Artificial Intelligence for Modeling Complex Systems: Taming the Complexity of Expert Models to Improve Decision Making

YOLANDA GIL, DANIEL GARIJO, DEBORAH KHIDER, CRAIG A. KNOBLOCK, VARUN RATNAKAR, MAXIMILIANO OSORIO, HERNÁN VARGAS, MINH PHAM, JAY PUJARA, BASEL SHBITA, and BINH VU, Information Sciences Institute, University of Southern California, Marina del Rey, CA 90292
YAO-YI CHIANG, DAN FELDMAN, YIJUN LIN, and HAYLEY SONG, Spatial Sciences Institute, University of Southern California, Los Angeles, CA 90089
VIPIN KUMAR, ANKUSH KHANDELWAL, MICHAEL STEINBACH, KSHITIJ TAYAL, and SHAOMING XU, Department of Computer Science, University of Minnesota, Minneapolis, MN 55455
SUZANNE A. PIERCE, LISSA PEARSON, and DANIEL HARDESTY-LEWIS, Texas Advanced Computing Center, University of Texas at Austin, Austin, TX 78758
EWA DEELMAN, RAFAEL FERREIRA DA SILVA, and RAJIV MAYANI, Information Sciences Institute, University of Southern California, Marina del Rey, CA 90292
ARMEN R. KEMANIAN, YUNING SHI, and LORNE LEONARD, Department of Plant Science, The Pennsylvania State University, University Park, PA 16802
SCOTT PECKHAM and MARIA STOICA, Institute of Arctic and Alpine Research, University of Colorado, Boulder, CO 80309

This research was funded by the Defense Advanced Research Projects Agency with grant no. W911NF-18-1-0027, the Planet Texas 2050 program of The University of Texas at Austin, and the National Science Foundation with grant no. ICER-1440323.

Authors' addresses: Y. Gil (corresponding author), D. Garijo, D. Khider, C. A. Knoblock, V. Ratnakar, M. Osorio, H. Vargas, M. Pham, J. Pujara, B. Shbita, B. Vu, E. Deelman, R. F. Da Silva, and R. Mayani, Information Sciences Institute, University of Southern California, Marina del Rey, CA 90292; emails: {gil, dgarijo, dkhider, knoblock, varunr, mosorio}@isi.edu, hern.var@gmail.com, minhpham@usc.edu, jpujara@isi.edu, {shbita, binhlvu}@usc.edu; Y.-Y. Chiang, D. Feldman, Y. Lin, and H. Song, Spatial Sciences Institute, University of Southern California, Los Angeles, CA 90089; emails: {yaoyic, danf, yijunlin, haejinso@usc.edu; V. Kumar, A. Khandelwal, M. Steinbach, K. Tayal, and S. Xu, Department of Computer Science, University of Minnesota, Minneapolis, MN 55455; emails: kumar@cs.umn.edu, {khand035, stei0062, tayal, xu000114}@umn.edu; S. A. Pierce, L. Pearson and D. Hardesty-Lewis, Texas Advanced Computing Center, University of Texas at Austin, Austin, TX 78758; emails: spierce@tacc.utexas.edu, Lissa.Pearson@austin.utexas.edu, dhl@tacc.utexas.edu; A. R. Kemanian, Y. Shi, and L. Leonard, Department of Plant Science, The Pennsylvania State University, University Park, PA 16802; emails: {kxa15, yshi, lnl3}@psu.edu; S. Peckham and M. Stoica, Institute of Arctic and Alpine Research, University of Colorado, Boulder, CO 80309; emails: {Scott.Peckham, Stoica}@colorado.edu; K. Cobourn and Z. Zhang,Department of Forest Resources and Environmental Conservation, Virginia Tech, Blacksburg, VA 24061; emails: {kellyc13, zzeya7}@vt.edu; C. Duffy, Department of Civil Engineering, The Pennsylvania State University, University Park, PA 16802; email: cxd11@psu.edu; L. Shu, Department of Land, Air and Water Resources, University of California Davis, Davis, CA 95616; email: shulele@lzb.ac.cn.

2160-6455/2021/06-ART11 $15.00
https://doi.org/10.1145/3453172

KELLY COBOURN and ZEYA ZHANG, Department of Forest Resources and Environmental Conservation, Virginia Tech, Blacksburg, VA 24061
CHRISTOPHER DUFFY, Department of Civil Engineering, The Pennsylvania State University, University Park, PA 16802
LELE SHU, Department of Land, Air and Water Resources, University of California Davis, Davis, CA 95616

Major societal and environmental challenges involve complex systems that have diverse multi-scale interacting processes. Consider, for example, how droughts and water reserves affect crop production and how agriculture and industrial needs affect water quality and availability. Preventive measures, such as delaying planting dates and adopting new agricultural practices in response to changing weather patterns, can reduce the damage caused by natural processes. Understanding how these natural and human processes affect one another allows forecasting the effects of undesirable situations and study interventions to take preventive measures. For many of these processes, there are expert models that incorporate state-of-the-art theories and knowledge to quantify a system's response to a diversity of conditions. A major challenge for efficient modeling is the diversity of modeling approaches across disciplines and the wide variety of data sources available only in formats that require complex conversions. Using expert models for particular problems requires integration of models with third-party data as well as integration of models across disciplines. Modelers face significant heterogeneity that requires resolving semantic, spatiotemporal, and execution mismatches, which are largely done by hand today and may take more than 2 years of effort.

We are developing a modeling framework that uses artificial intelligence (AI) techniques to reduce modeling effort while ensuring utility for decision making. Our work to date makes several innovative contributions: (1) an intelligent user interface that guides analysts to frame their modeling problem and assists them by suggesting relevant choices and automating steps along the way; (2) semantic metadata for models, including their modeling variables and constraints, that ensures model relevance and proper use for a given decision-making problem; and (3) semantic representations of datasets in terms of modeling variables that enable automated data selection and data transformations. This framework is implemented in the MINT (Model INTegration) framework, and currently includes data and models to analyze the interactions between natural and human systems involving climate, water availability, agricultural production, and markets. Our work to date demonstrates the utility of AI techniques to accelerate modeling to support decision-making and uncovers several challenging directions for future work.

CCS Concepts: • **Computing methodologies** → *Artificial intelligence*; *Ontology engineering*; *Planning and scheduling*; *Neural networks*; *Visual analytics*; *Modeling methodologies*; • **Applied computing** → *Decision analysis*; *Agriculture*;

Additional Key Words and Phrases: Intelligent user interfaces, integrated modeling, model metadata, regional-level decision-making, remote sensing data

ACM Reference format:
Yolanda Gil, Daniel Garijo, Deborah Khider, Craig A. Knoblock, Varun Ratnakar, Maximiliano Osorio, Hernán Vargas, Minh Pham, Jay Pujara, Basel Shbita, Binh Vu, Yao-Yi Chiang, Dan Feldman, Yijun Lin, Hayley Song, Vipin Kumar, Ankush Khandelwal, Michael Steinbach, Kshitij Tayal, Shaoming Xu, Suzanne A. Pierce, Lissa Pearson, Daniel Hardesty-Lewis, Ewa Deelman, Rafael Ferreira Da Silva, Rajiv Mayani, Armen R. Kemanian, Yuning Shi, Lorne Leonard, Scott Peckham, Maria Stoica, Kelly Cobourn, Zeya Zhang, Christopher Duffy, and Lele Shu. 2021. Artificial Intelligence for Modeling Complex Systems: Taming the Complexity of Expert Models to Improve Decision Making. *ACM Trans. Interact. Intell. Syst.* 11, 2, Article 11 (June 2021), 49 pages.
https://doi.org/10.1145/3453172

1 INTRODUCTION

Understanding complex systems requires developing models that can capture the underlying interacting multi-scale processes governing behaviors and analyzing possible actions that can change those behaviors to achieve desirable outcomes. Quantifying how human activities affect natural resources and how natural processes affect human life requires complex model simulations that cut across disciplinary boundaries. Population growth brings urban and industrial growth, with increased needs for water and energy. Food production through agriculture also increases the use of water, and production is greatly affected by climate, particularly droughts and flooding. Agricultural production needs to be addressed sustainably, as land use change and agrochemical inputs (e.g., fertilizers) can pollute the environment. Understanding how weather predictions and agricultural practices affect water availability and irrigation allocation, or how flooding affects planting strategies and population migration, requires integrating physics-based climate and hydrology models, biologically informed agriculture models, and socioeconomic models. The latter provides a mechanism for understanding how commodity prices and production costs, as well financial positions, affect farmers' choices and market distribution. A major challenge in integrating these models is that they can be complex, have limited software support, and can be designed for spatial and temporal operation scales that are not matched across models. As a result, it typically takes months to produce accurate results that can reveal useful insights for decision makers.

Ideally, decision makers seek integrated models that enable them to gain a causal understanding of the effects of interventions and nonaction. In addition to understanding the magnitude of intended effects, they reveal how an intervention may interact with other interventions so that the expected effect may be reduced or amplified, and they expose unintended side effects. In practice, simpler models are used that allow only the halting exploration of limited questions and do not result in effective decision-making.

There are many areas in the world where understanding complex systems that involve human activities and natural resources is crucial for decision-making. We introduce a few here. Consider Texas and the South-Central region of the United States, where the population is expected to double in the next 30 years, with concomitant urban, agricultural, and industrial growth posing increasing demands on water and energy resources. Major aquifers in the region are being depleted by hundreds of wells, reducing water reserves and causing sinking of land areas. In addition, extreme events such as extended droughts and destructive floods require accurate modeling of the potential overflow of rivers, particularly in urban areas. What levels of pumping in wells are sufficient to conserve water for expected drought periods? What areas will be safe from flooding so that infrastructure and critical services can be properly positioned? Another example is Ethiopia and other countries in Sub-Saharan Africa, where droughts and flooding affect agricultural production and food availability in a large country with limited capacity to compensate for local shortages with domestic or international trade. This results in migrations and, in extreme cases, famine. When flooding is expected, planting could be delayed in order to save seed, labor, and ultimately the crop harvest. But in what areas will flooding likely occur? What crops and under what conditions can harvest be accomplished before the floods to avoid food shortages and migration? While long-term planning for such situations is desirable, decision makers pose questions that often require rapid response in order to prepare for natural disasters or to decide on near-term policies. Creating models is effort intensive and hard to do in a timely manner. Furthermore, once the model has been created, delivering modeling systems and model outputs in a form that allows decision makers to explore scenarios and policies remains a challenge.

This article presents a modeling framework that uses artificial intelligence (AI) to make modeling more efficient and useful for decision-making. The contributions of this work include:

1. A semantic representation of models that captures different versions and settings, supports their execution, and identifies key model variables for decision-making that enable the use of modeling goals to constrain modeling choices
2. A semantic representation for datasets in terms of modeling variables that enable the use of AI planning for automated data selection and data transformations
3. An intelligent user interface that guides users through interactive scenario exploration, which allows users to state modeling goals, guides them through modeling steps, assists them by constraining choices, and automates time-consuming aspects of data preparation
4. An implemented framework that demonstrates the use of these AI techniques to analyze complex systems, designed to support a range of potential scenarios and interventions to support decision-making

Our current implementation in the MINT (Model INTegration) framework includes a range of models for weather, hydrology, agriculture, and econometrics as well as a wealth of regional-level data needed to run those models. Geospatial and temporal aspects of the data and models are central considerations for these problems. The framework can be used to analyze interactions between natural and human systems concerning water availability and food production.

The article begins describing general modeling scenarios and requirements that motivate this work. The next section discusses related work, as there is extensive literature in supporting specific aspects of modeling and decision-making. The following section introduces our approach and contributions to frame modeling problems, represent models and data, and support users in modeling tasks. The article also includes a description of the implementation of this approach in MINT and a walkthrough of the MINT user interface as a user is guided through modeling steps and receives assistance along the way. The article then summarizes the benefits of the approach and discusses, in a broader context, its merits for modeling and decision-making. The final section concludes with a summary of the novel contributions and prospects for future work.

2 MOTIVATING SCENARIOS AND REQUIREMENTS

Creating a model in a specific domain or area of expertise is very challenging because there are many models that can be adapted for specific areas or situations. Our focus has been on geoscience processes and their interaction with human processes. In this context, a model is an idealized representation of a physical system that can be used to characterize, understand, predict, and manage the system. The modeling approach can vary widely, from empirical (e.g., the likely return period of a weather event for a particular area based on prior events) to theoretical (e.g., biogeophysical laws), from having 2D to 3D spatial extent, varying spatial and temporal granularities, and different simplifications and assumptions. Although many basic modeling needs are shared in other domains, other domains may have specific requirements that are not addressed in our work. For example, synthetic biology requires the ability to compose hierarchical models of cells and their constituents, molecular dynamics simulations require atomic-level models that scale to billions of atoms, and network models require support for nondeterministic scenarios. Our focus is on supporting modeling of geoscience processes and the human processes that affect them—specifically, how floods and droughts affect crop production and food availability.

Let us consider a hypothetical scenario in which a decision maker wants to better understand how flooding in a specific area will affect roads in order to plan hunger relief shipments in case of crop failure. This complex scenario first requires understanding flooding in the area. Simple statistical models of flooding can be used to predict whether a 100-year flood could occur given a range of precipitation forecasts. While easy to use, such simple models give a binary answer:

whether there could be a 100-year flood that season or not. These simple models do not support many questions that may be of interest for decision makers, however, such as the water height in the area or drainage rate after the flood. This would require setting up more elaborate physics-based models that simulate how water would percolate through the ground and whether a river would break its banks. There are many off-the-shelf hydrology models available, but it would take some time to find one that produces useful flood maps and to understand how to use it. Often, the questions relevant to decision-making do not map very closely to the scientific questions that models were designed to answer. Thus, additional work needs to be done to understand how interventions to avoid undesirable outcomes can be represented in an off-the-shelf model (e.g., using sandbags to protect a road).

Once an appropriate hydrology model is found, using it is also challenging. It takes significant effort to locate appropriate data about the area (e.g., about terrain elevation, soil types, weather) with the right quality and granularity required by the model. It then takes time to understand and transform the data to the formats used in the target model. Although there are tools available to do data transformations, they are not well integrated and require deep expertise. Once that is done, some model parameters need to be customized to the particular area by tuning its parameters using historical data about that area. Ideally, this needs to be a systematic process independent of the user's skills. Tuning may also take many iterations and adjustments. As a result, many months of effort are typically required for this process.

These off-the-shelf models are often themselves combinations of model components that handle specific processes. For instance, the hydrology model could combine a soil component to describe how water infiltrates into the ground and a meteorological component to describe how rain falls on the land (weather models handle only the atmospheric processes leading to the precipitation). When creating a composite model of these multiple components, it is generally still the case that each component has its own input data requirements. The model takes as input a configuration file (or multiple files for each of its components), which is generally a set of key-value pairs. Values in these pairs may have any data type (e.g., integer, float, string), and some values of string type may simply be the names of other input files from which the model expects to read additional values (often as arrays of arbitrary dimensions and rank). These other input files may contain initial values for some of the model variables or forcing variables (e.g., rainfall rates) that vary in both space and time. A sophisticated spatial model may have a large number of input files in addition to its configuration file.

Each model (and associated components) is responsible for reading its own input files. Therefore, each model requires these files to adhere to supported file formats and data conventions. Models often fail if input data are not prepared correctly to meet these requirements. A model may or may not check whether all input values fall within a valid range or whether every formatting rule is followed. Instructions for data preparation are generally provided in some type of user's guide or online help system. Model developers generally assume that users have a basic understanding of numerical stability issues, such as the fact that most models become numerically unstable when their time step is set too large in relation to parameters such as grid cell size. For some models, stable time step size cannot be predicted by theory; thus, a trial-and-error approach is required. On the other hand, some models offer an automated approach to computing the time step, which can be adapted at each step.

Data preparation steps for a sophisticated model can be quite involved; spatially distributed hydrologic models provide a good illustration of this. These models carefully track all of the water in a watershed, which includes water coming in via space-time rainfall, water leaving by evaporation, water leaving by infiltration into the soil (which may be multilayered), water flowing in river channels, water flowing over hill slopes, water produced by melting snow, or water flowing into rivers from below ground (base flow). All of this water tracking is based on physical laws combined with empirical observations when necessary.

To set up the hydrology model, one would first obtain datasets that describe properties of the topography, river channels, soils, land use, and weather/climate variables (e.g., rainfall rate, temperature, relative humidity, wind speed) for the region of interest. These types of datasets can be downloaded from a variety of sources, typically as two-dimensional (2D) arrays of numbers (i.e., grids), in a variety of file formats. Mosaicking may be required to patch together grids for smaller regions to span the region of interest, given as a geographic bounding box. Clipping may also be required to match the region's bounding box. Resampling may be required to change the spatial resolution to match that of the model's computational grid. Unit conversion may be needed to obtain the measurement units expected by the model. Cleaning may be needed to deal with missing values. File format conversion may be needed. Reprojection (to a different map projection) may be needed. In addition to these types of transformations, it is often necessary to compute other required inputs from these raw inputs. For example, given a grid of elevations (a digital elevation model [DEM]) for a region, many other required grids can be computed as "derived products", such as the flow directions, river channel network, topographic slope, total contributing area and river basin boundaries. Similarly, given basic properties of the soil as grids for different soil layers, other properties can be computed from these that are needed to model the process of infiltration. Users may also need to identify points (as a longitude-latitude pair), polylines, or polygons of special interest. These are typically provided using vector versus raster file formats. Many models provide utilities with their source code to help with data preparation, but others assume that these tasks will be performed with geographic information systems (GIS) software or some other set of scripts or tools. Computational notebook interfaces provide a good mechanism for explaining and sometimes automating data preparation steps required to set up a model.

Decision-making may involve more than just flooding. In our example, a decision maker may want to know how the crop yield is affected if planting dates are moved forward to avoid the predicted floods in the farmland. This could require finding and tuning both a hydrology model and an agriculture model, necessitating twice the effort that was just described. In addition, when different models are used, their data and their assumptions must be compatible. For example, the agriculture model will be better if it uses the results of the hydrology model, such as the soil moisture over time (more moisture means more growth of the crops) and the flooding areas across the cropland over time (since flooding destroys crops). However, this requires that they use the same weather data and the same granularity in the simulation steps. This kind of coordinated modeling introduces further complexity in the modeling tasks in each of the domains involved. This severely limits the quality and timeliness of the models to the detriment of decision-making and discourages the creation of complex models in many cases because of the effort involved. The models themselves are not easy for a nonexpert to run. As a result, decision makers might be unable to explore possible situations, underutilize the power of the models, and have to rely on modelers to do specific model runs and present specific model results. The poor accessibility of the models severely limits their use by nonexperts, tilting the balance towards simpler models and perhaps more uncertain and less robust assessments.

The process we just described illustrates four major problems, which are summarized in Table 1:

1. *Delays in decision-making:* Several months are typically needed to generate modeling products. This is because data must be located and transformed as needed by models. Workarounds have to be found when data are not available, and models have to be set up and refined to make accurate predictions.

Table 1. Overview of Key Problems and Challenges in Modeling

Problems	Challenges
Delays in decision-making	• Locating data is difficult. • When data is not available, workarounds must be designed. • Transforming data to the desired formats is mostly done manually. • Customizing a model to a region is mostly a manual process.
Limited scenario exploration	• Models that consider no interventions or limited ones. • It takes significant effort to explore the side-effects of interventions.
Restricted domain modeling	• Models are designed to have dozens or hundreds of parameters, but those parameters may not need to be visible to decision makers. • Modelers lack the expertise to use models from other disciplines and ensure consistency with their own models.
Static analysis reports	• Modeling tools do not support aggregation of modeling products. • Visualizations of modeling products are generated manually.

2. *Limited scenario exploration:* Decision makers are interested in exploring interventions that solve potential problems. In contrast, many models typically focus on prediction from static inputs and have limited representations of interventions, if any.
3. *Restricted domain modeling:* While the decision maker has holistic questions about the system, in practice, models tend to have many parameters that can improve the quality of the model but are viewed as too detailed to matter for decisions. Furthermore, model parameters are often not independent; it requires substantial expertise to realize that altering one parameter means considering other parameters. When using several models, coordinating the settings of each increases modeling complexity. It takes significant effort for modelers to learn to use correctly new models outside their area of expertise.
4. *Static analysis reports:* Reports that summarize previously run executions and predefined visualizations do not adequately support exploration and understanding of nuanced patterns about the behavior of the system that are key to decision-making.

3 RELATED WORK

There is significant work on modeling frameworks, model repositories, and other infrastructure to support modelers that is reviewed in this section.

3.1 Modeling Frameworks

The Community Surface Dynamics Modeling System (CSDMS) [62] provides an open-source, community repository of earth surface process models and an integrated execution environment. The Community Earth System Model (CESM) contains atmospheric, oceanic, and land surface models. The Computational Infrastructure for Geodynamics (CIG) includes deep earth process models. The

Earth System Modeling Framework (ESMF) [31] contains models of climate, weather, and other geosciences applications. The Object Modeling System (OMS) includes mostly agricultural models [14].

