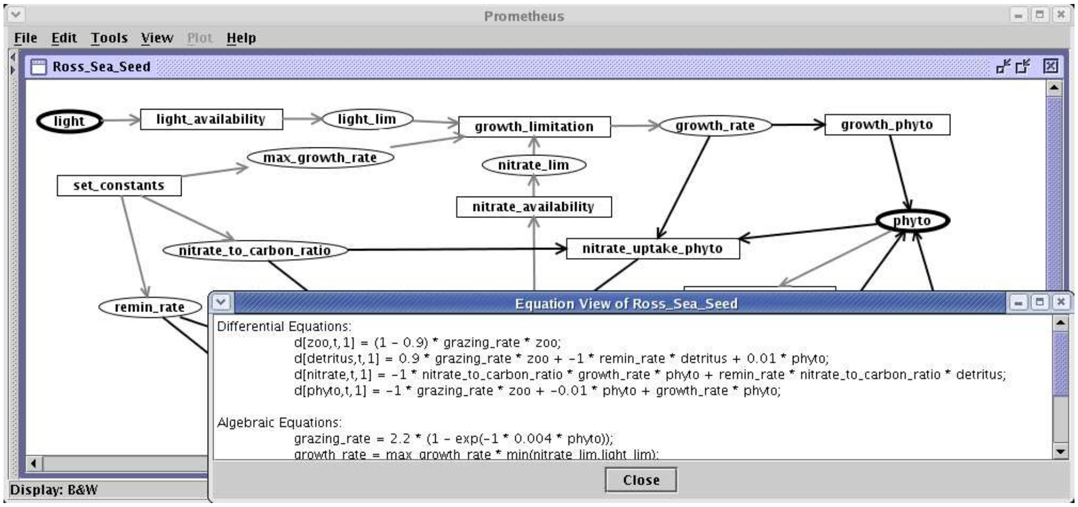

Figure 3: Prometheus can display both a causal diagram of a model and the underlying equations. In the diagram, the ovals are variables and the boxes are rectangles.

among multiple functional forms or can 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. One of these models may be selected, simulated, and evaluated by the scientist and if necessary, the revision process can continue.

Initial Experiences with Prometheus

We have evaluated Prometheus' behavior on 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 our experiences. We focus on model construction in all but the Ross Sea domain, but we have obtained similar results on the other data sets using model revision. We have reported detailed results in earlier papers (Langley *et al.* in press; Asgharbeygi *et al.* in press), 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 evaluation of our modeling environment. We focused on the protist system composed of the predator *Didinium nasutum* and the prey *Paramecium aurelia*, for which Jost and Ellner (2000) have reported concentrations at 12-hour intervals. The data are fairly smooth and demonstrate several clear cycles.

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 only their initial values. The system's search of this space produced a plausible model that included processes for growth, predation, and decay. The theoretical curves track the heights and timing of the observed trajectories quite well.

However, we encountered problems when we presented the system with the entire Jost and Ellner data, and obtained these results only when we provided it with a selected subset. The early measurements had considerably lower peaks, which suggested a different regime was operating for unknown reasons. This 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.* Clearly, human scientists have this capacity, and future versions of Prometheus from a similar capability. Providing computational support for such selective omissions is an important challenge for future research.

Population Dynamics in the Ross Sea

The Ross Sea in Antarctica involves a somewhat more complex ecosystem. Here the most important organism is phytoplankton, which undergoes repeated cycles of population increase and decrease. In this case, we had access two sets of 188 daily measurements for phytoplankton that span two successive years. Concurrent data were also available for nitrate concentrations, light levels, and ice coverage.

Based on discussions with our 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, we treated attributes of both as unobserved theoretical variables. In addition, we 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 we found 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

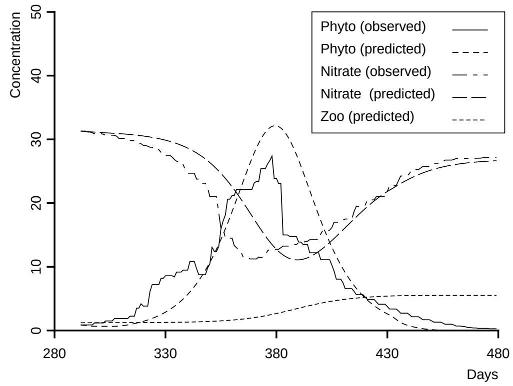

Figure 4: Performance on test data from the Ross Sea.

