A Semantic Model of Computation for Natural System Modelling

Ferdinando Villa

Ecoinformatics Collaboratory, University of Vermont, USA. E-Mail: **ferdinando.villa@uvm.edu**

Keywords: *Integrated Modelling Frameworks, Ontologies, Conceptual Modelling.*

EXTENDED ABSTRACT

This contribution is a reflection on the often mentioned idea of *declarative modelling*, linking it to the most recent advances in knowledge representation science and to the semantic web initiative. I will present a next-generation approach to modelling that has the potential of unifying representations of data and metadata, their use in the chain of processing in scientific workflows, and the definition of dynamic models. The key tool for this unification is the formal statement of the conceptualization that lies behind the choice of representation paradigm used (*ontology).* I will describe briefly the fundamental concepts, the dominant approaches and the most important uses of ontologies in natural system modelling, with reference to their increasing adoption in projects worldwide.

While laying out a taxonomy of different ways of conceptualizing a natural system for the purpose of simulation, I will highlight the implications of the choice of the abstraction level for modeling, data mining and data/model integration. The paper will discuss the potentials of extending the presented ontology-driven approach to the semantic annotation of datasets, the building of computational workflows for data analysis, the definition and computation of natural system models, and the specification of queries for knowledge discovery based on distributed databases.

I will present well-known environmental models as examples, with reference to existing and novel modelling systems and formalisms. My aim is to show how the choice of a purely declarative representation of a model, supported by appropriate domain ontologies, can not only allow a transparent integration of the model with independent, heterogeneous data and models, but can also turn a model representation into a powerful discovery tool that can be used as a constraint over a distributed knowledge base or semantic web in order to discover new knowledge in a fully automated way.

1. INTRODUCTION: ONTOLOGIES AND THEIR ROLE IN NATURAL SYSTEM MODELLING.

This paper links the often-mentioned notions of declarative modelling, metadata, scientific workflow, data integration and environmental databases in a synthesis inspired by the most recent advances in knowledge representation. I will start by introducing the key tool for this unification, the *ontology*. I will then proceed to describe how the formalization of a conceptual domain can inform the gathering of data, the conceptualization of models, and their simulation. I will describe a unifying perspective of natural system modelling with reference to the increasing adoption of a knowledge-based approach in projects worldwide. The synthesis described in this paper has wide-ranging implications for the practice of natural system modelling and for environmental research at large. I devote the last chapter to describing these prospective implications, trying to describe the novel scenarios and scientific approaches that are made possible by the adoption of a knowledge-based perspective.

Ontologies are formal descriptions of a conceptualization of a domain of interest. An ontology is a set of statements that define concepts and relationships between them. In an ontology, a **concept** (also referred to as a **class**) is the statement of a concept, usually including at least textual description and a short label. Concept names (IDs) and descriptions do not define meanings formally: concepts are always defined by their properties, not by their names. A **property** is the statement of an attribute associated with a concept. The statement always mandates the type of the property value and its *cardinality*, or the admitted number of possible values. For example, concept Person can have a *birth-date* property whose value must be a Date and whose cardinality is one. The value of a property can be a concept, an instance (see below) or a textual value. Special properties allow building the bones of the knowledge structure: notably the *is-a* (or *subclass-of*) property allows to build generalization-specialization hierarchies (e.g. Employee is-a Person). Properties are often treated using a class model in ontologies, and can be generalized or specialized just like concepts: e.g. "*depends-on-economically* is-a *depends-on*". **Instances** (or Objects) are the statement of an entity that must "exist" in some - real or virtual – world, and represents the "incarnation" of a stated concept. An ontology that defines a set of instances along with the concepts they incarnate is often called a **knowledge base**, although the term is not rigorous. Any database with a formally specified schema can be considered a set of instances of the correspondent ontology, where the schema acts as a proto-ontology that defines terms and basic relationships, and the database paradigm provides an interface and optimized methods for storage and retrieval of instances based on matching of properties with user-defined constraints. **Relationships** link an instance to the value of a specified property. E.g. the statement "JohnSmith birth-date 10-10-1972" can be considered the statement of a relationship. Relationships must adhere to the model specified by the concept and its properties.