Usually, the purpose of computational modeling is to predict how the values of one or more variables of interest in some system will change over time. The general approach is to start with a mathematical model of the system of interest, which is a set of equations that must be satisfied by a set of variables. These equations may be algebraic or differential, and they may be laws of physics or empirical (regression) formulas determined from data analysis. Some of them must also exhibit time dependence in order for future values to be computed from values at previous times. Time is discretized; these predictive models are often called "time-stepping models". Whenever the values of any given variable can be obtained from somewhere else—such as observational data values stored in a file or as output from another model—the number of equations the model must solve by numerical methods is reduced by one. Coupling a model to other models and datasets (stored in files) means making it possible for the model to retrieve the values of desired variables in the form that the model requires. Keep in mind that the values for a single variable that varies spatially may be stored as a large 2D or three-dimensional (3D) array of values associated with a computational grid. This coupling is often difficult because there are many ways in which the form these values are provided in can differ from the form in which they are needed by the model. For example, values may be provided on a different computational grid, with a different spatial or temporal resolution, with different measurement units, with a different variable name, with a different map projection, and so on. These differences—which result in what is often called "data friction" or "an impedance mismatch"—can be reconciled by applying transformations such as regridding, resampling, unit conversion, or reprojection. In the absence of an integrated model coupling framework, these transformations must be applied by a knowledgeable user, often as preprocessing or data preparation steps. However, sophisticated modeling frameworks such those mentioned earlier use a combination of a standardized model API and standardized metadata for variables and how they are stored in order to automatically apply these transformations with no user intervention. The modules in the framework that do these transformations are called mediators. Notice that this approach combines the well-known adapter pattern (via standardized data and model APIs, perhaps offered as services) and the mediator pattern. Mediators made available as web services are often referred to as brokers.

The modeling frameworks mentioned earlier—such as CSDMS, CESM, and CIG—support model coupling frameworks such as these, generally employ some variant of the adapter-mediator pattern, and include a repository of models.

A key distinguishing feature among model coupling frameworks is whether or not they support exchanging the (time-dependent) values of variables while the coupled models are running or whether each coupled model completes its execution before passing its values to another subsequent model. For the first type of framework, computational efficiency is critical. Thus, the time-dependent values are typically stored in RAM and passed by reference. Passing values via file I/O is typically avoided since it is so much slower. Examples of this type of framework include CSDMS, ESMF, and OMS. For the second type of framework, each model saves the values of the variables it computed in a file on exit, and the next model in the chain reads these values from that file as input before it executes. Examples of this type of framework, typically called a workflow system (e.g., [24]). Both types of framework may utilize mediators to reconcile differences between coupled models before exchanging the values of variables, such as regridding to match the model scales and message passing across models to synchronize the processes they each model.

Some computational models make extensive use of parallel processing and are meant to run on supercomputers. Exchanging values between coupled parallel models while they run is complicated by the fact that the values of a given variable are now spread across several processors. A common situation is two models that each use “domain decomposition”, but each uses a different number of processors. ESMF and the Model Coupling Toolkit (MCT) [54] both handle this problem efficiently.

The Basic Model Interface (BMI) [4] provides standardized, noninvasive, and framework-independent API for models [59]. BMI is easy to implement and yet provides all information needed to deploy a model in multiple model coupling frameworks. This is an important form of model integration, but it is not addressed in our work. While it greatly simplifies the coupling of models that must exchange data while they run, it does not address the many “upstream” issues associated with data preparation nor the many “downstream” issues associated with visualization and analysis of model results.

3.2 Model Repositories

General software repositories (e.g., GitHub [26]) help modelers store versions, test, integrate, and disseminate their code. However, these repositories represent only basic metadata, such as license, creator and installation instructions. Software container repositories (e.g., DockerHub [17]) address the execution of software with complex dependencies but also lack metadata necessary for effectively understanding the functionality of the software. None of those repositories has explicit knowledge or metadata to support reasoning and automation.

nanoHUB [94] is a repository of nanotechnology models and software. HUBzero [49] is the framework underlying nanoHUB, which has been used to develop repositories in many other domains. A key added value of the repository is the measures of quality of the software, which tracks usage statistics and citations. The focus of these repositories is to allow users to run individual models and software interactively, but users still have to figure out how to use the models and how to find and prepare the necessary data.

Some model repositories describe models using metadata organized in schemas or ontologies. The modeling frameworks mentioned in the previous section contains source code for hundreds of models. Metadata is collected for each model, with information such as authors, programming languages, pointers to code, licenses, and test datasets. A feature of CSDMS is that it enables users to describe models using standard names for model variables [81] so that the variables can be shared consistently across models. These standard names are essentially ontologies of domain-specific terms in geosciences.

Model metadata registries focus on metadata descriptions of executable models, complementing code repositories that focus on storing model code. Model metadata registries may not store the code itself but will likely have a pointer to a code repository to find it. There are multiple existing model metadata registries in different domains. [76] describe the practical experiences with a software repository for astronomy that includes hundreds of entries in which users did not want to include metadata that was hard to track and instead found beneficial that the repository identifies the code accurately and without ambiguity. OntoSoft [24] was developed to capture extensive information that is needed by scientists to understand how models work. Most of that information is available but scattered in publications, manuals, code documentation, and websites [20]. Having this information organized in a registry can save researchers a lot of time in understanding and comparing models.

Prior work on annotating executable software components with semantics has shown that it facilitates their findability and composition. In bioinformatics, initiatives such as the BioCatalog [72] led to the proliferation of dozens of semantically described services with tools for bioinformaticians. Efforts such as the SADI framework [92] leverage Semantic Web technologies to describe the inputs and outputs of components that users could combine in scientific workflow systems such as Galaxy [2], making them available to a wide range of researchers [59]. In the proteomics domain, new frameworks have been developed for automating workflow composition analysis [58]. The community has also started to apply some of these techniques to simulation models, aligning them to community-curated biomedical ontologies [34] and establishing best practices for requirements, design, and construction of biomedical simulations [33]. Our early work on the WINGS intelligent workflow system demonstrated the value of semantic metadata to automate workflow composition, interactive data and parameter selection, and validation of user-created workflows [24]. We build on all this prior work; however, the software components are consuming or generating entity identifiers (e.g., a gene name or protein name), which are easier to describe than the spatiotemporal datasets containing many physical variables that are used in geoscience models.

In summary, existing model catalogs contain useful metadata about models, and often facilitate model execution. However, they lack important information such as model variables or model processes, which are used by modelers to discern whether the model is appropriate for their analyses or not. Furthermore, once a model is selected, it takes significant effort to understand how to set it up and how to interpret its results.

3.3 Data Repositories

Data repositories are ubiquitous in science. Some notable examples are Dataverse [41] and Humanitarian Data Exchange [32]. While both provide mechanisms to describe and search datasets by the associated metadata, there are very few required types of metadata besides the high-level description (such as dataset's name and description). On one hand, it makes it straightforward for data providers to share their datasets. On the other hand, this leads to large variability in metadata quality and vocabulary, making standardization and reconciliation of datasets difficult to automate. This is addressed to some extent by the recently introduced Google Dataset Search [6] by using Schema.org [29] and W3C Data Catalog Vocabulary (DCAT) [15], which essentially offer general ontologies of core metadata for datasets. Domain-specific metadata requires effort to specify, and many data repositories capture only general metadata. Unfortunately, data catalogs generally do not require machine-readable descriptions of data, deferring that effort to data consumers.

In order to provide interoperable data, data catalogs must support many formats (e.g., XML, netCDF, spreadsheets) and layouts (e.g., relational or matrix tables). Mapping a dataset from its original format and layout into a common representation (e.g., RDF [70]) is a popular approach to address this problem. However, this mapping process is very labor-intensive and often requires users to write custom code. To accelerate this process, some users rely on tools to easily map a dataset by providing the dataset description. Methods such as RML [16], xR2RML [50], and KR2RML [79] are capable of handling datasets with heterogeneous formats such as XML and CSV, but they work with data only in the nested relational model layout. Other tools, such as XLWrap [45] and T2WML [85], can describe data in many different layouts, but they can map data only in tabular formats. There is no unified tool that can be used for these different kinds of datasets.

Creating representations of datasets generally requires a workflow of many steps, including describing the data types and data relationships. These tasks are referred to as semantic labeling and semantic modeling [28, 65, 68, 71]. Semantic labeling and semantic modeling are necessary steps to create an ontological description of a dataset. Work from the information extraction community has considered classifying and mapping tables found on webpages [8, 23, 74] as well as using similar query-based approaches for relation extraction [1, 30]. Table-understanding approaches have been described from a formal, database-centric perspective [93] focused on layout and more algorithmic approaches to understanding table layouts [42, 18]. Semantic descriptions of tables and other datasets are still largely created manually and could be more automated.

Previous research [43] has developed approaches to allow easier data representation, cleaning, and transformation. However, previous research still depends on human interaction in early data processing stages. Leveraging the power of D-REPR allows MINT to map different data formats and layouts into one common representation, allowing MINT's transformation system to be format independent. There have been multiple systems designed to tackle the data transformation/cleaning problem in the past. However, existing systems (e.g., OpenRefine [57], Wrangler [36], Trifacta [86]) support only some popular input formats (e.g., csv, json, xml) and layouts of these input data needs to follow a set of common conventions so that the content can be handled correctly. Many models use multidimensional formats, such as netCDF4 or geotiff, which are not handled by these tools.

An important task in scientific data cleaning and normalization is the identification, representation and transformation of scientific measurement units that are associated with the data. Existing frameworks such as the yt Project [87] and Measurement-units-in-R [60] give users the option to enforce a unit of measure for a given fixed set of data. These frameworks enable one to add, subtract, multiply, and divide using quantities and dimensional arrays. When used in expressions, some of these platforms automatically convert units, and simplify them when possible. Measurement-units-in-R gives the user the flexibility to expand beyond predefined units but it requires an initial user definition and understanding of data. Even with these tools, the process of data understanding, normalization, and transformation is laborious and could be more automated.

4 TECHNICAL APPROACH

Our approach has four key ideas:

1. **Goal-oriented modeling** as a principle to encapsulate model software that takes into account the questions and framing of potential interventions and decisions. These interventions and decisions are both geographically dependent and time sensitive.
2. **Modeling as problem solving**, in which modeling goals drive the selection of models and their particular configurations and settings.
3. **Representing and transforming data for scientific modeling**, in which data are described using a variety of metadata extraction and generation techniques, and data transformations to create a desired format.
4. **Interactive scenario exploration**, through a user interface that drives users through structured stages of modeling, provides dynamic visualizations of results, and generates provenance for model products to support explanation and reproducibility.

These key ideas are summarized in Table 2 and elaborated in the rest of this section.

4.1 Goal-Oriented Modeling

A major difficulty in modeling is framing the problem in terms of the desired outcomes and decisions under consideration for a specific area within a defined time frame and how they should be mapped into modeling tasks that can help in understanding possible future situations, potential interventions, and decision trade-offs. A major source of this difficulty is that this framing determines what models and data are needed and what modeling detail is required. This process takes significant effort and involves discussions about model capabilities, estimations of the effort involved in developing the models, and trade-offs between the time and effort required for modeling and the criticality of the decision.

Table 2. Approach to Integrated Modeling Taken in MINT

Approach	Key Ideas
Goal-Oriented Modeling	• Problem framing based on decision space • Models extended to expose potential interventions • Interactive dashboards to explore interventions and their outcomes
Modeling as Problem Solving	• Model metadata that enables model discovery based on modeling goals • Automated checking of model requirements and data needs • Guided model configuration and calibration
Representing and Transforming Data	• Metadata that enables data discovery • Interoperability of data • Composable data transformations • Generating novel data for modeling
Interactive Scenario Exploration	• Interactive dashboards to explore model results • Stylized narratives of modeling choices and scenarios • Provenance records with metadata of model runs

Our approach is to guide users to do goal-oriented modeling by casting their questions in terms of *modeling tasks* that capture modeling goals. We define a modeling task as a tuple:

$$\text{MT} = \langle \text{TR}, \ \text{TD}, \ \text{TI}, \ \text{TA}, \ \text{TP} \rangle.$$

where

- TR is a **response** from the system that is relevant to the decisions under consideration. System response can be estimated through a set of indicators that provide insights into the behaviors and patterns of the system under study. Indicators can be modeling variables, typically output variables of a model, or functions of modeling variables into an aggregate quantity (or index). Examples of responses of interest include crop yield and drought indices.
- TD are a set of **drivers** that enable studying different possible situations. Drivers can be input variables to the models (e.g., rainfall) or adjustable parameters that reflect changes in initial conditions. A thread could consider crop yield without flooding, another thread could consider a moderate amount of rain during the growing rainy season, and another thread major rainfall conditions.
- TI are **interventions** that represent actions that have the potential to affect outcomes. For example, a desired outcome to increase crop yield could be addressed with interventions such as planting earlier or shorter cycle crops (or a combination of both in a fraction of the area), which would allow harvesting earlier before flooding takes place. The interventions we consider are those that can be incorporated into models through specific drivers.
- TA is a **geographical area** for the model. When decisions concern an administrative region, appropriate modeling areas would be identified. For example, hydrology models would be created for specific river basins, and agriculture models would target different farms and land crop areas. Each task would focus on a single modeling area.

- TP is a **time period** for running a model for the task. This is often several years, as some models require a spin up time for the simulation that enables the model to pick up seasonal patterns in the driving variables.

Modeling tasks are often interrelated; we allow modeling tasks to be grouped thematically into *modeling problems*. Modeling problems serve the purpose of aggregating the results from different tasks. In some cases, modeling tasks may look at different modeling areas or regions. For example, a modeling problem could be to analyze how flooding will affect crop yield in a region, which may lead to several tasks to do hydrology modeling for different basins and several tasks to do agriculture modeling in several farmlands, taking into account the flooded areas. In other cases, modeling problems can be fleshed out into separate modeling tasks that each explore alternative interventions or responses (e.g., different drought indices).

The time frame of a task does not necessarily reflect the time period in which the model is run. For example, a task to analyze the effects of flooding in crop yield during the wet season may require running the agriculture model starting further back during the planting time.

Each modeling task is explored by creating different *modeling threads* that consider alternative modeling assumptions, such as the use of different models and/or variations in parameters and input datasets. For example, a modeling task to explore crop yield may have two modeling threads, each using a different agriculture model. Separate threads can be created to explore different initial conditions and input data sources (e.g., alternative weather forecasts). The use of alternative models and data sources and the comparison of results is crucial to assess uncertainty and increase confidence in the estimated outcomes.

Note that the focus of our goal-oriented modeling is not on the decision problem but rather on creating goal-oriented modeling tasks that can drive modeling and make it more efficient. We establish a relationship between modeling tasks and decision-making through the explicit representation of interventions. Much more work remains to be done to relate decisions to modeling problems and tasks.

Table 3 summarizes major concepts in goal-oriented modeling. By capturing the goals of modeling, modeling tasks drive the modeling process and constrain the choices of models and data. In the next two sections, we present how models and data are described so they can be matched with the goals represented in modeling tasks.

4.2 Modeling as Problem Solving

Once the modeling goals have been specified, we can marshal models and data in service of those goals. This section describes how models are retrieved and applied to modeling tasks.

4.2.1 Creating Problem Solving Components from Expert Models. Expert models capture sophisticated *model theories* that are often the result of years of work and are implemented in software packages, often with many versions and many possible settings. Given the software package for a model, we create software components that can be used that bundle together specific functionality in the model. First, we create *model configurations* that include specific combinations of processes and inputs required to execute a model. For example, for arid regions we may create a configuration of a hydrology model that does not include snowmelt processes. For a given configuration, we then create *model setups* that are customized for a specific scope. For instance, we may create a setup for a hydrology model that is customized for a specific river basin. Model setups are often created by adjusting model parameters using historical data, which can be automated (referred to as *model calibration* or *model parameterization*) or a manual process.

Several considerations are important to the design of model setups as problem-solving components.

Table 3. Glossary of Major Concepts in Goal-Oriented Modeling

Concept	Description
Indicators and indices	An indicator is a quantifiable variable that is identified as playing a special role, namely, to help characterize a complex property of a system being modeled. Indicators can be single variables or combinations of variables, called indices. Indices are created to summarize several indicators in an easy-to-grasp single value that can be used for assessment of alternative modeling scenarios.
Modeling problems	A modeling problem is a theme that is useful to group a set of modeling tasks. A modeling problem can be expressed as a statement and is not machine readable. It is simply a convenient mechanism to organize modeling tasks.
Models	A model in our context is a software implementation of an idealized representation of a physical system that can be used to make predictions and manage the system.
Modeling tasks	A modeling task is accomplished through a series of model runs in order to answer a question of interest.
Modeling threads	A modeling thread groups together model runs that are conceptually related.
Adjustable parameters	Parameters of a model whose value affects an input variable, and can be adjusted to explore different situations. For example, an agriculture model can have an adjustable parameter that sets the crop-to-weeds ratio (or range thereof) so that users can explore different weed growth situations.
Interventions	Interventions reflect human actions that can change the course of a system's behavior. They can be explored through the settings of adjustable parameters and input variables. For instance, interventions to improve weed management practices and increase crop yield could be studied in an agriculture model by adjusting the crop-to-weeds ratio.

First, an important aspect of creating problem-solving components out of models is exposing drivers, interventions, and adjustable parameters as inputs to the model. These require:

- **Adjustable parameters** whose variations could expose important patterns in the system and facilitate the exploration of system behaviors. In the documentation of models there is always reference to model parameters, which are used to customize models (e.g., for a specific region). Although those parameters are by definition adjustable and have a clear role in modeling, they are not necessarily important for exploring possible future situations.

Identifying appropriate adjustable parameters can be facilitated by examining examples of drivers and interventions under consideration.

- **Data inputs** that reflect possible initial states of the system as well as external variables that affect the system behavior. These are data that represent drivers and interventions of interest.
- **Responses** that can be generated by a model setup need to be identified and made explicit in the description of model results.

Second, post-processing of model results is often needed in order to support decision-making. Model outputs are typically designed to provide a scientific characterization of the system but are often not directly usable to convey patterns of behavior to nonscientists. Post-processing workflows can be associated with model setups to address the following requirements:

- **Indices** often have to be generated by combining raw model outputs with other information (e.g., drought indices) in order to provide variables and abstractions that can convey the state of the system and behavior patterns that are useful to audiences beyond the model developers. Indices are often statistical in nature and describe deviation from an average condition. For instance, a drought index value of 4 means extreme drought conditions (4 standard deviations from the mean).
- **Visualization designs** that are useful for a model need to be captured as well. This includes extracting useful variables from the inputs and outputs of the model and creating appropriate views for visualization (e.g., coarser-grained results, statistical properties, etc.). Generating proper visualizations may include fetching other data as reference (e.g., fetching historical data in order to compare the model predictions with annual averages).
- **Combined results** often need to be generated from many model executions, and these combinations are specific to each model.
- **Responses** that are not directly generated by the model but can be derived from model outputs.

Table 4 summarizes major concepts to describe models as problem-solving components. The rest of this section provides formal definitions for the terms that are used in our work.

Important modeling processes—such as model calibration, gridding, and sensitivity analysis—are not currently included in our framework. Model calibration (or parameterization) requires adjusting model parameters so that its predictions are consistent with historical data. Gridding requires setting up spatial grids of a shape (e.g., regular cubes, irregular polygons) and size (e.g., 1 km, 1 m) that allow the model to capture the physical environment with adequate granularity. Sensitivity analysis provides information about how uncertain the results would be given underdetermined parameter values. These processes can be largely automated but may require manual intervention and checking. This is a priority area for future work.

4.2.2 *Model Setups and Model Discovery.*

A model setup is a tuple:

$$\mathrm{MS} = \langle \mathrm{SC}, \mathrm{SE}, \mathrm{SA}, \mathrm{SF}, \mathrm{SP}, \mathrm{SI}, \mathrm{SO}, \mathrm{SM}, \mathrm{SN}, \mathrm{SR}, \mathrm{SS}, \mathrm{ST} \rangle$$

where

- SC are preselected input file types that are to be used with the model setup, including configuration files that specify values for some of the model parameters, and possibly also input datasets that are fixed for that setup.
- SE are preselected parameter values that are to be used with the model setup.

Table 4. Glossary of Major Concepts to Describe Models

Concept	Description
Model theories	The principles underpinning the design and implementation of the model, including physical laws, biological postulates, chemical reactions, or socioeconomic theories.
Model parameters	The parameters in the equations that express model theories. To apply the model to a specific system, model parameters are often adjusted based on the observations collected for that specific system.
Model variables	The observed or inferred quantities that can be measured or estimated about a complex system to describe its state over time. A model can have input variables, internal variables, and output variables.
Model processes	The dynamic drivers that make a system change state and, therefore, the value of its variables.
Model software	A software package that includes many different functions to set up and run a model under a variety of assumptions. A model software can have different versions.
Model configurations	A specific invocation function for model software that ensures the inclusion of certain model processes and variables while excluding others.
Model setups	The adaptation of a generic model configuration to a specific system, so that model parameters are adjusted to that system based on the observations collected about the system's past behaviors.
Adjustable parameters	A parameter of a model whose value affects an input variable and can be adjusted to explore different situations. For example, an agriculture model can have an adjustable parameter that sets the crop-to-weeds ratio (or range thereof) so that users can explore different weed growth situations.
Interventions	Human actions that can change the course of a system's behavior and can be explored through the settings of adjustable parameters and input variables.
Model files	Files that are inputs to a model or generated by a model and contain input and output variables as well as model parameters.