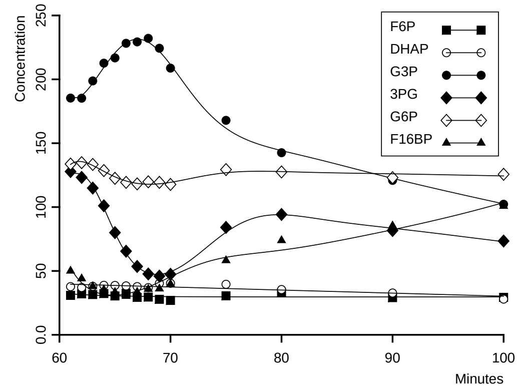

Figure 5: Observed data and predicted trajectories from biochemical kinetics.

differences between the two years. Discussion with our oceanographer led us to include another generic process which states that phytoplankton's absorb 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 4 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 claim from an ecological perspective.

The original vision for Prometheus was that it 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 have also tested Prometheus' ability in a biomedical domain— biochemical kinetics—which studies physiological changes in metabolites over time. We drew upon time-series data collected by Torralba and colleagues (2003) about the glycolysis pathway, which converts glucose into pyruvate and which plays an essential role in most life forms. They used an impulse response method that, after a biochemical system has reached steady state, briefly increases the inflow of one substance and measures its effects on others over time. We had access to 14 data points for six distinct metabolites known to be involved in glycolysis.

For this domain, we provided the system with 5 generic processes that encoded four types of metabolic reaction that appear in pathway models. These differ in how they affect positive and negative fluxes of the substances involved, with positive flux describing a metabolite's rate of flow into a reaction pathway and its negative flux specifying its rate of flow out. We crafted 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 Torralba *et al.* data and these generic processes, Prometheus searched a space of 172 distinct model and estimated parameters for each candidate. Figure 5 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 includes no inhibition or 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 models with unobserved terms, but only when given by the user. Introducing the ability to postulate new entities, although still constrained by background knowledge, would extend the system's ability to generate explanatory models that scientists find meaningful.

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 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 different techniques from artificial intelligence and system identification in response.

The formalism for quantitative process models has some clear advantages. First, it can be directly translated into a more familiar representation for the scientists, thereby addressing the challenge of communication. Second, the emphasis on domain knowledge cast as processes leads to mechanisms that explain the behavior of the system under study. 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 & 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.,* Mooney & Ourston 1994) can be applied to support interactive search, letting the user gauge the size 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 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. This can increase the search space substantially, so we need more intelligent mechanisms to guide search through the model space.

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 incredibly 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 the ideas that scientists have about 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.

Another challenge arises from the fitness measure used during search. While ecologists and other scientists often report the sum of squared error or a related number, we have been told informally that a model's ability to match the shape of observed trajectories holds more importance. For instance, scientists will tolerate some amount of phase shift, which can have strong adverse effects on the sum of square error. To combat this problem, we could introduce a quantitative measure of shape that takes the expert's evaluation of the observations into account. We believe that such a measure would have broad applicability for artificial intelligence applications that must interpret time-series data.

In summary, we have seen that Prometheus introduces a number of innovations that respond directly to the issues outlined in the introduction. These include a representation for models and background knowledge that support communication with scientists, integration of AND/OR search through a space of model structures with gradient descent search to estimate parameters, and incorporation of initial models and user input to guide revision. However, we have also seen that this combination of ideas is not sufficient to support scientists in developing models of dynamic systems. We need additional research that extends the power and flexibility of our modeling methods to more fully serve the needs of its scientific users.

Acknowledgements

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

Asgharbeygi, N.; Bay, S.; Langley, P.; and Arrigo, K. in press. Inductive revision of quantitative process models. *Ecological Modeling*.

Åström, K. J., and Eykhoff, P. 1971. System identification—a survey. *Automatica* 7:123–167.

Bradley, E.; Easley, M.; and Stolle, R. 2001. Reasoning about nonlinear system identification. *Artificial Intelligence* 133:139–188.

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.

Forbus, K. 1984. Qualitative process theory. *Artificial Intelligence* 24:85–168.

Jørgensen, S. E., and Bendoricchio, G. 2001. *Fundamentals of Ecological Modelling*. New York City, NY: 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. in press. Constructing explanatory process models from biological data and knowledge. *AI in Medicine*.

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 Michalski, R. S., and Teccuci, G., eds., *Machine Learning: A Multistrategy Approach*. San Mateo, CA: Morgan Kaufmann. 141–164.

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.

Todorovski, L.; Bridewell, W.; Shiran, O.; and Langley, P. 2005. Inducing hierarchical process models in dynamic domains. In *Proceedings of the Twentieth National Conference on Artificial Intelligence*, 892–897.

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.