A useful ontology usually contains more than just a list of terms and their definitions. Ontologies can be used simply as **controlled vocabularies,** a set of concepts with no properties, whose IDs define a set of usable terms. A step further is a **taxonomy,** where concepts are arranged in a generalization (is-a) hierarchy. A **schema** is an ontology where properties other than is-a are defined for classes. A generalization hierarchy may or may not be present**.** These **normative** uses, where ontologies provide guidance for conceptualization, encompass formalizations that are common in natural system information management and modelling, such as metadata standards and database schemata. The main rationale for the existence of ontologies, however, is to enable **reasoning** on natural systems, either by a human actor or by an automated program (**reasoner**). The main operations in reasoning are **subsumption** (inferring that concept A is more general that concept B**)** and **classification** (inferring that instance X is the incarnation of concept A). In the **frame-oriented** approach, concepts are arranged in a taxonomy, with arbitrary properties, constraints, objects, values. In the **description logics** framework, necessary and sufficient conditions for concepts and instances to belong to a class can be specified, allowing automated first-order logics reasoning. These uses of ontologies open new doors to natural system modelling that will be explored in the following. This use relies mostly on **restrictions**, the equivalent of first-order logic statements that define allowable values and patterns of values for instances to belong to concepts within an ontology. The description logics ontology framework which has been adopted by the Semantic Web activity is the Ontology Web Language OWL: (W3C 2004).

2. CONCEPTUALIZING DATA

Data can be considered "static models" of systems: they always conform to conceptual models, and depend on (usually implicit) assumptions and

world views just as much as dynamic models. Conventional wisdom divides "raw data" – actual numeric measurements – from "metadata", the information that allows a user to "make sense" out of the numbers, providing needed spatiotemporal, measurement, and other informative contexts for the numbers. In an ontology-informed framework, the starting point is the careful formalization of the concept that is represented in the data, including the definition of all its characteristics as properties. Each dataset or piece of data will be represented as one instance of that concept. One of its properties will be "numeric-value" and will link to the raw data, e.g. representing number using textual or binary values. The other properties constitute the "metadata", which are, in this case, connected to an explicit knowledge model (the ontology) whose arbitrary richness now allows reasoning and mediation with other data using other knowledge models whose compatibility can be assessed by reasoners. The ontology paradigm in its simplest incarnation offers a field-by-field substitution for the relational schemata conventionally used to represent data. Schema information contained in normative ontologies can be construed as an add-on to an existing database management system, to be implemented into a relational database engine or another (such as an XML database) in a modular fashion. In addition to that, ontologies offer a powerful synthetic way to specify both the data schema and the structure of the knowledge behind it. While a relational schema can be considered the "structural" component of the knowledge, ontologies allows specifying the *how* and the *why* at the same time. The database paradigm that includes a rich knowledge model such as that specified by description logics is often called *deductive database*.

Many metadata standards have been formalized as ontologies: e.g. the ISO 19115 standard for spatial information (ISO 2004). Yet, providing a set of ontologies to define the actual *meaning* of the metadata properties is a much more sophisticated activity, because the terms of a metadata dictionary lend themselves to different, often competing interpretations that ultimately depend on the application. All mediator systems that can integrate data and models need such a set of ontologies (Ludaescher, Gupta et al. 2001); the ones related to the process of measurement and the conceptualization of time and space are usually crucial. Efforts are underway in projects such as Kepler (Kepler 2004; SEEK 2004), IMA (Villa 2001; Villa In press) and SEAMLESS (SEAMLESS 2005) to develop a set of ontologies that provides a useful and consensual description of an application domain. Concepts are slowly consolidating toward a set of common terms and definition to define semantically-explicit modelling. Conceptualizations evolve with their applications, and this process is not expected to be ever complete, but efforts are underway to identify tools and protocols to make it manageable and trackable in the long term. The reminder of this paper adopts the terminologies and higher-level conceptualization laid out in the IMA and Kepler approaches.