- SA is the set of adjustable parameters that are exposed in the model invocation signature, each specified with a valid range of values for that setup.
- SF is the set of input file types that still need to be provided in order for the setup to be executed.
- SP is the set of output file types that will contain the model results when the model is executed.
- SI is the set of input variables that are associated with SF.
- SO is the set of output variables that are associated with SP.
- SM is a set of mappings that specifies how SI, SO, and SA are represented in SF and SP.
- SN is the set of interventions that can be associated with SI.
- SR is the set of responses that can be associated with SO.
- SS is the area (or scope) where the model setup is appropriate, expressed as a polygon for geographical areas.
- ST is the time period when the model can be run, expressed as begin and end dates.

A model setup MS = <SC, SE, SA, SF, SP, SI, SO, SM, SN, SR, SS, ST> is a match for the goals of a modeling task MT = <TR, TD, TI, TA, TP> iff:

TR $\subseteq$ SR and TD $\subseteq$ SI and TI $\subseteq$ SN and

TA is geographically contained in SS and

TP contains ST

Several model setups can match any given modeling task.

Note that almost all of the elements of an MS tuple are used for goal-based modeling. In effect, they are metadata that enable model discovery.

4.2.3 Mapping Model Variables to Data. A set of mappings SM in the model setup specifies how the model variables and parameters SI, SO, and SA are represented in the input and output files (SF and SP, respectively). Each file has its own format, typically a standard such as CSV, netCDF, or shapefiles. The SM mappings expose where each modeling variable can be found within the file. In addition, it specifies a unique variable name and units required by the model.

Standard variable names are taken from the Scientific Variables Ontology (SVO) [61, 82, 83, 84], an ontology that describes thousands of variables in geosciences applications and has been mapped to other domain-specific standards such as the Climate and Forecasting Conventions and Metadata [11]. For example, a variable in a model may be informally referred to as "streamflow" while in another model it may be called "discharge", but both represent the same SVO physical variable "watershed_outlet_water__volume_flow_rate". We chose SVO among other existing ontologies (such as SWEET [69] and ENVO [7]) because it adopts a principled design of an upper ontology and naming patterns to create unique identifiers for physical variables. For example, other ontologies have a concept for "precipitation", but precipitation can be an amount, a flux, or a rate, and there are separate terms for each in SVO. SVO captures the context and relationships of variables; thus, it is not just a concept hierarchy. In addition, SVO has a clear set of principles for creating new variables in case a new concept needs to be created for one of our models. SVO variable identifiers include the physical object being measured (the water in the watershed outlet), the property of that object that is being measured (outflow rate) and the quantity (volume), in addition to other qualifiers that define the context in which a variable is used.

4.2.4 Representing Adjustable Parameters. Adjustable parameters are very important for exploring drivers and interventions. Their representations must describe any constraints that will guide users to create sensible initial conditions and explore system patterns.

An adjustable parameter is a tuple:

$$\mathrm{AP} = \langle \mathrm{PE}, \mathrm{PU}, \mathrm{PV}, \mathrm{PD}, \mathrm{PT}, \mathrm{PM} \rangle$$

where

- PE is an explanation of what the parameter represents (e.g., the parameter weed fraction is the proportion of weeds that remain after a given weeding practice)
- PU is the units for the parameter value (e.g., a fraction or percentage)
- PV is the range of values that the parameter can take, which can be a discrete set of values or a range expressed as a minimum and maximum (e.g., between 0 and 1)
- PD is the default value for the parameter (e.g., 0.25)
- PT is the parameter type
- PM is the model variables affected when the parameter is adjusted

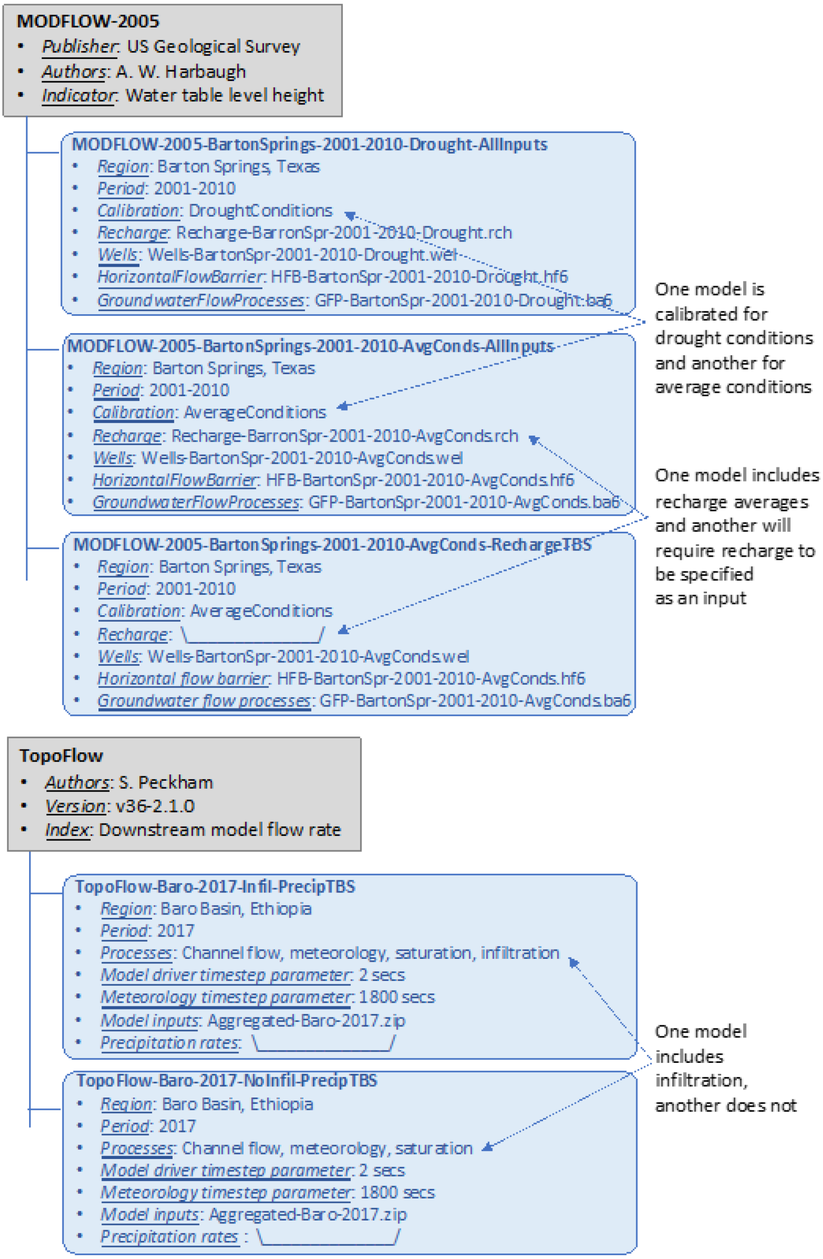
In addition, parameters are associated with interventions. For example, an adjustable parameter for weed fraction in an agriculture model may have an associated intervention representing weed control and weed management practices, where the intervention is specified by indicating in this parameter the fraction of weeds that will remain after the weed treatments are applied. A forced migration due to political instability may require weed control to be set to low due to its influence on labor availability. Weeds may fester and yields will be affected once the crops are harvested. Model inputs and model parameters allow the translation of practical questions into model operations.

4.2.5 Other Model Metadata. Besides supporting discovery, model metadata are needed for other important aspects of modeling.

Some metadata are useful for model execution. The adjustable parameters SA and the input file types SF will need to be specified by the user and, together with the preselected inputs SC and SE, will turn a model setup into an executable model, as we describe in Section 4.4. The mappings SM of variables and parameters to files also enables model execution, as these mappings are used to do automated transformations of data, described in Section 4.3.

Other metadata allow users to understand the model. This includes extensive documentation, including model authors, model assumptions (e.g., that the region is arid), model constraints (e.g., that the input weather data provides daily values), usage notes (e.g., the outputs of the model use a certain coordinate projection) and other information relevant for reproducibility and understandability. This documentation is typically scattered in publications, software manuals, and code documentation, often obtained through personal communication with model authors. Providing this documentation is crucial for usability so that users can understand the model and its uses and limitations. It is also crucial for documenting the provenance of modeling products, for reproducibility, and for future exploration of variations of the model executions.

Figure 1 illustrates how models can be characterized and differentiated through their metadata, though only a select subset of the information is shown here for space reasons. The model shown at the top is version 2005 and is used to estimate the height of the water table (i.e., the top of the aquifer). For the Barton Springs area, two calibrations were done that correspond to drought and average conditions. One of the setups of the model allows recharge to be specified, while the other has already preset inputs with average values. The model shown at the bottom is used to estimate downstream model flow rates and is version v36-2.1.0. For the Baro basin in Ethiopia, a setup



MODFLOW-2005

- *Publisher*: US Geological Survey
- *Authors*: A. W. Harbaugh
- *Indicator*: Water table level height

MODFLOW-2005-BartonSprings-2001-2010-Drought-AllInputs

- *Region*: Barton Springs, Texas
- *Period*: 2001-2010
- *Calibration*: DroughtConditions
- *Recharge*: Recharge-BarronSpr-2001-2010-Drought.rch
- *Wells*: Wells-BartonSpr-2001-2010-Drought.wel
- *HorizontalFlowBarrier*: HFB-BartonSpr-2001-2010-Drought.hf6
- *GroundwaterFlowProcesses*: GFP-BartonSpr-2001-2010-Drought.ba6

MODFLOW-2005-BartonSprings-2001-2010-AvgConds-AllInputs

- *Region*: Barton Springs, Texas
- *Period*: 2001-2010
- *Calibration*: AverageConditions
- *Recharge*: Recharge-BarronSpr-2001-2010-AvgConds.rch
- *Wells*: Wells-BartonSpr-2001-2010-AvgConds.wel
- *HorizontalFlowBarrier*: HFB-BartonSpr-2001-2010-AvgConds.hf6
- *GroundwaterFlowProcesses*: GFP-BartonSpr-2001-2010-AvgConds.ba6

MODFLOW-2005-BartonSprings-2001-2010-AvgConds-RechargeTBS

- *Region*: Barton Springs, Texas
- *Period*: 2001-2010
- *Calibration*: AverageConditions
- *Recharge*: _______________/
- *Wells*: Wells-BartonSpr-2001-2010-AvgConds.wel
- *Horizontal flow barrier*: HFB-BartonSpr-2001-2010-AvgConds.hf6
- *Groundwater flow processes*: GFP-BartonSpr-2001-2010-AvgConds.ba6

One model is calibrated for drought conditions and another for average conditions

One model includes recharge averages and another will require recharge to be specified as an input

TopoFlow

- *Authors*: S. Peckham
- *Version*: v36-2.1.0
- *Index*: Downstream model flow rate

TopoFlow-Baro-2017-Infil-PrecipTBS

- *Region*: Baro Basin, Ethiopia
- *Period*: 2017
- *Processes*: Channel flow, meteorology, saturation, infiltration
- *Model driver timestep parameter*: 2 secs
- *Meteorology timestep parameter*: 1800 secs
- *Model inputs*: Aggregated-Baro-2017.zip
- *Precipitation rates*: _______________/

TopoFlow-Baro-2017-NoInfil-PrecipTBS

- *Region*: Baro Basin, Ethiopia
- *Period*: 2017
- *Processes*: Channel flow, meteorology, saturation
- *Model driver timestep parameter*: 2 secs
- *Meteorology timestep parameter*: 1800 secs
- *Model inputs*: Aggregated-Baro-2017.zip
- *Precipitation rates* : _______________/

One model includes infiltration, another does not

Fig. 1. Characterizing and differentiating related configurations and setups of models.

of the model was calibrated that includes infiltration processes and another setup was calibrated without infiltration. Many different configurations, calibrations, and setups can be created for the same model software. All the metadata and information captured not only characterizes each of them but helps relate them to one another as well as making useful distinctions and comparisons.

In other publications, we provide more details on the rationale for characterizing models as software [25], their versions and calibrations [9], and configurations and setups [21]. In other work, we describe where modelers typically document this important information in different locations, including published articles, technical reports, code documentation, and websites [19], making it time-consuming for others to understand and compare alternative implementations.

4.3 Representing and Transforming Data for Scientific Modeling

In this section, we describe the techniques to prepare and organize the data for use in the various models. We first describe how we represent, store, and use metadata to support the search and discovery of new datasets. Next, we describe the semantic representation of the individual datasets to support search and transformations, how we automatically import new datasets to create this semantic representation, and how we automatically perform unit detection on the cells of data. Then, we describe how the system composes data transformations on the datasets using the rich semantic representations of the data. Finally, we present a method for generating derived data products from raw satellite imagery that can be used for calibrating scientific models.

4.3.1 Metadata that Enables Data Discovery. A major challenge in integrating cross-disciplinary models is the amount of effort required to locate model-appropriate data (e.g., elevation, weather, or soil type) with the right quality and granularity, both temporal and spatial. Additionally, once a dataset is found, modelers often have to go through the additional challenge of transforming the data into a format that is required by their model (e.g., cropping a global dataset to the specific region of interest, selecting only a relevant subset of the dataset's variables, or translating variables from one system of units to another).

Our approach was designed to address these issues. At a high level, it facilitates data discovery using a search and filtering mechanism based on the temporal and spatial extents of datasets as well as keywords and variable names. In contrast with existing approaches that put an onerous burden on data consumers, we take a more principled approach to data sharing whereby we require data publishers to also describe a dataset's variables in a semantically standardized manner (using SVO). The philosophy behind this approach is that data are usually produced once but consumed many more times. Therefore, by requiring data providers to go through the extra step of describing their data using a machine-readable format, it reduces the amount of effort end users would need to spend on (traditionally time-consuming) data preprocessing steps.

More concretely, we consider a dataset to be a logical grouping of data about specific system variables contained in one or more resources (i.e., a set of files in a file system, web resources, or API endpoints). The resources in a dataset share metadata such as geospatial and temporal extent and provenance. Each dataset contains information about one or more variables or scientific quantities of interest with a precise ontological definition. Variables are associated with one or more SVO names. We define variable presentations that include information about the variable's representation, such as the units of measure, handling of missing values, and metadata about collection. For each resource, we define a layout that captures the physical relationships between variables in the resource. For example, in a CSV file with columns corresponding to months and rows corresponding to different variables of interest (e.g., GDP, inflation rate, imports, exports, etc.), the layout specifies which row contains each variable and how those variables relate to the columns (time), while the variable presentation provides metadata such as units and how the variables were measured.

Raw datasets frequently contain very little context to determine their contents, often limited to a few keywords within its resources or to filenames. This presents a challenge for data discovery. Using these meager clues to determine the correct semantic data types or ontological classes pertinent to the data poses a technical challenge. We address this challenge through an augmentation-based approach that improves the alignment between ontological classes and keywords within data.

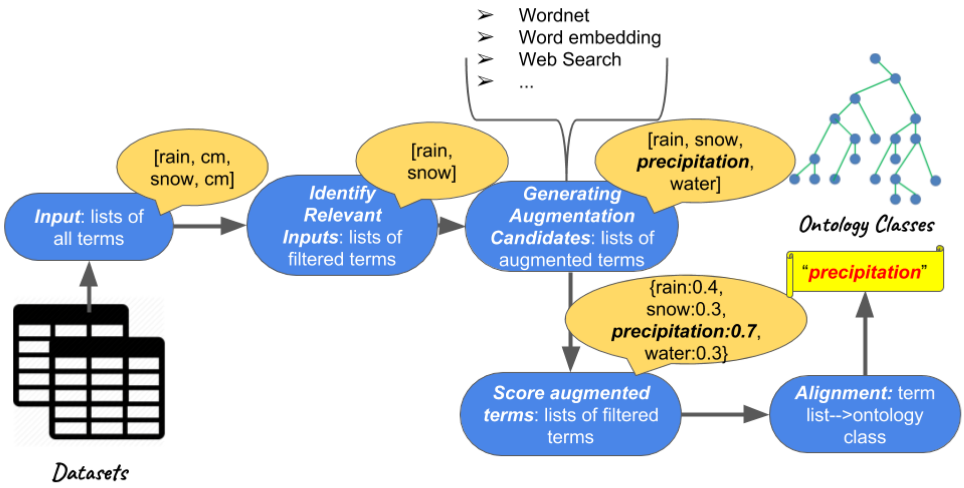

Wordnet
Word embedding
Web Search
...
Input: lists of all terms
[rain, cm, snow, cm]
Identify Relevant Inputs: lists of filtered terms
[rain, snow]
Generating Augmentation Candidates: lists of augmented terms
[rain, snow, precipitation, water]
Ontology Classes
"precipitation"
{rain:0.4, snow:0.3, precipitation:0.7, water:0.3}
Score augmented terms: lists of filtered terms
Alignment: term list-->ontology class
Datasets

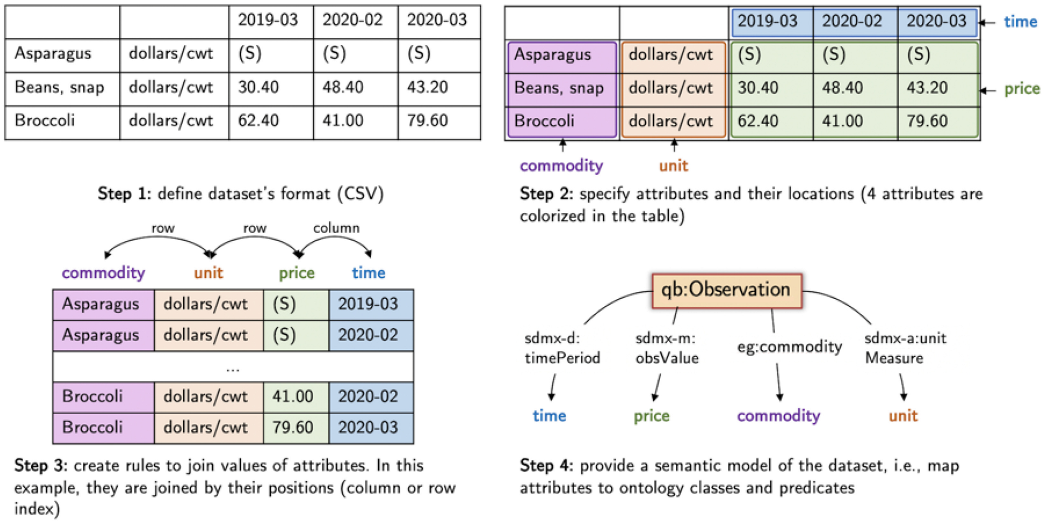
Fig. 2. Generating additional keywords for datasets through a fuzzy augmentation process.

Figure 2 illustrates the overall approach to create dataset entries using a fuzzy augmentation process. The first step of this process identifies keywords or terms within datasets. In a second step, these keywords are augmented using semantic sources and statistical techniques, generating an expanded set of dataset descriptors that are then reweighted to create the final set of keywords. The resulting keywords enable fuzzy search capabilities, where either informal keywords or technical terms used for search can be quickly matched to the relevant datasets.

Table understanding enables automated detection of headers and attributes (described in the next section), allowing the system to identify prominent keywords within the dataset. Since many extracted keywords may not provide meaningful information about ontological classes, such as stop words, units of measurement, or general metadata about collection, the system must discard some keywords. Our system uses common information retrieval filtering techniques such as term frequency–inverse document frequency (TF-IDF).

The second step uses these relevant keywords to generate additional keywords using three different sources. The first is a set of semantic resources such as WordNet [52], DBPedia [3], or ConceptNet [48], which include synonyms, hypernyms, meronyms, and relations to other concepts. The second source of keywords are statistical models such as word embedding—for example, Word2Vec [51] and GloVe [64]—and topic models [5] trained on corpora of scientific literature. The final source of keywords are queries to the World Wide Web, where a query is constructed using the dataset's keywords and results are retrieved from commercial search engines that offer public API access.

Through these three augmentation techniques, a set of augmentation candidates is generated. These candidates are reweighted based on the number of sources that support the candidate and then filtered using a similar TF-IDF technique. The final step of the fuzzy augmentation process is aligning the dataset to a user query or target ontology. Often, it is helpful to use the same augmentation techniques applied to dataset keywords to the query keywords or ontological data description. Using a combined set of weighted augmented keywords from both the data and user query, our system uses several different alignment techniques. We use well-known approaches, such as the cosine similarity and the Jaccard index, which can be extended to weighted set elements.



		2019-03	2020-02	2020-03
Asparagus	dollars/cwt	(S)	(S)	(S)
Beans, snap	dollars/cwt	30.40	48.40	43.20
Broccoli	dollars/cwt	62.40	41.00	79.60

Step 1: define dataset's format (CSV)

Step 2: specify attributes and their locations (4 attributes are colorized in the table)

commodity	unit	price	time
Asparagus	dollars/cwt	(S)	2019-03
Asparagus	dollars/cwt	(S)	2020-02
...			
Broccoli	dollars/cwt	41.00	2020-02
Broccoli	dollars/cwt	79.60	2020-03

Step 3: create rules to join values of attributes. In this example, they are joined by their positions (column or row index)

Step 4: provide a semantic model of the dataset, i.e., map attributes to ontology classes and predicates

Fig. 3. Steps for building a D-REPR model for an agricultural price dataset.

4.3.2 Interoperability of Data. A key aspect of our work to support interoperability of data is developing representations that support data integration and transformations. Public datasets are available in various formats (e.g., XML, netCDF, spreadsheets) and often have different data layouts (e.g., relational or matrix tables) and vocabularies to describe the data. To use these datasets for model calibration or prediction, we need the data to be represented in a unified and consistent way. Specifically, we use SVO and other standard ontologies (e.g., RDF Data Cube [13]) as a standard vocabulary and RDF as a unified data model. Once the datasets are virtually or materially mapped to RDF, downstream tasks such as unit transformation, cropping by a bounding box or reformatting to prepare data for running models can be easily done.

D-REPR Representation Language: To make the mapping process easier and less laborious, we developed a language called D-REPR [90] for describing datasets. Users build a D-REPR model for a dataset through four steps. First, they specify the dataset's format and then define the dataset's attributes with their locations in the dataset. As the values of an attribute after the second step are collected as an array, this results in a set of arrays, in which each array is associated with an attribute. In the third step, users specify rules to join these arrays together to form tables containing all records in the dataset. Finally, they provide the semantic meaning of each attribute and the relationships between the attributes using ontology classes and predicates. Figure 3 depicts the process of building a D-REPR model for an agricultural price dataset.

D-REPR offers several advantages over existing mapping systems such as RML [15] or XLWrap [45] or SPARQL-Generate [47]: it can model datasets in different formats and layouts and can virtually map gigantic datasets to RDF. The latter feature is critical, especially in the scientific domain. One example is that the GLDAS [27] weather dataset is stored efficiently in hundreds of GBs in netCDF, but it could be ten times bigger if stored in RDF triples. By processing the dataset using D-REPR in virtual mode, we achieve roughly the same speed as directly using netCDF without sacrificing the benefits of the RDF data model.

Automatic Table Understanding: Another key aspect of our work to support data integration is table understanding. Data representations such as D-REPR enrich data interoperability, but constructing such representations can involve significant user effort and become a barrier to adoption of data catalogs. To assist users with data curation, the MINT Data Catalog supports a

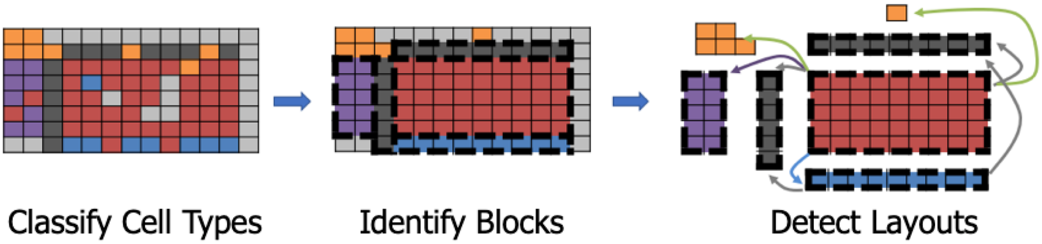


Fig. 4. Stages in table understanding.

sophisticated set of tools for automatically profiling datasets to determine the syntactic and semantic representation and then generating a D-REPR definition. Once this automated process is complete, a user may inspect the D-REPR file and correct any errors.

The tools we present are focused on understanding tabular data, which provides coverage over a large portion of scientific datasets. Figure 4 shows how we decompose the table understanding problem into three stages: cell classification, block identification, and layout detection. Each stage of the process provides a different contribution to the D-REPR definition of a dataset by identifying ontological types, data locations, and join relationships, respectively.

Cell classification assigns each data item to a label class, either at a syntactic level (floating point value) or a semantic level (mass) depending on the needs of the domain. Cell classification techniques can benefit from models for semantic labeling or semantic typing; our tools provide the semantic types used in the semantic model of the D-REPR definition. Our implementation of this task uses probabilistic graphical models, models that are able to use information of neighboring cells or headers in a table to make predictions. The blocks of functionally similar table cells are identified using cell labels.

Block identification identifies spatially contiguous regions of a table that have a similar functional role in the dataset. Identifying blocks uses information about the cell types, and our implementation adopts an entropy-based approach that evaluates candidate blocks based on the homogeneity of the cell types in the dataset. Once block boundaries are identified, they are used to determine the location of attributes in the dataset in the D-REPR definition.

The final step of table understanding is layout detection. Layout detection is formulated as a link prediction task between blocks identified in the previous stage. This task attempts to determine whether a join relationship exists between two blocks, such as when an attribute block describes a set of observations. During layout detection, these relationships can be labeled optionally with properties in a semantic model based on type information from cell classification.

We have released a flexible framework for table understanding, providing the core APIs for cell classification, block identification, and layout detection, along with reference implementations and utilities to provide visualizations of model output to assist in debugging and tools to translate input data into normalized data frames [66]. In our current architecture, these three stages are performed as a linear workflow, with earlier predictions used to influence subsequent decisions. The user can interact with the results in each of these three stages to review and make corrections as needed. In the future, we envision iteratively or jointly performing all three tasks and more sophisticated user interaction workflows.

Representing and Transforming Units: For the kinds of modeling domains that we focus on, representing and transforming units of measurement is key. The identification of measurement units that are associated with source data is a challenging task because it requires having some domain knowledge about the process that produced the data. Frequently, units appear in files within

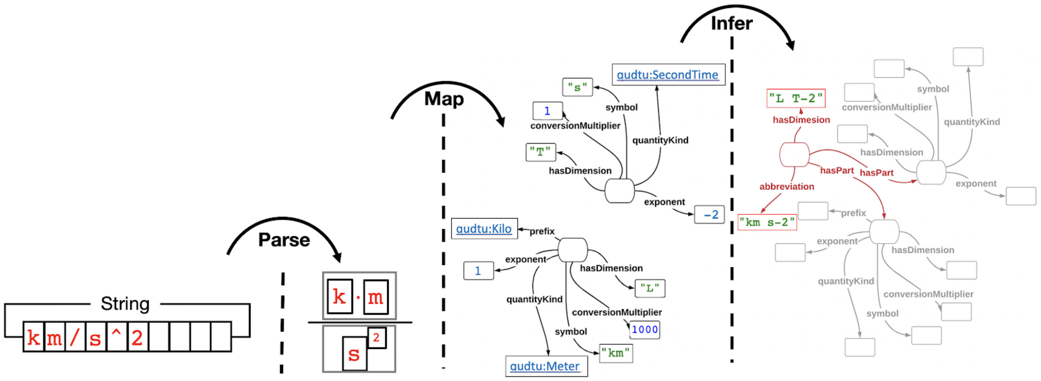


Fig. 5. The CCUT process demonstrated over a compound unit of textual format ‘km/s^2’.

datasets in a textual representation that is not easily recognized and does not carry any semantic or dimensional meaning. We developed CCUT [77], an approach that uses grammar tools to automatically parse the different components in a unit found in textual data in files and map them to elements of a standard ontology that is used extensively in geosciences called QUDT [10] to form a structured semantic output. The output depicts the different relationships, attributes, and semantics of units, which allows users to have a better understanding of their data.

The underlying pipeline behind CCUT is illustrated in Figure 5. First, we identify and parse the individual prefixes, single units, their exponents, and multipliers that compose a string of a compound unit. Then, we map each unit to its correct ontology in the schema. Finally, we compute the dimension of the compound unit and construct a normalized representation of the unit with attributes that are required for transformation. The evaluation of an early prototype has demonstrated a faster process of data analysis and understanding.

4.3.3 Composable Data Transformations. In order to combine, transform, or reformat datasets, we developed a framework called D-TRAN, which constructs a transformation pipeline based on some specification from users. The framework uses the D-REPR representation presented in Section 4.3.2 to represent the actual data that may need to be transformed into one standard format for later uses. The idea is that we use smaller components (we refer to them as adapters or building blocks), which we concatenate to form a transformation flow. This modular design allows us to reuse existing modules and wrap ready-scripts to create a format-independent module and pipeline.

In order to ensure the format independence of our system, all system inputs and outputs are described semantically using D-REPR models. Based on the D-REPR models, data in different formats will be converted into our internal D-TRAN format. Therefore, all of the transformation adapters written in our system need to work only on the D-TRAN format and, thus, can be generalized easily. The D-TRAN format supports a graph-like interface, which allows users to search and process data in an ontological manner. Using the D-REPR models, our system can serialize input data into a graph form for normal use cases or build a set of indices that map data values to their locations, allowing graph queries for high-volume datasets.

There are three types of adapters in our transformation system:

- Reader adapters are the entry point in the pipeline. A reader adapter reads a set of input files (data) and their descriptions using D-REPR, then creates a dataset in D-TRAN format.

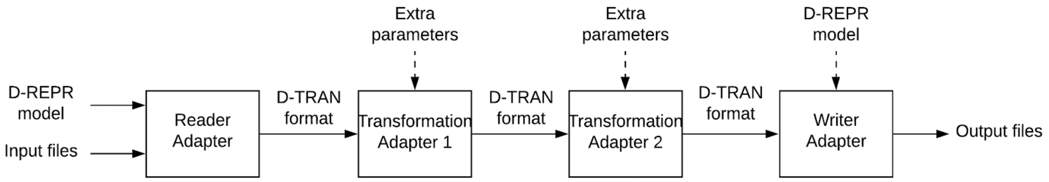


Fig. 6. General approach to data transformations.

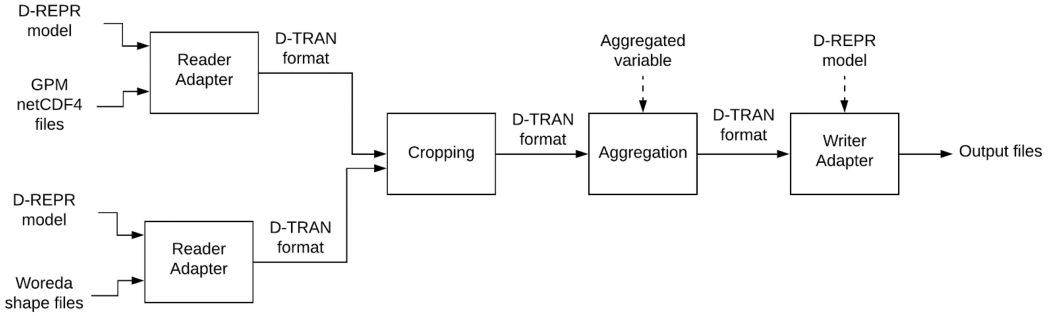


Fig. 7. A data transformation pipeline for a hydrology model.

- Transformation adapters are the main execution modules in our pipeline. A transformation adapter takes a D-TRAN dataset as input, transforms it, and outputs the resulting D-TRAN dataset.
- Writer adapters are the exit points of the pipeline. A writer adapter outputs a set of files based on its input D-TRAN format following the design specified in a D-REPR model.

Figure 6 depicts the general idea of our architecture that is based on reader, transformation, and writer adapters and components that can be concatenated. The figure shows a simplified scheme of a transformation pipeline involving a reader adapter, two transformation adapters and a writer adapter. The reader adapter takes the input files and their D-REPR model to create a dataset in D-TRAN format. Then, the dataset is transformed using two different transformation adapters. In the end, the transformed dataset will be written into files using a new D-REPR model.

Figure 7 shows a transformation pipeline for a hydrology model in which our transformation pipeline processes the global daily GPM1 weather data in 2018 in netCDF4 [88] format to create a CSV file that contains the monthly precipitation for every administrative region. First, the transformation pipeline takes both GPM netCDF4 and administrative region shape files as the input. Based on their D-REPR models, the pipeline creates two D-TRAN datasets and transfers them to later adapters. The cropping adapter crops the global spatial dataset into multiple subsets for all Ethiopia districts based on their shape files. Then, the precipitation values are aggregated for every month in the aggregation adapter. Finally, the resulting D-TRAN dataset will be materialized based on its D-REPR model.

Our approach is the first to support transformation between any type of data format. By leveraging the data representation power of D-REPR, D-TRAN can process and export data in any format while allowing users to write format-independent transformation functions.

Finally, a very important aspect of data transformations is unit conversion. Scientific units of measurement are critical when end users and non–domain experts desire to transform quantitative data. Unit conversions, which are commonly necessary in modeling world systems, can be automated using the same CCUT approach described earlier. We adapt the dimensions-based approach

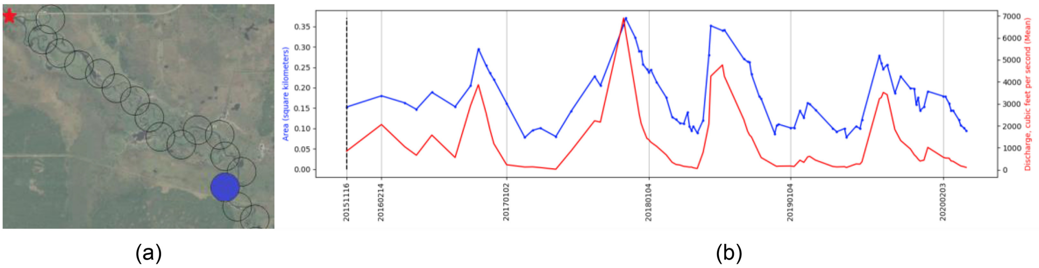


Fig. 8. An illustrative example of the potential of satellite imagery analysis to provide calibration data for hydrological models. (a) Study region: St. Johns river near Christmas, Florida. The red star on the image shows the location of the USGS gage station (ID:02232500). The blue circle shows the river segment (~8 km away from the gage station) analyzed using machine learning and satellite imagery from Sentinel 2 between 2015 and 2020. (b) Estimated river segment surface area (in blue) and daily discharge measured from the gage station (in red).

encoded in the QUDT ontology, which relates each unit to a system of base units using numeric factors. For example, any measurement of length can be expressed as a number multiplied by the unit meter (the SI base for the length dimension). Given that—and the set of exponents, prefixes and multipliers derived from the grammar and put in a structured semantic output—we are able to generate the required calculation to perform unit conversions of the same dimension. This allows a safe and fast conversion between complex compound units without requiring the user to specify conversion multipliers or numerical offsets. As described in [74], our method has been tested on spreadsheets and can be easily deployed over a range of quantitative data resources; thus, it can accelerate and improve the modeling process in any scientific domain.

4.3.4 Generating Novel Data for Modeling from Remote Sensing Sources. Physical models rely heavily on ground observations to ensure robust performance. These observations are primarily used to calibrate or customize the models for any given region. For example, hydrological models contain numerous parameters (e.g., soil conductivity at different grid points) whose values need to be calibrated for each study region with the help of observations. However, in many regions, calibration is a key challenge because these ground observations are scarce or absent. Gage stations are costly to install and maintain; thus, they are limited in number. This paucity of observational data can lead to poorly calibrated models that provide incorrect predictions or have high uncertainty in practice. For this reason, we incorporate in our approach techniques to derive new data products from raw data, which we consider as a special kind of data transformation.

Our approach is to use novel machine learning techniques to derive new data products from freely available satellite imagery data (such as Sentinel and Landsat). We describe here a method to generate river surface area dynamics. For hydrological models, the most commonly used observation is discharge (volume per second). Even though discharge cannot be estimated directly from satellite imagery, it can be approximated using surface extent of rivers. Specifically, surface extent of rivers can be used to estimate proxies for discharge if the extent estimates are available for several locations on the river at regular intervals (due to physical relationships between width and discharge). To illustrate this relationship, Figure 8 shows the comparison between surface area variation in a river segment and discharge estimates from a nearby gage station. The USGS gage station (ID:02232500) is located on St. Johns river near Christmas, Florida. A river segment ~ 8 km away was selected to compare the surface area variations with discharge estimates from the gage. The surface area estimates were created by analyzing satellite imagery data from the Sentinel-2 satellite (10 m spatial resolution, ~10-day repeat frequency). As we can see, surface area (shown

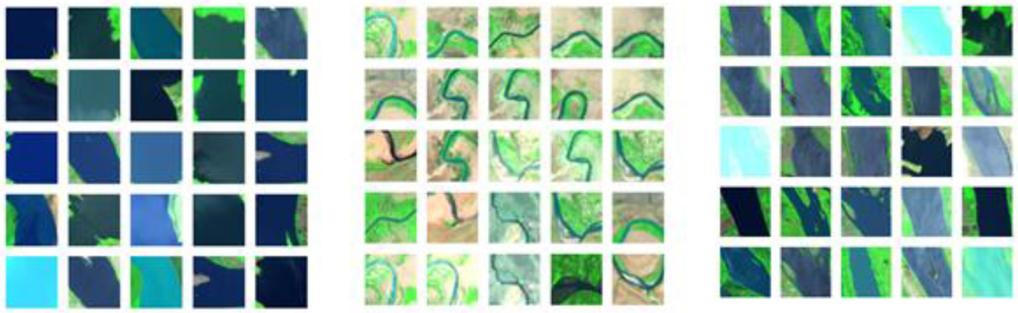


Fig. 9. An illustrative example to demonstrate the ability of auto-encoder architecture to learn inherent characteristics in the data. The three sets of images show a sample of 25 images from three different clusters that were obtained by clustering images based on the features learned by the auto-encoder architecture.

as blue time series) and discharge (shown as red time series) show a very high correlation. This illustrates the promise of our approach in using river width as a surrogate for river gauges where such calibration data is difficult to obtain.

A key challenge in deriving surface extents of river segments is the high degree of heterogeneity in spatial properties of land and water across different geographies and time, which makes it difficult for traditional pixel-based machine learning algorithms to achieve good performance [37]. These issues are exacerbated by atmospheric disturbances such as clouds, cloud shadows, and haze. In MINT, we have developed new techniques based on deep convolutional neural networks (DCNNs) to estimate the surface extents of river segments. These methods can produce spatially consistent mappings of land and water that are robust against atmospheric effects such as clouds, haze, and missing data. However, for performing accurately, these methods require a lot of training data, which is very difficult and expensive to obtain on a global scale, especially given the heterogeneity in both space and time. To address this issue, we used an auto-encoder architecture to automatically construct highly expressive features in an unsupervised setting. Specifically, we trained the auto-encoder architecture using 11,000 images (each image covered roughly a region of 1 km x 1 km) that were sampled manually from river networks around the world.

The effectiveness of this architecture can be seen in the ability of these features to cluster similar river segments as shown in Figure 9. We first used the auto-encoder architecture [46] to learn low-dimensional features. These features were then used to cluster images into 100 clusters using a k-means algorithm. The figure shows a sample of 25 images from three different clusters. As we can see, different clusters capture very different types of river segments and there is a lot of similarity among the images within a cluster.

A convolutional neural network (CNN) based on a semantic image segmentation network [73] is initialized using this unsupervised framework and then trained using 2,900 training images for which we manually constructed ground truth. This architecture was able to much better handle issues related to light haze and shadows that often confound the performance of pixel-based methods for identifying land and water pixels, as we reported in [91]. To make the paradigm more robust, we incorporated physical principles into traditional machine learning frameworks [39, 40]. Specifically, pixels of a river segment do not change independently but are related to each other through hydraulic and bathymetric constraints. These constraints can be used to identify and correct physical inconsistencies in land/water labels obtained from machine learning algorithms.

Figure 10 illustrates the utility of this physics-guided machine learning approach to obtain robust surface area estimates. Figure 9(a) shows the false color image (near infrared as red channel,

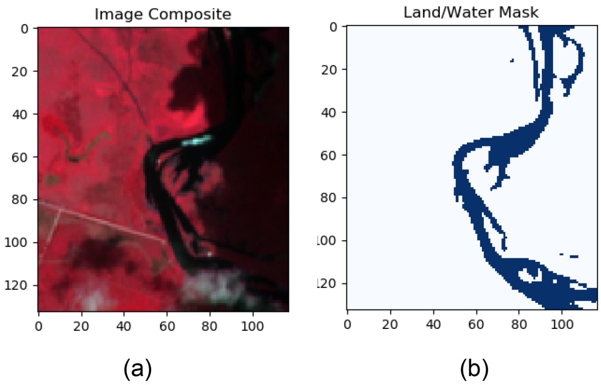
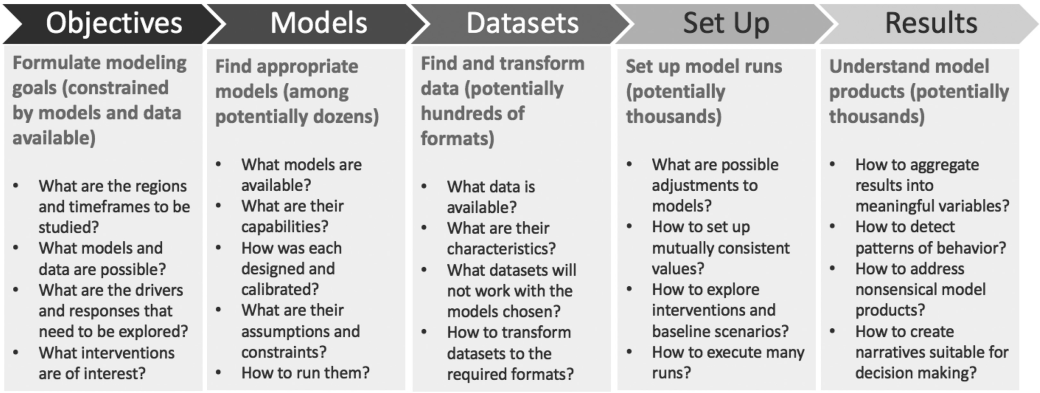


Fig. 10. An illustrative example to demonstrate the utility of our physics-guided machine learning approach. (a) False color composite of the river segment (same segment as Figure 7) on June 23rd, 2016. (b) The corresponding land/water mask that captures the surface extent despite the presence of issues such as clouds and shadows.



Formulate modeling goals (constrained by models and data available)

- What are the regions and timeframes to be studied?
- What models and data are possible?
- What are the drivers and responses that need to be explored?
- What interventions are of interest?

Find appropriate models (among potentially dozens)

- What models are available?
- What are their capabilities?
- How was each designed and calibrated?
- What are their assumptions and constraints?
- How to run them?

Find and transform data (potentially hundreds of formats)

- What data is available?
- What are their characteristics?
- What datasets will not work with the models chosen?
- How to transform datasets to the required formats?

Set up model runs (potentially thousands)

- What are possible adjustments to models?
- How to set up mutually consistent values?
- How to explore interventions and baseline scenarios?
- How to execute many runs?

Understand model products (potentially thousands)

- How to aggregate results into meaningful variables?
- How to detect patterns of behavior?
- How to address nonsensical model products?
- How to create narratives suitable for decision making?

Fig. 11. Major steps involved in modeling.

red as green channel, green as glue channel) of the river segment (same segment as Figure 8) on June 23, 2016. This band combination highlights vegetation; water appears distinctively black in color. Figure 9(b) shows the corresponding land/water mask obtained using our approach. Water pixels are shown in dark-blue; a large shadow and some clouds can be seen on the bottom of the image. Even with the presence of these occlusions, the machine learning algorithms are able to effectively estimate the surface extent. We continue to improve these algorithms and to reduce the amount of labeled data needed.

4.4 User Guidance for Interactive Scenario Exploration

Ultimately, the guidance provided to users is key to the efficient selection and use of models. This section describes how the semantic representations about models and data are used to guide users through structured stages of modeling.

4.4.1 Guiding Users Through Modeling Stages. MINT guides users through several steps, illustrated in Figure 11.

1. *Formulate modeling objectives*: This is done by specifying a modeling task consisting (as defined earlier) of drivers, responses, interventions, region, and time period.
2. *Select models*: MINT then shows users the models available that generate the indicators of interest and that have the adjustable parameters and intervention inputs desired by the analyst. Users can compare models and select one or more models to run.
3. *Select datasets*: MINT then shows users the datasets that are available as inputs to the models selected, either directly in their existing formats or that can be transformed into the formats required. Users can compare datasets and select one or more datasets to run.
4. *Set up models*: MINT shows users the adjustable parameters that are input to the model and the possible values that they can take. Users can select multiple parameter values, which result in different runs.
5. *Monitor the status of model runs*: This allows users to track model executions that take a long time and to be informed of execution failures.
6. *View results of model executions*: Users can download and save any results from models.
7. *Visualize model results*: MINT generates interactive visualizations that allow users to understand the model results.

Models and datasets are described using the techniques mentioned in Sections 4.1 and 4.2. This enables MINT to find models that are relevant to the modeling objectives defined. Once a model is selected, its data requirements result in a search for relevant datasets that either already have the model's input formats or can be transformed into those formats.

4.4.2 Model Executions and Ensembles. To run a model, a user would choose a model setup MS = <SC, SE, SA, SF, SP, SI, SO, SM, SN, SR, SS, ST> and specify a model setup assignment. A model setup assignment MA is a set of bindings B for all the files in SF and all the adjustable parameters SA for a model setup MS. The model is then executed using the setup assignment, the files and parameters in SC and SE, and the area TA and time period TP of the modeling task MT = <TR, TD, TI, TA, TP> being solved.

Once a modeling task is specified and models and data are selected, users may want to run the model under different assumptions and initial conditions.

Users often want to see how the model behaves under different assumptions or initial conditions; thus, each model is typically run many times to capture these different initial conditions. For example, a hydrology model can be run with different forecasts of rainfall (e.g., 20% less rain than the previous year, 10% less rain, 10% more rain, etc.). Therefore, it is useful to define a model ensemble as a model setup and a collection of model setup assignments that need to be executed. A collection of model setup assignments can be specified as a model ensemble specification in which the columns correspond to adjustable parameters and inputs and each row specifies the values chosen for each run. Once the model is executed, the execution results become part of the ensemble specification and each row is augmented with an additional column that links to the model outputs of that run.

4.4.3 Provenance for Explanation and Reproducibility. Provenance is key to generating explanations of model products. Extensive provenance accompanies the model runs. The provenance includes what model and software version was used plus all of the parameter values used for each run. This is summarized for the user so that the provenance of the data products is well documented. Because provenance records contain references to all model setups, data, and parameters used, they can be used to grab any information required for explanation. In that sense, provenance records serve as the basis for explaining and presenting model products to a user.

Because the provenance records are linked to specific choices during the modeling process, users can browse a provenance report and drill down to examine other alternatives and the reasons for a certain selection.

Provenance records are also useful for reproducibility in case the model needs to be re-executed with different initial situations. In addition, this enables the re-execution of the model in the future when the forecasts change.

The treatment of provenance as a mechanism for drill down to details, revisiting choices, and re-running analyses is crucial to creating interactive reports for decision-making.

4.4.4 Interactive Dashboards. MINT creates interactive dashboards with visualizations that take the results of individual model executions and aggregate the results to allow users to contrast different scenarios and interventions. Once models or model ensembles are executed, the data are reorganized by extracting relevant variables only, which are those specified in TR, TD, and TI of the modeling task at hand. This process may take time, particularly if a model ensemble contains tens of thousands of runs. MINT then generates visualizations that are designed based on the type of model and type of data.

The values specified by the user for the adjustable parameters of the model are used to create user controls in the dashboard. Therefore, when creating models, it is important to specify adjustable parameters based on what the users interacting with these visualizations would want to see.

These dashboards can be integrated within a user's report so that a decision maker can explore the results of different trade-offs and additional outcomes.

5 MODELING WITH MINT

This section provides a walkthrough of how a user interacts with MINT along with the models and data capabilities currently available.

5.1 Models and Data

Our work to date has focused on the impacts of drought and flooding in crop production in Sub-Saharan Africa as well as water availability in the southcentral region of the United States. This requires models and data that span climate, hydrology, agriculture, and economics. MINT contains a range of relevant models and datasets, including the following.

- Hydrology models to simulate water movement on the land surface, including river flow, flooding, and infiltration. These models require a large number of spatially distributed input variables that describe various properties of the topography (e.g., elevation, slope, flow direction, total contributing area), the meteorology (rainfall rate, relative humidity, air temperature, surface temperature, etc.) and the soil (including many intrinsic and hydraulic properties) and river banks (e.g., slope).
- An agriculture model that generates regional potential crop yields (e.g., maize, sorghum, wheat, sesame, cassava, teff, and peanuts) for a choice of planting dates, fertilization rates, and weed pressure levels.
- An econometrics model that represents the effect of decisions by agricultural households on estimates of crop production in a region. These decisions include subsidies for fertilizers and/or land as well as the effect of crop price.
- A groundwater model for storage and recharge of aquifers—for instance, in response to depletion of groundwater through pumping through wells and irrigation for farming.

- A drought model that uses data from several climate sources on precipitation and temperature to generate three useful drought indices based on precipitation, precipitation evapotranspiration, and evapotranspiration.
- Climate data that include precipitation, temperature, and other variables from monthly to daily frequencies. These data are extracted from sources that provide this information at global scales. Subsets of interest are automatically extracted so that they are readily available for modeling.
- Historical water levels extracted from remote sensing data, since observations from river gauges are available only for some points and only for a few years for some regions.

MINT includes other datasets needed by the models, such as soil data, digital elevation, and market prices.

Major models that we use include PIHM [78, 67], TopoFlow [63], Cycles [12, 38, 80], HAND [95], and MODFLOW [55] among others. These models are configured by experts for a variety of regions in different regional testbeds, as described here.

These models and data support a range of scenarios and interventions:

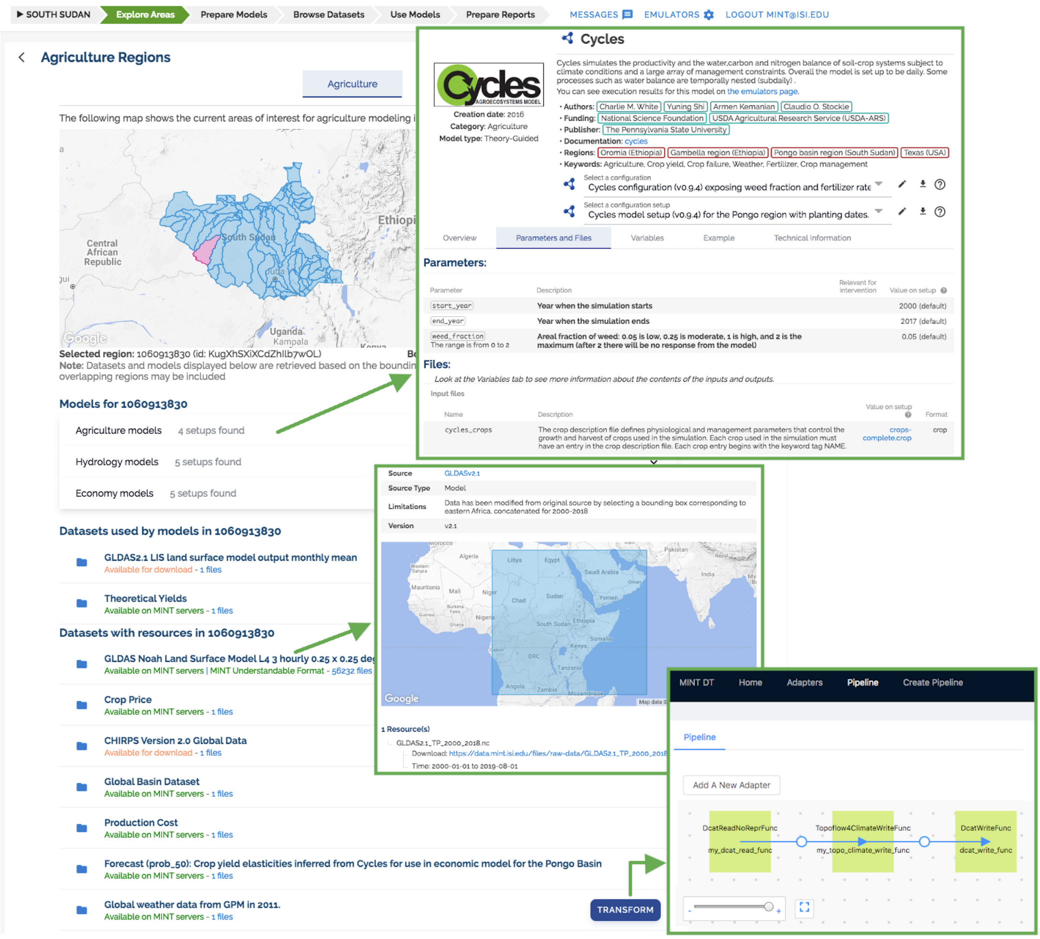
- Crop yield under different weather conditions, planting date and fertilizer choices, and weed management practices. Soil moisture affects plant growth within a given planting window. Interventions that force potential planting windows can be specified as start and end planting dates. Interventions concerning weed control and weed management practices can be reflected as a parameter for the weed fraction remaining after the weed treatments applied by farmers.
- Crop production under different farmer decisions. Interventions concerning fertilizer subsidies can be expressed as a percentage of fertilizer prices.
- Flooding under different weather conditions, with detailed flood maps that outline not only the areas that are likely to be affected by floods but the dates when flooding is likely.
- Drought severity scenarios under different weather forecasts.

Both model and dataset metadata can be edited through the MINT user interface, allowing model developers to describe their models with the desired level of detail, add configurations for new regions, or specify parameter values for manual calibration.

5.2 Interacting with MINT

The MINT user interface (UI) guides users through modeling steps, providing assistance and automation along the way. When a step has been completed, it is shown in a darker color. Users can revisit an earlier step; if the choices are changed for that step, then the subsequent steps are canceled and need to be redone. Each of these steps is done in a separate webpage and has its own URL, which allows users to share with others a particular selection or result by sharing its URL.

Figure 12 illustrates the first step (shown in green at the top left), where users can select a region and browse the models and data available. Here, the user has selected a region in South Sudan where there are several agriculture models prepared by experts for that region that can be explored (top right), and datasets that can be downloaded and transformed to run the model (bottom right). This step allows users to understand the scope of modeling capabilities available for that region, which helps them frame the modeling problem in a later step. Users can also prepare models and browse datasets, as shown at the top. The model selection, in this case Cycles, exposes the user to model input that can be changed. Preparing models is a stage in which users can fine-tune an existing model for a region by adjusting parameters with values that can improve model



The following map shows the current areas of interest for agriculture modeling i

Selected region: 1060913830 (id: KugXhSXiXCdZhlIb7wOL)

Note: Datasets and models displayed below are retrieved based on the boundin overlapping regions may be included

Models for 1060913830

Agriculture models	4 setups found
Hydrology models	5 setups found
Economy models	5 setups found

Datasets used by models in 1060913830

- GLDAS2.1 LIS land surface model output monthly mean
  Available for download - 1 files
- Theoretical Yields
  Available on MINT servers - 1 files

Datasets with resources in 1060913830

- GLDAS Noah Land Surface Model L4 3 hourly 0.25 x 0.25 deg
  Available on MINT servers | MINT Understandable Format - 56232 files
- Crop Price
  Available on MINT servers - 1 files
- CHIRPS Version 2.0 Global Data
  Available for download - 1 files
- Global Basin Dataset
  Available on MINT servers - 1 files
- Production Cost
  Available on MINT servers - 1 files
- Forecast (prob_50): Crop yield elasticities inferred from Cycles for use in economic model for the Pongo Basin
  Available on MINT servers - 1 files
- Global weather data from GPM in 2011.
  Available on MINT servers - 1 files

Cycles

Creation date: 2016
Category: Agriculture
Model type: Theory-Guided

Cycles simulates the productivity and the water,carbon and nitrogen balance of soil-crop systems subject to climate conditions and a large array of management constraints. Overall the model is set up to be daily. Some processes such as water balance are temporally nested (subdaily) .

You can see execution results for this model on the emulators page.

- **Authors:** Charlie M. White, Yuning Shi, Armen Kemanian, Claudio O. Stockle
- **Funding:** National Science Foundation, USDA Agricultural Research Service (USDA-ARS)
- **Publisher:** The Pennsylvania State University
- **Documentation:** cycles
- **Regions:** Oromia (Ethiopia), Gambella region (Ethiopia), Pongo basin region (South Sudan), Texas (USA)
- **Keywords:** Agriculture, Crop yield, Crop failure, Weather, Fertilizer, Crop management

Select a configuration: Cycles configuration (v0.9.4) exposing weed fraction and fertilizer rate

Select a configuration setup: Cycles model setup (v0.9.4) for the Pongo region with planting dates.

Parameters:

Parameter	Description	Relevant for intervention	Value on setup
start_year	Year when the simulation starts		2000 (default)
end_year	Year when the simulation ends		2017 (default)
weed_fraction The range is from 0 to 2	Areal fraction of weed: 0.05 is low, 0.25 is moderate, 1 is high, and 2 is the maximum (after 2 there will be no response from the model)		0.05 (default)

Files:

Look at the Variables tab to see more information about the contents of the inputs and outputs.

Input files

Name	Description	Value on setup	Format
cycles_crops	The crop description file defines physiological and management parameters that control the growth and harvest of crops used in the simulation. Each crop used in the simulation must have an entry in the crop description file. Each crop entry begins with the keyword tag NAME.	crops-complete.crop	crop

Source	GLDASv2.1
Source Type	Model
Limitations	Data has been modified from original source by selecting a bounding box corresponding to eastern Africa, concatenated for 2000-2018
Version	v2.1

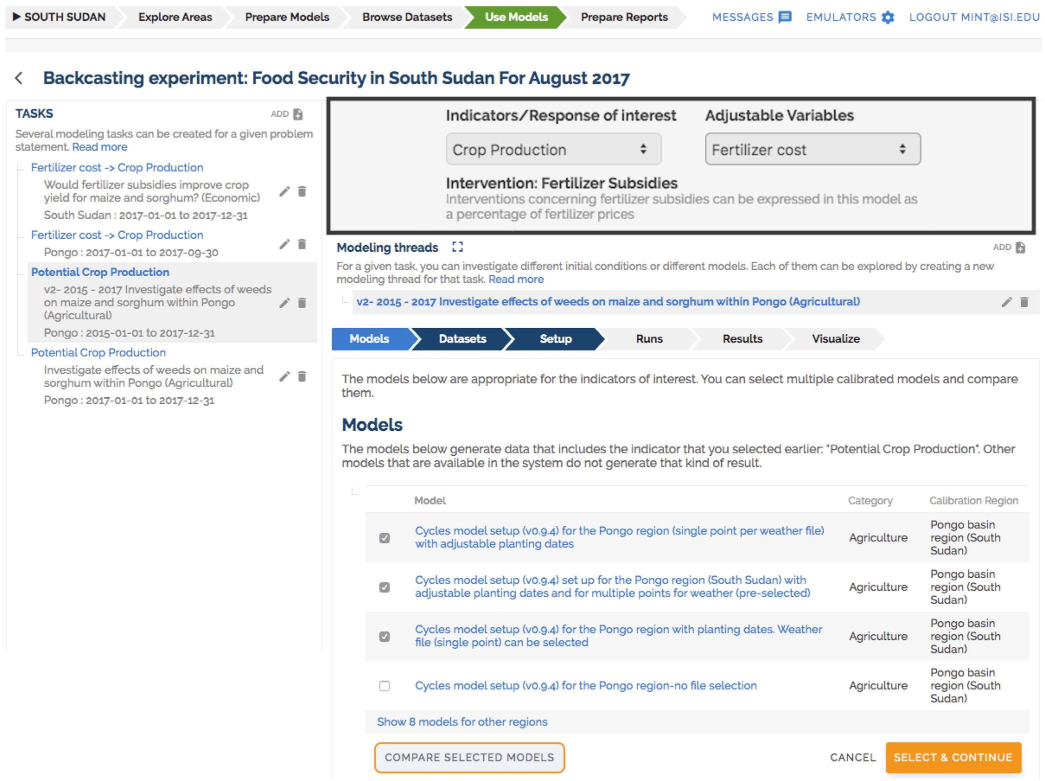
1 Resource(s)
GLDAS2.1_TP_2000_2018.nc
Download: https://data.mint.isi.edu/files/raw-data/GLDAS2.1_TP_2000_2018
Time: 2000-01-01 to 2019-08-01

Fig. 12. MINT User Interface: Exploring data and models available.

accuracy for that region. Repeating the process for multiple models or with the same model but different data sources while seeking the same output variables provides the user with a quick, but not systematic, way of estimating uncertainty. Output convergence increases confidence and output divergence suggests that further exploration is needed.

The next stage is to use the models and generate results. Before setting up and running models, MINT asks users to frame the modeling problem. Figure 13 shows that the user has specified modeling tasks (left), and for each task has defined variables of interest (top right). These include an indicator or response of interest as well as input variables that can be specified by the user; MINT notes the possibility of exploring interventions through adjustments to the input variables. Next, the user creates modeling threads to explore alternative problems related to the task, possibly through different data inputs, different models, or different parameters and interventions. For each thread, MINT guides users through several substeps as shown in the figure, with the first substep highlighted to indicate that the user needs to start by choosing the models to be used. MINT shows the models that are relevant to the region and have variables specified in the task.

Figure 14 illustrates the next two substeps. The user selects input datasets for the model selected (left side of the figure). Similar to models, MINT shows datasets that are compatible with the



Backcasting experiment: Food Security in South Sudan For August 2017

TASKS

Several modeling tasks can be created for a given problem statement. Read more

Fertilizer cost -> Crop Production
Would fertilizer subsidies improve crop yield for maize and sorghum? (Economic)
South Sudan : 2017-01-01 to 2017-12-31

Fertilizer cost -> Crop Production
Pongo : 2017-01-01 to 2017-09-30

Potential Crop Production
v2- 2015 - 2017 Investigate effects of weeds on maize and sorghum within Pongo (Agricultural)
Pongo : 2015-01-01 to 2017-12-31

Potential Crop Production
Investigate effects of weeds on maize and sorghum within Pongo (Agricultural)
Pongo : 2017-01-01 to 2017-12-31

Indicators/Response of interest: Crop Production

Adjustable Variables: Fertilizer cost

Intervention: Fertilizer Subsidies
Interventions concerning fertilizer subsidies can be expressed in this model as a percentage of fertilizer prices

Modeling threads

For a given task, you can investigate different initial conditions or different models. Each of them can be explored by creating a new modeling thread for that task. Read more

v2- 2015 - 2017 Investigate effects of weeds on maize and sorghum within Pongo (Agricultural)

The models below are appropriate for the indicators of interest. You can select multiple calibrated models and compare them.

Models

The models below generate data that includes the indicator that you selected earlier: "Potential Crop Production". Other models that are available in the system do not generate that kind of result.

	Model	Category	Calibration Region
☑	Cycles model setup (v0.9.4) for the Pongo region (single point per weather file) with adjustable planting dates	Agriculture	Pongo basin region (South Sudan)
☑	Cycles model setup (v0.9.4) set up for the Pongo region (South Sudan) with adjustable planting dates and for multiple points for weather (pre-selected)	Agriculture	Pongo basin region (South Sudan)
☑	Cycles model setup (v0.9.4) for the Pongo region with planting dates. Weather file (single point) can be selected	Agriculture	Pongo basin region (South Sudan)
□	Cycles model setup (v0.9.4) for the Pongo region-no file selection	Agriculture	Pongo basin region (South Sudan)

Show 8 models for other regions

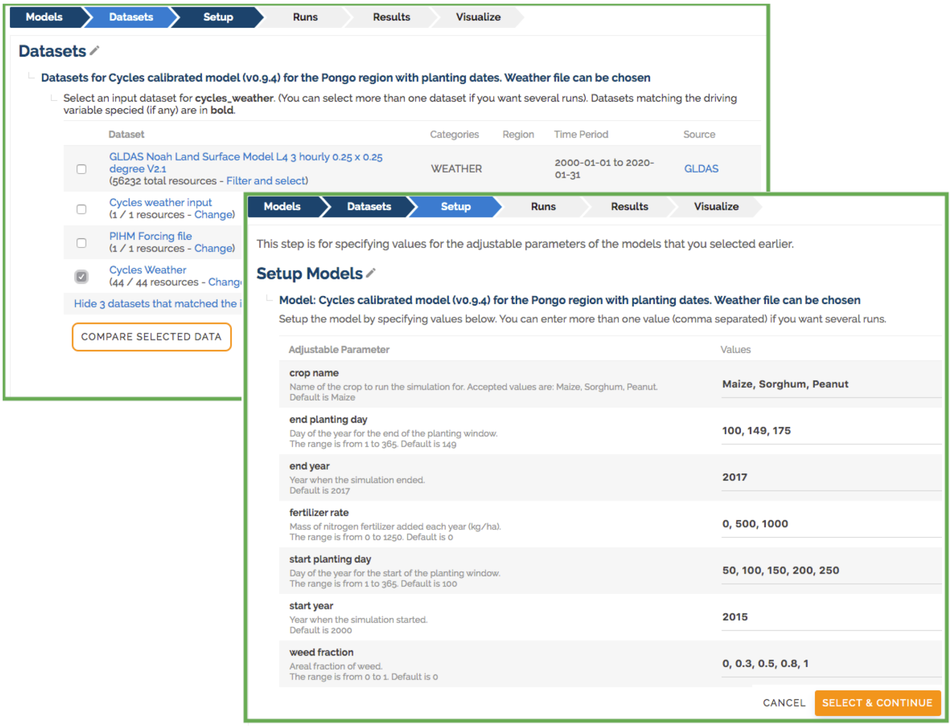
Fig. 13. MINT User Interface: Specifying modeling tasks.

model selected. Users can compare their main features and do more detailed exploration if needed. Next, the user sets up parameter values for the model selected (right side of the figure). For each parameter, several possible values can be specified, and MINT will run all combinations of values. Note that these parameters are not all model parameters but rather a carefully selected subset of parameters that are of interest for decision-making. Each combination of parameter values is submitted as a model execution.

The next substep allows a user to track model executions and note any execution failures due to missing or low-quality data in input files, inconsistencies in parameter settings, and other modeling issues. At the moment, MINT proceeds with the executions that are successful and leaves it to the user to work on repairs to any of the failures. These repairs may be as simple as fixing a missing value in an input file or adjusting some parameters in the model to be mutually consistent or, in some cases, rejecting an input dataset because it is not appropriate for the model. With the validation that we have done for our models in our regions of interest, execution failures are rare. However, execution failures will not be uncommon when using models in new regions. This is an area of future work, in which we plan to extend MINT with knowledge about model constraints that can be used to avoid these execution failures or to advise users on how to address them.

In the substeps that follow, the user can examine and visualize the model execution results. Users may browse and download model outputs and use them in custom visualizations or download them.

Figure 15 shows an example of a visualization dashboard in MINT. The first and second visualizations show the results from an agriculture model that estimates crop yield for several different



Datasets

Datasets for Cycles calibrated model (v0.9.4) for the Pongo region with planting dates. Weather file can be chosen

Select an input dataset for **cycles_weather**. (You can select more than one dataset if you want several runs). Datasets matching the driving variable specied (if any) are in **bold**.

	Dataset	Categories	Region	Time Period	Source
□	GLDAS Noah Land Surface Model L4 3 hourly 0.25 x 0.25 degree V2.1 (56232 total resources - Filter and select)	WEATHER		2000-01-01 to 2020-01-31	GLDAS
□	Cycles weather input (1 / 1 resources - Change)				
□	PIHM Forcing file (1 / 1 resources - Change)				
☑	Cycles Weather (44 / 44 resources - Chang				

Hide 3 datasets that matched the i

This step is for specifying values for the adjustable parameters of the models that you selected earlier.

Setup Models

Model: Cycles calibrated model (v0.9.4) for the Pongo region with planting dates. Weather file can be chosen

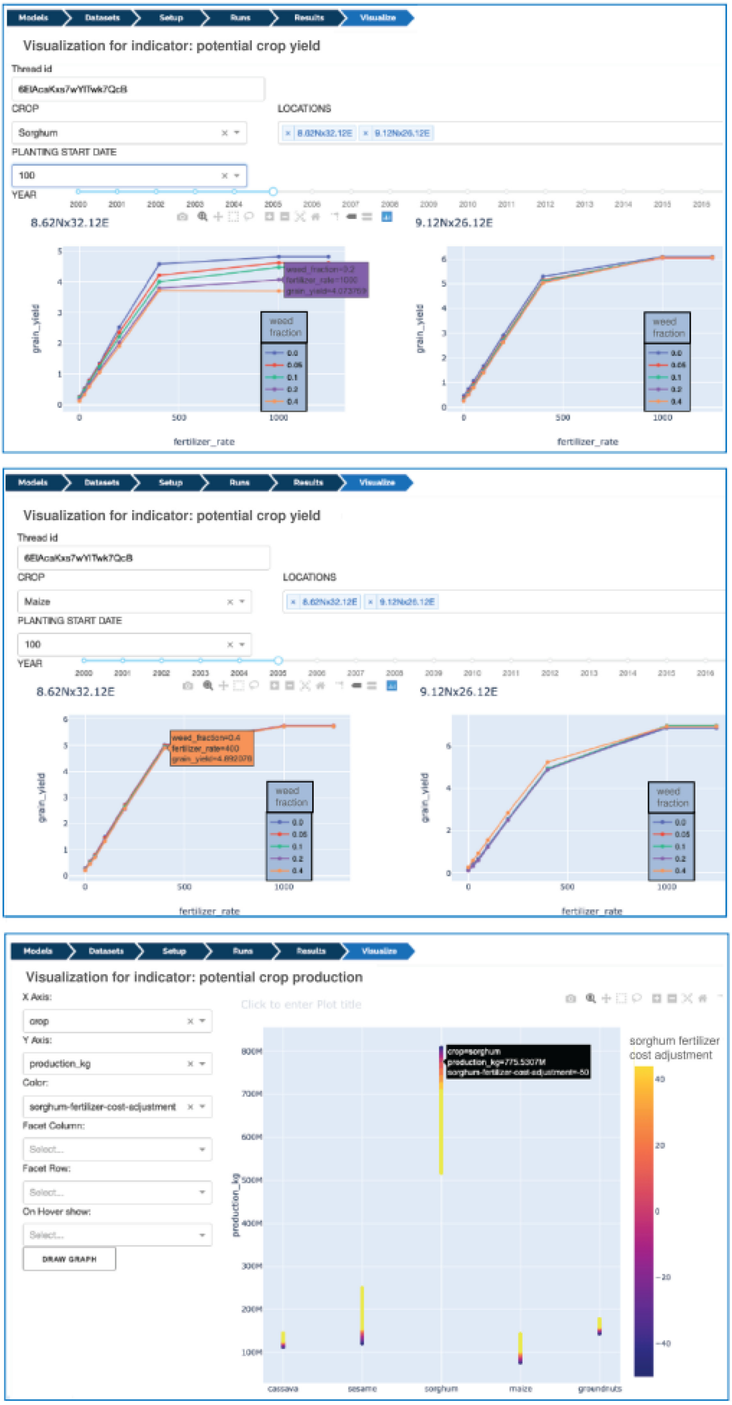
Setup the model by specifying values below. You can enter more than one value (comma separated) if you want several runs.

Adjustable Parameter	Values
crop name Name of the crop to run the simulation for. Accepted values are: Maize, Sorghum, Peanut. Default is Maize	Maize, Sorghum, Peanut
end planting day Day of the year for the end of the planting window. The range is from 1 to 365. Default is 149	100, 149, 175
end year Year when the simulation ended. Default is 2017	2017
fertilizer rate Mass of nitrogen fertilizer added each year (kg/ha). The range is from 0 to 1250. Default is 0	0, 500, 1000
start planting day Day of the year for the start of the planting window. The range is from 1 to 365. Default is 100	50, 100, 150, 200, 250
start year Year when the simulation started. Default is 2000	2015
weed fraction Areal fraction of weed. The range is from 0 to 1. Default is 0	0, 0.3, 0.5, 0.8, 1

Fig. 14. MINT User Interface: Selecting model inputs, both datasets and parameters.

locations within the Western Flood Plains of South Sudan. An ensemble was set up that varied the location, year, start planting date, and weed fractions for several crops. The graphs in the figure show crop yield for sorghum and maize in the year 2005 for various amounts of fertilizers, with varying weed fractions shown as different colored lines. Users can slide over different years, plotting the potential crop yield with different weed fractions. Interacting with this dashboard shows that (1) the response to the use of fertilizer varies greatly for different locations, and (2) the weather in 2017 leads to a much lower yield overall than other years. The third visualization in the figure shows the results from an economic model that generates potential crop production by considering the potential crop yield from the agriculture model, projected market prices for different crops, and fertilizer cost (which can be affected through an intervention such as fertilizer subsidies) to analyze what crops farmers will most likely be planting. Through interaction with the visualization dashboard, users can easily see that (1) higher subsidies for sorghum fertilizer will increase sorghum production and decrease production of all other crops; (2) if sorghum prices fall, production for all other crops will increase; and (3) sorghum has the highest range of possible production outcomes depending on the chosen intervention.

Finally, users start to prepare reports for decision makers and can get from MINT information extracted from the provenance records for each thread. Figure 16 shows a summary of all of the details involved in generating a particular model output or visualization. Users will add many additional aspects to their reports—notably, the user's comments about the findings based on the modeling results. What MINT offers is a sound justification for how the results were obtained and the means to quickly update the results when more data or models become available.



Visualization for indicator: potential crop yield

Thread id: 6ElAcaKxs7wYlTwk7QcB

CROP: Sorghum

LOCATIONS: 8.62Nx32.12E, 9.12Nx26.12E

PLANTING START DATE: 100

YEAR

Visualization for indicator: potential crop yield

Thread id: 6ElAcaKxs7wYlTwk7QcB

CROP: Maize

LOCATIONS: 8.62Nx32.12E, 9.12Nx26.12E

PLANTING START DATE: 100

YEAR

Visualization for indicator: potential crop production

X Axis: crop

Y Axis: production_kg

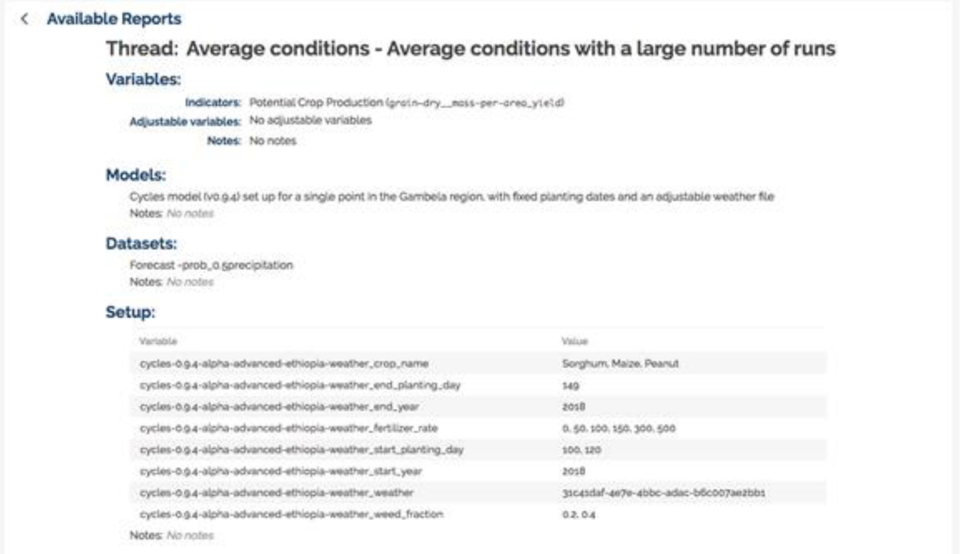
Color: sorghum-fertilizer-cost-adjustment

Facet Column: Select...

Facet Row: Select...

On Hover show: Select...

Fig. 15. Interactive dashboards in MINT.



< Available Reports

Thread: Average conditions - Average conditions with a large number of runs

Variables:

Indicators: Potential Crop Production (grain-dry__mass-per-area_yield)
Adjustable variables: No adjustable variables
Notes: No notes

Models:

Cycles model (v0.9.4) set up for a single point in the Gambela region, with fixed planting dates and an adjustable weather file
Notes: *No notes*

Datasets:

Forecast -prob_0.5precipitation
Notes: *No notes*

Setup:

Variable	Value
cycles-0.9.4-alpha-advanced-ethiopia-weather_crop_name	Sorghum, Maize, Peanut
cycles-0.9.4-alpha-advanced-ethiopia-weather_end_planting_day	149
cycles-0.9.4-alpha-advanced-ethiopia-weather_end_year	2018
cycles-0.9.4-alpha-advanced-ethiopia-weather_fertilizer_rate	0, 50, 100, 150, 300, 500
cycles-0.9.4-alpha-advanced-ethiopia-weather_start_planting_day	100, 120
cycles-0.9.4-alpha-advanced-ethiopia-weather_start_year	2018
cycles-0.9.4-alpha-advanced-ethiopia-weather_weather	31c41daf-4e7e-4bbc-adac-b6c007ae2bb1
cycles-0.9.4-alpha-advanced-ethiopia-weather_weed_fraction	0.2, 0.4

Notes: *No notes*

Fig. 16. Showing provenance of model results.

Individual models can be combined to create integrated models of complex systems. For example, hydrology model outputs can be used as input in the daily time step of agriculture models provided that these models have a means to uptake such data. In such cases, the input weather data sources have to be the same for both models in order for the model outputs to be meaningful. Another example of model integration is the use of agriculture model outputs to estimate crop yield elasticity as a factor for modeling actual crop production. Combining models, even in simple ways, enables powerful integrated modeling and opens the door to causal reasoning about interventions, their intended consequences, and potential undesirable side effects.

5.3 Accessibility

We show here two focal testbeds that we have created to support our research.

5.3.1 Sub-Saharan Africa. Food shortages may result in human migration and displacement. With droughts reducing water availability and floods destroying crop fields, many areas have food shortages and food insecurity for large populations. Some countries have limited capacity to compensate for local shortages with domestic or international trade, which results in migrations and, in extreme cases, famine. When flooding is expected, planting could be delayed in order to save seed, labor, and ultimately the crop harvest. But in what areas will flooding likely occur? For what crops and under what conditions can the harvest be accomplished before the floods to avoid food shortages and migration? While long-term planning for such situations is desirable, decision makers pose questions that are often short fused in order to prepare for natural disasters or to decide on near-term policies. Delivering modeling systems and outputs in a form that allows decision makers to explore scenarios and policies remains a challenge.

We have been developing models for Ethiopia and South Sudan, including hydrology models for major river basins and agriculture models for large administrative regions and smaller administrative units. MINT contains many models and model outputs for major regions of interest. Most

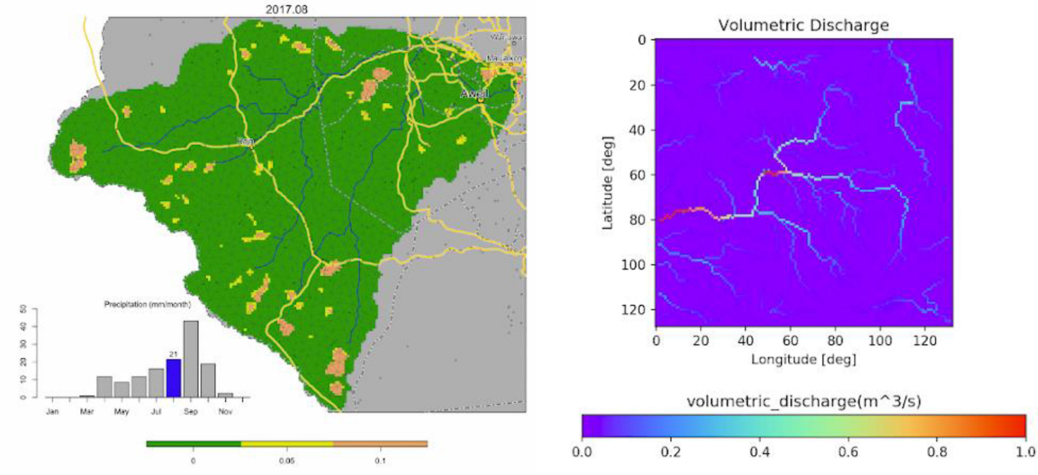


Fig. 17. Models of flooding in South Sudan and Ethiopia.

models focus on the Oromia and Gambella regions of Ethiopia, and the Western Flood Plains of South Sudan.

This testbed currently includes 26 model configurations and 95 model setups. It includes 297 datasets with over 2.46 million resources for relevant regions of Sub-Saharan Africa. MINT has models and data for different administrative regions of Ethiopia, all the way down to woredas that make independent land allocations and other decisions that affect crop production. MINT also has models for major river basins, in particular for Baro, Muger, Guder, Bashilo, Beko-Tippi, Ganale, Tezeke, Shebelle, Awash, and Jamma. The models were run for backcasting for the past decade, using alternative weather sources. A total of 33,412 model runs have been executed to date, with more than 3.1 Terabytes of model outputs.

Figure 17 shows an example of two models of flooding, on the left for the 2017 wet season in the Bahr El Ghazal basin in South Sudan generated with PIHM, and on the right for the Baro basin in Ethiopia in 2017 generated with TopoFlow. These models generate detailed time series of water flow along the river channel. The figure on the left shows in pink the locations with high risk of flooding at that particular point in time, showing roads in yellow and cities in red. The figure on the right shows in red two separate segments of the river that have higher water volumes at that time.

5.3.2 Southcentral United States. Southcentral regions of the United States, such as the state of Texas, are faced with unprecedented risks due to expected intensification of weather patterns and the projected doubling of population in the next 30 years, with concomitant urban, agricultural, and industrial growth posing increasing demands on water and energy resources. Vulnerability to drought is exacerbated by increased pumping of groundwater as population expands. This leads to regional depletion of major aquifers by hundreds of wells, reducing water reserves and resulting in impacts to ecosystems such as lower spring flow rates, limitations to land use due to subsidence, and sinking of land areas. In addition, extreme events such as destructive floods require accurate modeling of potential overflow of rivers, particularly in urban areas. Models of the hydrological and groundwater systems help answer important questions and support planning for future scenarios. What levels of pumping in wells are sufficient to conserve water for expected drought periods? What areas will be safe from flooding so that infrastructure and critical services can be properly positioned?

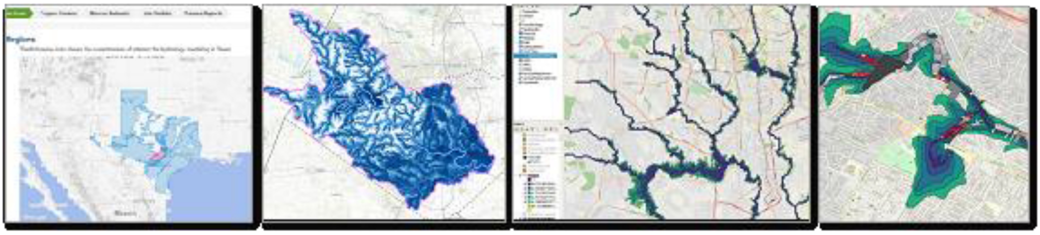


Fig. 18. Models of flood vulnerability in central Texas.

In collaboration with the Planet Texas 2050 research initiative at The University of Texas at Austin, we have been including in MINT hydrologic models for groundwater across the state as well as surface water for flooding and other related models.

Figure 18 shows model results of flooding risk for a small urban area near Austin, generated using the HAND model. The model is simple and can be run quickly for large areas. The model does an assessment of flood risk down to the building level.

6 USER EVALUATIONS

The MINT framework is continuously improving based on user feedback, as we incorporate additional features and improve its usability. We recognize the importance of formal user evaluations, even though they require significant effort for a complex system like MINT. Evaluating MINT with modelers and decision makers would be very difficult since they are scarcely available and designing and arranging the evaluations would require significant resources. In addition, that level of effort would be worth doing only with a more polished and improved user interface that we are more confident will reveal useful insights. Since we are still early in the development of MINT, we have not done formal user evaluations in our work to date. Instead, we have focused on formative evaluations with a few users intended to inform our research, reveal usability issues, and prioritize planned extensions and future work.

We carried out an initial formative evaluation in September of 2019 when MINT was in the early stages of development. This evaluation focused on model search and model execution capabilities of the MINT UI. The subjects were graduate students with basic expertise in economic modeling but no previous exposure to MINT. Each participant was provided a user guide of MINT plus instructions to complete a series of modeling tasks using a simplified economic model for agriculture production in South Sudan. All subjects were able to complete their modeling tasks within 40 to 70 minutes. When asked to give their impression of MINT, all participants gave positive feedback about the modeling capabilities of the system, specifically commenting on the ability to easily learn more about models and the ability to execute models with different data. Users also reported some difficulties when using the UI, in particular, having to make assumptions about how the models worked in order to successfully finish their tasks. We used this feedback to extend model metadata, to improve the documentation about models, and to allow users to access information based on the region of interest.

A second formative evaluation was carried out in December 2020 with a more advanced version of MINT. The nine participants ranged from graduate students with some modeling background to modelers who had an interest in AI technologies. Each participant attended a 20-minute tutorial overview of MINT and was given an hour to complete three tasks: (1) find models according to given keywords or indices and describe their inputs and variables; (2) find existing results and datasets in the data and provenance catalogs; and (3) execute two models for assessing drought and crop production, respectively, and answer questions about the results. Once the tasks were completed, participants were asked to fill in a survey about the usability of MINT. All usability questions followed a five-point Likert scale, ranging from "very easy" to "very difficult" (the neutral answer was "moderate").

Eight participants were able to complete all three tasks correctly. All participants managed to find their target datasets and models in MINT, execute models, and answer brief questions about the results. A ninth participant completed most of the tasks successfully but was not able to complete the third, not being able to locate the execution results of the crop and planting date specified in the task. We believe that this is due to the current design of the user interface that separates the search for raw data (done in the data catalog) and the search for data products of models (done in the provenance catalog), and that a unified, consistent view on data would improve usability. When asked for feedback, all participants found it easy or very easy to find a model to fit their purposes in MINT, and the majority found it easy or moderate to understand the purpose of a model. Most participants (56%) found it moderate to compare the differences between existing models for a given task. All participants except one found it easy or very easy to find datasets and existing results, and most participants found that it was easy or very easy to set up a new modeling task. As for the points for improvement, respondents mentioned that the user interface is sometimes "a little overcrowded with text", and suggested improvements for the specification of problem statements. They also asked for better support for visualizations of model results.

Overall, 77% of the participants found it easy or very easy to use MINT once familiar with the platform (23% considered it moderate), mentioning that they were surprised by the suggestions made by the system at least once. Several comments mentioned that the user interface was clear and the information well organized (e.g., "This is a fantastic tool and will be very helpful for scientists to run models", "The interface is very intuitive").

We continue to improve the MINT system and its user interface. Major areas currently being redesigned are the specification of modeling tasks and threads, integrated access of raw data and model products, and the development of visualizations that highlight spatiotemporal patterns in the data.

7 MINT SYSTEM ARCHITECTURE AND SOFTWARE

Figure 19 shows an overview of the MINT architecture components, illustrating the services and catalogs that implement the capabilities described in previous sections. The MINT User Interface and Scenario Exploration component, described in Section 4.4 and illustrated in Section 5.2, serves as an entry point for users and orchestrates the invocation of all other MINT components. This user interface allows users to define their own tasks and problems to investigate, issue ensembles of model runs for execution, keep track of the problems and tasks defined by others; and explore datasets, provenance of existing runs and models by querying the data, provenance, and model services, respectively.

A snapshot of the MINT software and documentation can be found under open-source licenses in [53], including data services, model services, execution services, and the user interface.

Our representation for models and associated metadata is available as the Software Description Ontology for Models [75]. All of the model, model version, model configuration, and model setup metadata are described in a model catalog that uses semantic web standards [22], with links to external resources (e.g., GitHub, DockerHub) to appropriately version and store code and execution environments. The SVO ontology is available at [84] http://www.geoscienceontology.org/svo/1.0.0/.

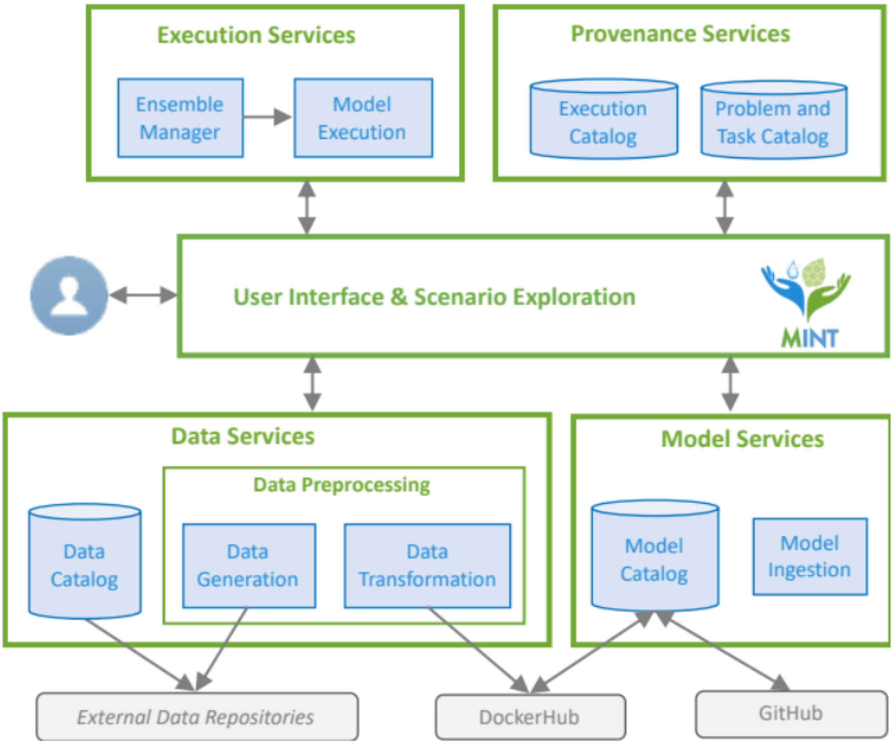


Fig. 19. An overview of the MINT architecture, showing the interdependencies between its modules.

8 DISCUSSION: INTEGRATED MODELING AND DECISION-MAKING

Table 5 revisits the modeling challenges in Table 1 and points out the relevant capabilities and benefits of MINT. Modeling in the context of decision-making involves diverse users and tasks, particularly when it concerns complex systems that require the integration of models from different disciplines. This section discusses the particular aspects of modeling that are addressed by our work to date.

A literature review about the use of models for decision-making is presented in [19], with a focus on groundwater models but offering general views on the modeling context. It describes the diversity of perspectives and values brought in by different stakeholders as considerations evolve from basic concerns with safe yield, to longer-term sustainable yield, to more comprehensive sustainable water management. It is important also to acknowledge the plurality of modeling expertise and types of knowledge that can be brought to bear in any given context [44]. Because the underlying systems are interconnected and interdependent, and each may be studied by a different discipline, integrated modeling can be approached as integrating that diverse expertise. Stakeholders are often seen simply as providers of data, but participatory modeling emphasizes the use models to empower stakeholders [56].

One way to set the context for our work is to examine the different kinds of users that would be involved in different aspects of those modeling and decision-making phases and stages. In that regard, we distinguish three different types of users based on required skill sets that can use MINT for different purposes:

Table 5. Benefits of AI Capabilities in MINT

Problem Addressed	Relevant AI Capabilities in MINT	Benefit
Delays in decision-making	• Metadata that enables data discovery • Generating novel data for modeling • Automated data transformations • Metadata that enables data and model discovery • Automated checking of model requirements and constraints • Step-by-step guidance for modeling tasks	Timely analysis
Limited scenario exploration	• Problem framing based on decision space • Models extended to expose potential interventions • Interactive dashboards to explore scenarios and interventions as well as their outcomes	Intervention-centered modeling
Restricted domain modeling	• Models expose only parameters that are relevant for decision-making • Models are preconfigured-precalibrated for easy use by others for scenario exploration	Accessible models
Static analysis reports	• Interactive dashboards to explore scenarios • Stylized narratives of modeling choices and scenarios • Provenance records with metadata of model runs	Interactive model products

- *Modeler*: This user generates model outputs and has basic expertise in the modeling domain. A modeler can use MINT to browse through preprepared models and preprepared datasets, select appropriate ones for their problems, and set up and run models. A modeler can also use MINT to explore interventions in models. This user can also employ MINT to publish model outputs to be used by other modelers (with expertise in other disciplines) or by analysts.
- *Analyst*: This user defines problems/scenarios for the modeler and creates reports for decision makers based on model products. The analyst may have some limited modeling expertise in the domain, enough to be able to run models already selected by the modelers. An analyst can use MINT to generate model outputs, determine problem areas, explore interventions, and create reports for decision makers. An analyst can use MINT to do large amounts of model runs in order to explore scenarios, which can then be used in other tools to estimate and characterize uncertainty.

- *Decision maker*: This user makes decisions and understands causality for interventions based on reports from analysts. The decision maker can use MINT simply to browse reports from analysts to understand situations and interventions, drilling down if needed to understand how model outputs were generated.

We also consider two additional types of users that populate MINT with models, data, and related software:

- *Expert modeler*: This user has deep expertise in a particular domain (e.g., agriculture modeling), along with a detailed understanding of modeling variables and processes as well as model software. This user would add new models with appropriate model configurations and setups for the problems and decisions of interest, and to run models with example/test data. The expert modeler would also implement custom data preparation, model calibration, data post-processing, and visualization codes that enable easy use of each model. This user would also do a sensitivity analysis to determine what model inputs result in the most uncertainty for the model outputs.
- *Data specialist*: This user is proficient in data formats and data systems, and can characterize, catalog, and curate data sources useful for modeling in the context of interest. This user would incorporate new data sources and datasets with necessary metadata. The data specialist would also populate the system with specialized data products, for example, extracted from remote sensing data, rescaling products, and highly curated data. This user would also implement custom data preparation and data transformation procedures for commonly used formats and models.

These tasks need to be accomplished before modelers and analysists can use MINT, and they are currently supported through APIs and services to add models and data. We note that an expert modeler is typically proficient in a particular domain; thus, in the case of complex systems, several expert modelers may be involved. Similarly, a range of skills may be required for handling data; thus, different data specialists may be involved. We are extending MINT to support these kinds of users.

Figure 20 illustrates the context in which MINT operates. Our focus has been on empowering modelers who are not necessarily expert in a domain to find appropriate models that have already been configured for a specific context (e.g., for a region) and find necessary data to run them. This kind of user is shown in the middle of the figure. In order for their work to be possible, expert modelers and data scientists would have to populate the system ahead of time by preparing the models for the regions of interest and incorporating relevant datasets into the framework. They also scope the context for modeling—in our case, defining regions for agriculture models and well-outlined river basins. Once modelers have created models, analysts can do uncertainty analysis and generate reports with appropriate explanations and supporting materials extracted from the provenance records so that decision makers can drill down and understand scenarios and interventions.

Figure 21 takes an even broader context, showing an idealized view of the stages involved in modeling in the context of decision-making, inspired by the cycle of participatory modeling proposed in [89] and the model-building steps articulated in [35]. Before the modeling stage that MINT addresses, there is an initial phase in which an expert modeler would confer with an analyst and decision maker to frame the modeling problem based on key issues facing the complex system under study. Then, the expert modeler would focus on key variables of interest in the system under consideration as well as reference behaviors and desirable outcomes. Once this is accomplished, the modeler moves to a second phase to explore potential scenarios. In a modeling stage, models

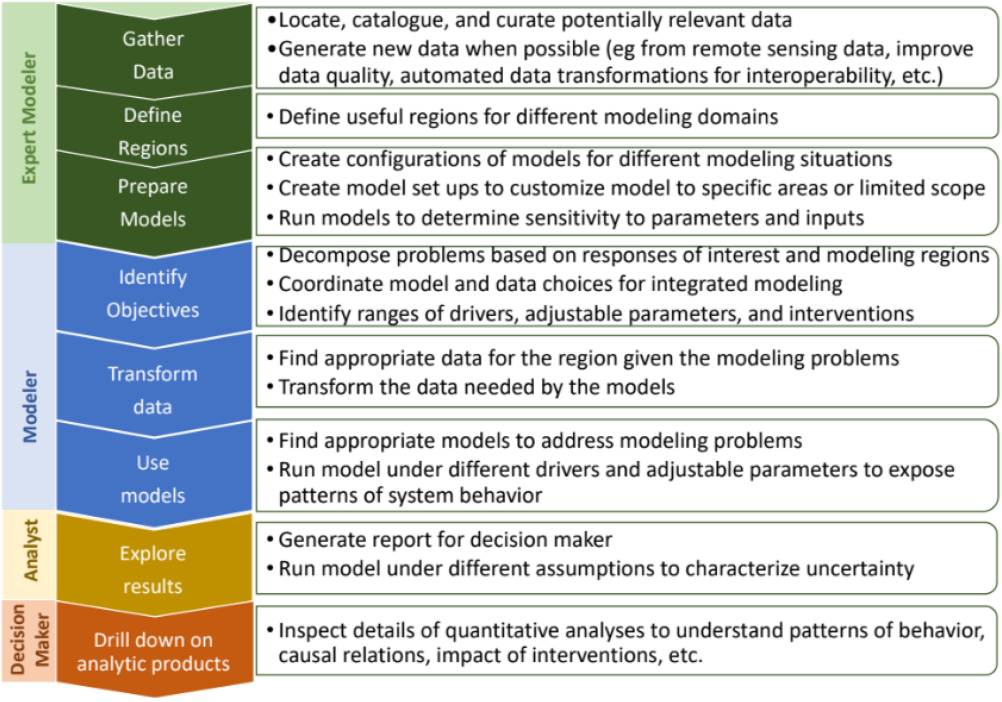

Expert Modeler
Gather Data
•Locate, catalogue, and curate potentially relevant data
•Generate new data when possible (eg from remote sensing data, improve data quality, automated data transformations for interoperability, etc.)
Define Regions
• Define useful regions for different modeling domains
Prepare Models
• Create configurations of models for different modeling situations
• Create model set ups to customize model to specific areas or limited scope
• Run models to determine sensitivity to parameters and inputs
Modeler
Identify Objectives
• Decompose problems based on responses of interest and modeling regions
• Coordinate model and data choices for integrated modeling
• Identify ranges of drivers, adjustable parameters, and interventions
Transform data
• Find appropriate data for the region given the modeling problems
• Transform the data needed by the models
Use models
• Find appropriate models to address modeling problems
• Run model under different drivers and adjustable parameters to expose patterns of system behavior
Analyst
Explore results
• Generate report for decision maker
• Run model under different assumptions to characterize uncertainty
Decision Maker
Drill down on analytic products
• Inspect details of quantitative analyses to understand patterns of behavior, causal relations, impact of interventions, etc.

Fig. 20. Major stages in integrated modeling.

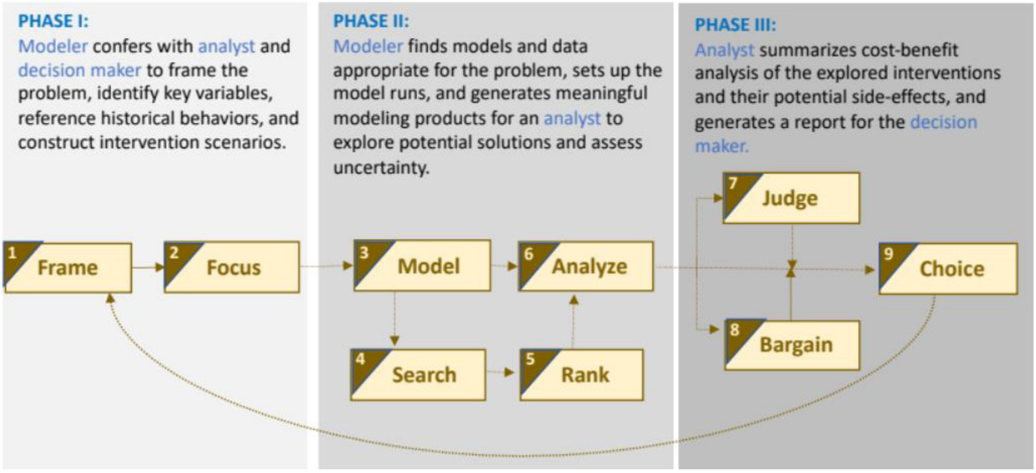

PHASE I:
Modeler confers with analyst and decision maker to frame the problem, identify key variables, reference historical behaviors, and construct intervention scenarios.
PHASE II:
Modeler finds models and data appropriate for the problem, sets up the model runs, and generates meaningful modeling products for an analyst to explore potential solutions and assess uncertainty.
PHASE III:
Analyst summarizes cost-benefit analysis of the explored interventions and their potential side-effects, and generates a report for the decision maker.
1 Frame
2 Focus
3 Model
4 Search
5 Rank
6 Analyze
7 Judge
8 Bargain
9 Choice

Fig. 21. Modeling in the context of decision-making.

are created and a baseline case is generated. The next stage involves search to explore the solution space and alternative scenarios. A rank stage is used to filter and rank solutions to select those of interest. The analyst then goes through an analysis stage to create a summary of the solutions explored and assess uncertainty. The third and final phase involves the generation of a report for decision makers and a judgment stage to adjudicate possible interventions, a bargain stage to consider trade-offs between them, and a choice stage to make recommendations and converge on decisions. These steps are iterated to refine, reframe, and refocus modeling goals. Our work to date focuses on the modeling stage of the second phase, but MINT can provide support throughout this process.

We have not discussed the economic cost of interventions, which usually offer trade-offs that are crucial for decision-making. Optimization algorithms are important for understanding these trade-offs and finding choice points in the decision space.

There are many opportunities to use AI to make these modeling processes more efficient by optimizing the role of the modeler and accelerating learning by the analyst. There are many opportunities to help an analyst work more efficiently. Provided that the computing and data storage capacities allow running simulations unattended, there are multiple options to run and rank simulations to explore a large array of model input and intervention combinations. An AI system can learn which of those combinations provide solutions within acceptable boundaries. For such a task, the modeling system needs to be able to run unsupervised, still a tall order for many expert domain models. Indicator variables in the outputs as well as indices need to be carefully selected and weighted to allow an AI system to grade scenarios so that it can learn in a way that mimics modelers and analysts. The AI system can be nested—for example, testing first the agricultural space to rule out conditions that are not worth exploring further with other models. Finding the right combination of automated and human effort requires further work to exploit synergies, work that can only be done systematically with robust systems such as those describe here.

9 CONCLUSIONS

This article describes AI techniques to assist modelers to create models of complex systems efficiently and in a form that will be relevant to decision makers. Our work makes several innovative contributions. First, goal-oriented modeling is used to frame modeling questions and formulate potential interventions and decision variables. Second, problem solving is used to select models and datasets relevant to the goals. This requires models to be encapsulated into configurations and settings that expose only relevant parameters and variables as drivers and responses, and relates them to interventions and decisions. Third, data are represented in terms of metadata, formats, structure, and contents so that they can be found and automatically transformed. Fourth, novel machine learning techniques are used to extract data from remote sensing sources that can be used when historical observations are not available to calibrate models. Fifth, an intelligent user interface guides users through structured stages of modeling, allows interactive exploration of scenarios, and generates provenance for model products to support explanation and reproducibility of the resulting reports presented to decision makers.

These innovations are implemented in the MINT framework, which includes real models and datasets for two different regions to analyze the interactions between natural and human systems—in particular, the relationships between climate, water availability, agriculture production, and markets. Users with general modeling background are guided to use sophisticated models in an accessible manner.

There are many directions for future work. We plan to extend MINT to assist expert modelers to encapsulate their models so that they can be easily used by other modelers. We plan to improve MINT to allow data scientists to extend the library of data transformations and data formats that can be handled, many of them of spatiotemporal nature. We are also starting to apply our framework to modeling in other regions (Asia) and domains (fire). Finally, while our formative evaluations have shown how users can use MINT successfully to find models and existing results and set up modeling tasks and problems, we also plan to extend our assessment by evaluating the use of MINT by expert modelers to apply models created by others and by analysts to run models and create reports with useful scenarios and uncertainty records that enable them to explore interventions and trade-offs.

MINT provides assistance for core modeling tasks surrounding the execution of models, such as identifying modeling objectives, transforming data, and running models. MINT could be extended to provide assistance in upstream tasks involving framing the modeling problem and preparing models for a region. MINT could also be extended to assist with subsequent tasks such as uncertainty analysis and report preparation.

A major area for future work is to leverage our approach to streamline model integration. In many cases, models are tightly integrated through model coupling in which state variables are exchanged continuously across models during simulation. Our approach would be best suited for other types of model integration that are more sequential in nature, in which the results of a model would be used by another model. Given the rich semantic data and model representations that we have developed, it should be possible to automate or assist with data transformations needed to convert a model's results into the formats needed by another model. An important aspect in model integration is ensuring that the models are used consistently, for example, in terms of their assumptions, the treatment of processes, and the use of the same or at least compatible data sources. Future research is needed to capture such forms of model dependencies as constraints, and to develop constraint reasoning techniques to assist users to ensure that the integrated models can produce valid results.

The need for modeling complex systems is crucial for the environmental sciences. This need is ubiquitous in many sciences, from physics to biology to medicine. By providing assistance and automation and by ensuring proper use of models, AI has immense potential to make modeling more efficient by orders of magnitude. This will accelerate the progress of science and our understanding of the world.

ACKNOWLEDGMENTS

We would like to thank our collaborators over the years for many useful insights that inspired and shaped this work.

REFERENCES

[1] M. Abulaish, and L. Dey. 2007. Biological relation extraction and query answering from MEDLINE abstracts using ontology-based text mining. *Data & Knowledge Engineering* 61, 2 (2007), 228–262.

[2] E. Afgan, D. Baker, B. Batut, M. Van Den Beek, D. Bouvier, M. Čech, J. Chilton, D. Clements, N. Coraor, B. A. Grüning and A. Guerler. 2018. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. *Nucleic Acids Research* 46, W1 (2018), W537–W544. DOI: 10.1093/nar/gky379

[3] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*. Springer, 722–735.

[4] Basic Model Interface. Available from https://bmi-spec.readthedocs.io/en/latest/.

[5] D. M. Blei, A. Y. Ng, and M. I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research* 3, 993–1022.

[6] D. Brickley, M. Burgess, and N. Noy. 2019. Google Dataset Search: Building a search engine for datasets in an open Web ecosystem. In *The World Wide Web Conference (WWW'19)*. Association for Computing Machinery, New York, NY, USA, 1365–1375.

[7] P. L. Buttigieg, N. Morrison, B. Smith, C. J. Mungall, and S. E. Lewis. 2013. The environment ontology: contextualising biological and biomedical entities. *Journal of Biomedical Semantics* 4, 1 (2013), 43. DOI: 10.1186/2041-1480-4-43

[8] M. J. Cafarella, A. Halevy, D. Z. Wang, E. Wu, and Y. Zhang. 2008. Webtables: exploring the power of tables on the web. *Proceedings of the VLDB Endowment* 1, 1 (2008), 538–549.

[9] L. Carvalho, D. Garijo, C. B. Medeiros, and Y. Gil. 2018. Semantic Software Metadata for Workflow Exploration and Evolution. *Proceedings of the Fourteenth IEEE International Conference on eScience*, Amsterdam, The Netherlands.

[10] S. Chalk, R. Hodgson, and S. Ray. 2017. Qudt toolkit: Development of framework to allow management of digital scientific units. In *Abstracts of Papers of the American Chemical Society*, 253.

[11] The Climate and Forecasting (CF) Conventions and Metadata. 2020. Available from https://cfconventions.org/.

[12] Cycles. 2020. http://plantscience.psu.edu/research/labs/kemanian/models-and-tools/cycles.

[13] DataCube. 2020. https://www.w3.org/TR/vocab-data-cube/.

[14] O. David, J. Ascough II, W. Lloyd, T. Green, K. Rojas, G. Leavesley, and L. Ahuja. 2013. A software engineering perspective on environmental modeling framework design: The Object Modeling System. *Environmental Modelling & Software* 39 (2013), 201–213.

[15] The Data Catalog Vocabulary (DCAT). 2020. https://www.w3.org/TR/vocab-dcat/.

[16] A. Dimou, M. Vander Sande, P. Colpaert, R. Verborgh, E. Mannens, and R. Van de Walle. 2014. RML: A Generic Language for Integrated RDF Mappings of Heterogeneous Data. *Proceedings of the 7th Workshop on Linked Data on the Web.* 1184.

[17] D. Hub. 2020. https://hub.docker.com/.

[18] H. Dong, S. Liu, S. Han, Z. Fu, and D. Zhang. 2019. Tablesense: Spreadsheet table detection with convolutional neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence.* Vol. 33.

[19] A. S. Elshall, A. D. Arik, A. El-Kadi, S. Pierce, M. Ye, K. Burnett, C. Wada, L. Bremer, and G. Chun. 2020. Groundwater sustainability: A review of the interactions between science and policy. *Environmental Research Letters.* https://doi.org/10.1088/1748-9326/ab8e8c.

[20] B. T. Essawy, J. L. Goodall, H. Xu, and Y. Gil. 2017. Evaluation of the OntoSoft Ontology for Describing Legacy Hydrologic Modeling Software. *Environmental Modelling & Software* 92.

[21] D. Garijo, D. Khider, Y. Gil, L. Carvalho, B. Essawy, S. Pierce, D. H. Lewis, V. Ratnakar, S. D. Peckham, C. Duffy, and J. Goodall. 2018. A semantic model registry to support comparison and reuse. In *Proceedings of the Ninth International Congress on Environmental Modeling and Software*, Ft Collins, CO.

[22] D. Garijo, M. Osorio, D. Khider, V. Ratnakar, and Y. Gil. 2019. OKG-Soft: An open knowledge graph with machine readable scientific software metadata. In *Proceedings of the 15th IEEE Conference on eScience.*

[23] W. Gatterbauer, P. Bohunsky, M. Herzog, B. Krüpl, and B. Pollak. 2007. Towards domain-independent information extraction from web tables. In *Proceedings of the 16th international conference on World Wide Web.* ACM, 71–80.

[24] Y. Gil, P. A. Gonzalez-Calero, J. Kim, J. Moody, and V. Ratnakar. 2011. A Semantic Framework for Automatic Generation of Computational Workflows Using Distributed Data and Component Catalogs. *Journal of Experimental and Theoretical Artificial Intelligence* 23, 4 (2011).

[25] Y. Gil, D. Garijo, S. Mishra, and V. Ratnakar. 2016. OntoSoft: A Distributed Semantic Registry for Scientific Software. In *Proceedings of the Twelfth IEEE Conference on eScience.* Baltimore, MD.

[26] GitHub. 2020. Available from https://github.com/.

[27] GLDAS. 2020. Available from https://ldas.gsfc.nasa.gov/gldas.

[28] A. Goel, C. Knoblock, and K. Lerman. ICAI. Exploiting structure within data for accurate labeling using conditional random fields. In *Proceedings on the International Conference on Artificial Intelligence (ICAI).*

[29] R. V. Guha, D. Brickley, and S. Macbeth. 2016. Schema.org: evolution of structured data on the web. *Communications of the ACM* 59, 2 (2016), 44–51.

[30] Z. GuoDong, S. Jian, Z. Jie, and Z. Min. 2005. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics.* Association for Computational Linguistics, 427–434.

[31] C. Hill, C. DeLuca, M. Suarez, and A. R. L. I. N. D. O Da Silva. 2004. The architecture of the earth system modeling framework. *Computing in Science & Engineering* 6, 1 (2004), 18–28.

[32] Humanitarian Data Exchange. 2020. Available from https://data.humdata.org/.

[33] J. L. Hellerstein, S. Gu, K. Choi, and H. M. Sauro. 2019. Recent advances in biomedical simulations: a manifesto for model engineering. F1000Research 8 (2019).

[34] R. Hoehndorf, M. Dumontier, J. H. Gennari, S. Wimalaratne, B. De Bono, D. L. Cook, and G. V. Gkoutos. 2011. Integrating systems biology models and biomedical ontologies. *BMC Systems Biology* 5, 1 (2011), 124.

[35] A. J. Jakeman, R. A. Letcher, and J. P. Norton. 2006. Ten iterative steps in development and evaluation of environmental models. *Environmental Modelling & Software* 21, 5 (2006). https://doi.org/10.1016/j.envsoft.2006.01.004

[36] S. Kandel, A. Paepcke, J. Hellerstein, and J. Heer. 2011. Wrangler: Interactive visual specification of data transformation scripts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* 3363–3372.

[37] A. Karpatne, A. Khandelwal, X. Chen, V. Mithal, J. Faghmous, and V. Kumar. 2016. Global monitoring of inland water dynamics: State-of-the-art, challenges, and opportunities. In *Computational Sustainability. Studies in Computational Intelligence*, J. Lassig, K. Kersting, and K. Morik (Eds.). Vol. 645, Springer, Cham. https://doi.org/10.1007/978-3-319-31858-5_7.

[38] A. R. Kemanian and C. O. Stöckle. 2010. C-Farm: A simple model to evaluate the carbon balance of soil profiles. *European Journal of Agronomy* 32, 1 (2010), 22–29.

[39] A. Khandelwal, A. Karpatne, M. E. Marlier, J. Kim, D. P. Lettenmaier, and V. Kumar. 2017. An approach for global monitoring of surface water extent variations in reservoirs using MODIS data. *Remote Sensing of Environment* 202 (2017), 113–128.

[40] A. Khandelwal. 2019. ORBIT (Ordering Based Information Transfer): A Physics Guided Machine Learning Framework to Monitor the Dynamics of Water Bodies at a Global Scale.

[41] G. King. 2007. An Introduction to the Dataverse Network as an Infrastructure for Data Sharing. *Sociological Methods & Research* 36, 2 (2007), 173–199.

[42] E. Koci, M. Thiele, O. Romero, and W. Lehner. 2016. Cell classification for layout recognition in spreadsheets. In *International Joint Conference on Knowledge Discovery, Knowledge Engineering, and Knowledge Management.* 78–100.

[43] S. Krishnan, J. Wang, E. Wu, M. J. Franklin, and K. Goldberg. 2016. Activeclean: interactive data cleaning for statistical modeling. *Proceedings of the VLDB Endowment* 9, 12 (2016).

[44] T. Krueger, T. Page, K. Hubacek, L. Smith, and K. Hiscock. 2012. The role of expert opinion in environmental modelling. *Environmental Modelling & Software* 36. https://doi.org/10.1016/j.envsoft.2012.01.011

[45] A. Langegger and W. Wöß. 2009. XLWrap — Querying and Integrating Arbitrary Spreadsheets with SPARQL. In *ISWC'09 Proceedings of the 8th International Semantic Web Conference*. 359–374.

[46] Q. V. Le. 2013. Building high-level features using large scale unsupervised learning. 2013. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. 8595–8598.

[47] M. Lefrançois, A. Zimmermann, and N. Bakerally. 2017. A SPARQL extension for generating RDF from heterogeneous formats. In *European Semantic Web Conference*. 10249. 35–50.

[48] H. Liu and P. Singh. 2004. ConceptNet—a practical commonsense reasoning toolkit. *BT Technology Journal* 22, 4 (2004), 211–226.

[49] M. McLennan, and R. Kennell. 2010. HUBzero: A Platform for Dissemination and Collaboration in Computational Science and Engineering. *Computing in Science & Engineering* 12, 2 (2010), 48–53.

[50] F. Michel, L. Djimenou, C. Faron Zucker, and J. Montagnat. 2015. Translation of relational and non-relational databases into RDF with xR2RML. In *11th International Conference on Web Information Systems and Technologies (WEBIST'15)*. 443–454.

[51] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*. 3111–3119.

[52] G. A. Miller. 1995. WordNet: a lexical database for English. *Communications of the ACM* 38, 11 (1995), 39–41.

[53] MINT software. Available from https://mintproject.github.io/mint/.

[54] Model Coupling Toolkit. Available from https://www.mcs.anl.gov/research/projects/mct/.

[55] MODFLOW. 2020. US Geological Survey https://water.usgs.gov/ogw/modflow/.

[56] D. M. Oliver, R. D. Fish, M. Winter, C. J. Hodgson, A. L. Heathwaite, and D. R. Chadwick. 2012. Valuing local knowledge as a source of expert data: Farmer engagement and the design of decision support systems. *Environmental Modelling & Software* 36. https://doi.org/10.1016/j.envsoft.2011.09.013.

[57] OpenRefine. 2020. Available from https://openrefine.org/.

[58] M. Palmblad, A. L. Lamprecht, J. Ison, and V. Schwämmle. 2019. Automated workflow composition in mass spectrometry-based proteomics. *Bioinformatics* 35, 4 (2019), 656–664. https://doi.org/10.1093/bioinformatics/bty646.

[59] J. M. Perkel. 2017. How bioinformatics tools are bringing genetic analysis to the masses. *Nature* 543, 7643 (2017), 137–138. DOI: 10.1038/543137a

[60] E. Pebesma, T. Mailund, and J. Hiebert. 2016. Measurement units in R. *The R Journal* 8, 2.

[61] S. D. Peckham and M. Stoica. 2018. Principle-based, Semi-automatic Ontology Generation to Support Cross-Domain Interoperability of Data Sets and Models. In *Proceedings of the 9th International Congress on Environmental Modelling and Software*.

[62] S. D. Peckham, E. W. H. Hutton, and B. Norris. 2013. A component-based approach to integrated modeling in the geosciences: The design of CSDMS. *Computers and Geosciences* 53, 3–12.

[63] S. D. Peckham, M. Stoica, E. E. Jafarov, A. Endalamaw, and W. R. Bolton. 2017. Reproducible, component-based modeling with TopoFlow, a spatial hydrologic modeling toolkit. *Earth and Space Science* 4, 6 (2017).

[64] J. Pennington, R. Socher, and C. Manning. 2014. Glove: Global Vectors for word representation. In *Proceedings of the 2014 Conference On Empirical Methods in Natural Language Processing (EMNLP)*. 1532–1545.

[65] M. Pham, S. Alse, C. A. Knoblock, and P. Szekely. 2016. Semantic labeling: a domain-independent approach. *International Semantic Web Conference*. Springer, Cham.

[66] J. Pujara, A. Rajendran, M. Ghasemi-Gol, and P. Szekely. 2019. A Common Framework for Developing Table Understanding Models. *International Semantic Web Conference - Posters*.

[67] Y. Qu and C. J Duffy. 2007. A semidiscrete finite volume formulation for multiprocess watershed simulation. *Water Resources Research* 43, W08419. https://doi.org/10.1029/2006wr005752

[68] S. K. Ramnandan, A. Mittal, and C. Knoblock. 2015. *Assigning Semantic Labels to Data Sources*. European Semantic Web Conference. Springer, Cham.

[69] R. G. Raskin and M. J. Pan. 2005. Knowledge representation in the semantic web for Earth and environmental terminology (SWEET). *Computers & Geosciences* 31, 9 (2005), 1119–1125. https://doi.org/10.1016/j.cageo.2004.12.004.

[70] RDF. 2020. Available from https://www.w3.org/RDF/.

[71] D. Ritze, O. Lehmberg, and C. Bizer. 2015. Matching HTML Tables to DBPedia. In *Proceedings of the 5th International Conference on Web Intelligence, Mining and Semantics*. ACM, 10.

[72] P. Rodriguez-Tomé. 1998. The BioCatalog. *Bioinformatics* 14, 5 (Jun 1998), 469–470, https://doi.org/10.1093/bioinformatics/14.5.469

[73] O. Ronneberger, P. Fischer, and T. Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer 234–241.

[74] S. Sarawagi. 2008. Information extraction. *Foundations and Trends in Databases* 1, 3 (2008), 261–377.

[75] The Software Description Ontology for Models. 2020. Available from https://w3id.org/okn/o/sdm/.

[76] L. Shamir, J. F. Wallin, and A. Allen. 2013. Practices in source code sharing in astrophysics. *Astron Comput.* 1 (2013), 54–58.

[77] B. Shbita, A. Rajendran, J. Pujara, and C. Knoblock. 2019. Parsing, Representing and Transforming Units of Measure. *Modeling the World's Systems Conference*. Washington DC.

[78] Y. Shi, K. J. Davis, C. J. Duffy, and X. Yu. 2013. Development of a coupled land surface hydrologic model and evaluation at a critical zone observatory. *Journal of Hydrometeorology* 14, 1401–1420, DOI: 10.1175/JHM-D-12-0145.1

[79] J. Slepicka, C. Yin, P. Szekely, and C. Knoblock. 2015. KR2RML: An Alternative Interpretation of R2RML for Heterogeneous Sources. In *Proceedings of the 6th International Workshop on Consuming Linked Data (COLD)*.

[80] C. O. Stöckle, A. R. Kemanian, R. L. Nelson, J. C. Adam, R. Sommer, and B. Carlson. 2014. CropSyst model evolution: From field to regional to global scales and from research to decision support systems. *Environmental Modelling & Software* 62 (2014), 361–369.

[81] M. Stoica and S. D. Peckham. 2018. An ontology blueprint for constructing qualitative and quantitative scientific variables. *International Semantic Web Conference (P&D/Industry/BlueSky)*.

[82] M. Stoica and S. D. Peckham. 2019. Incorporating new concepts into the scientific variables ontology. In *Workshop on Advanced Knowledge Technologies for Science in a FAIR World*.

[83] M. Stoica and S. D. Peckham. 2019. The scientific variables ontology: A blueprint for custom manual and automated creation and alignment of machine-interpretable qualitative and quantitative variable concepts. *Modeling the World's Systems Conference*.

[84] The Scientific Variables Ontology (SVO). 2020. Available from http://www.geoscienceontology.org/svo/1.0.0/.

[85] P. Szekely, D. Garijo, D. Bhatia, J. Wu, Y. Yao, and J. Pujara. 2019. T2WML: Table To Wikidata Mapping Language. In *Proceedings of the 10th International Conference on Knowledge Capture (K-CAP'19)*. Association for Computing Machinery, New York, NY, USA, 267–270.

[86] Trifacta. 2020. Available from https://www.trifacta.com/.

[87] M. J. Turk, B. D. Smith, J. S. Oishi, S. Skory, S. W. Skillman, T. Abel, and M. L. A. Norman. yt: A multi-code analysis toolkit for astrophysical simulation data. *The Astrophysical Journal Supplement Series* 192, 1.

[88] Unidata. 2012, *Integrated Data Viewer (IDV) version 3.1 [software]*. Boulder, CO: UCAR/Unidata. http://doi.org/10.5065/D6RN35XM

[89] A. Voinov, N. Kolagani, M. K. McCall, P. D. Glynn, M. E. Kragt, F. O. Ostermann, S. A. Pierce, and P. Ramu. 2016. Modelling with stakeholders – Next generation. *Environmental Modelling & Software* 77. https://doi.org/10.1016/j.envsoft.2015.11.016

[90] B. Vu, J. Pujara, and C. Knoblock. 2019. D-REPR: A Language for Describing and Mapping Diversely-Structured Data Sources to RDF. In *Proceedings of the 10th International Conference on Knowledge Capture (K-CAP'19)*. Association for Computing Machinery, New York, NY, USA, 189–196.

[91] Z. Wei, K. Jia, X. Jia, A. Khandelwal, and V. Kumar. 2020. Global river monitoring using semantic fusion networks. *Water* 12, 8 (2020), 2258.

[92] M. D. Wilkinson, B. Vandervalk, and L. McCarthy. 2011. The Semantic Automated Discovery and Integration (SADI) Web service Design-Pattern, API and Reference Implementation. *Journal of Biomedical Semantics* 2, 8 (2011). https://doi.org/10.1186/2041-1480-2-8

[93] R. Zanibbi, D. Blostein, and J. R. Cordy. 2004. A survey of table recognition. *IJDAR* 7, 1–16. https://doi.org/10.1007/s10032-004-0120-9

[94] L. Zentner, M. Zentner, V. Farnsworth, M. McLennan, K. Madhavan, and G. Klimeck. 2014. Nanohub.org: Experiences and challenges in software sustainability for a large scientific community. *Journal of Open Research Software* 2(1), e19. DOI:http://doi.org/10.5334/jors.bd

[95] X. Zheng, D. G. Tarboton, D. R. Maidment, Y. Y. Liu, and P. Passalacqua. 2018. River channel geometry and rating curve estimation using height above the nearest drainage. *Journal of the American Water Resources Association* 54 4 (2018), 785–806. https://doi.org/10.1111/1752-1688.12661.

Received May 2020; revised January 2021; accepted February 2021