3. SCIENTIFIC WORKFLOWS

A *scientific workflow* is a pathway between two or more processing steps, along which a flow of data is transformed until a desired result is reached. Workflows are usually assembled by users to connect sources of input data (such as a database query) to models, pre- and post-processing algorithms (such as statistical data reduction or calculation of indicators from model results) and visualization software. A *workflow environment* is thus a "black board" for users to assemble the flow of computation needed to address a particular problem. In functional terms, a dynamic model can often be seen as a workflow, because in the end all models process input information and produce a set of output results. The similarity, however, does not hold when models and workflows are considered in semantic terms: the *meaning* of a model is not to produce outputs, but to describe a natural process. In fact, a more correct generalization sees workflows as special cases of models, whose "paradigm" entails data transfer and transformation along the connections of an artificial system.

In this sense, a workflow is amenable to the same ontology-based description as any other model: concepts of "input", "output", "processing step" can be specialized as needed using ontologies that can describe all steps of any workflow environment. The most important use of ontologies in workflow environments, however, is another: to allow the system to enforce meaningful, correct connections between inputs and outputs, and – if necessary and possible – insert transformation steps in the workflow that guarantee a proper match. The operation of enforcing and supporting semantic consistency along data paths in workflows is usually called *semantic mediation* (Ludaescher, Gupta et al. 2001), and it is essential to guaranteeing correct results particularly when processing steps are heterogeneous and users are not domain experts. To allow semantic mediation, all inputs and outputs must come tagged with concepts from ontologies that are known to the workflow environment, and the latter needs to use a reasoner program that ensures the consistency of concepts along each connection made by the user. The operation of associating concepts from ontologies to input and output "ports" of workflow

components is usually called *semantic annotation*, and it is done by the same experts that have developed the models or processing steps. Conceptual compatibility of inputs and outputs is most often tested with a reasoning operation that assesses semantic differences based on matching of properties and not of names. The example below illustrates how this may be done.

A model X is "packaged" as a workflow component and all its inputs are semantically annotated by its developer according to a set of commonly understood ontologies. The semantic annotation operation requires that the conceptual details of each "port" that are relevant to the calculations are understood and appropriately defined. As an example, an input I representing temperature at surface may require that the temperature is expressed as monthly data over the simulated time span, and the model has only been calibrated for temperatures in the 19–30 C range so it should not accept data outside of these boundaries. Semantic annotation is a way to express such conditions (which normally are only expressed verbally in the model's documentation) in a formal and machine-readable way. In order to do so, an ontology is created to define model X, where each exposed "port" is expressed as a concept defined in terms of known concepts. Using restrictions and concepts from appropriate ontologies, the concept definition associated with input I may look similar to that shown in the text box.

```
I ::=
        is-a: Temperature,
        vertically-distributed-in:
        PlanetarySurface,
        has_unit: Fahrenheit,
        max-value:
        (Temperature,  has-value:  30.0,
        has-unit: Celsius)
        min-value:
        (Temperature,  has-value:  19.0,
        has-unit: Celsius)
        distributed-in:
        (TimeSpan,  step:  1,  has-unit:
        Month)
```

When a semantically annotated model is used in a workflow, inputs and outputs are connected by the user. For example, a temporal series of temperature data retrieved from a database may be connected to input I. Upon connection, a semantically aware workflow environment can ensure the appropriate match between the input and the output by feeding the respective semantic annotations to a reasoner and ask if they describe the same concept (a subsumption operation). A reasoner can make the necessary inferences to assess the equivalence of types that have different names, based on their properties. In some cases, the compatibility may not exist directly, but the reasoner can establish that a transformation can be inserted in the dataflow to make the input and output compatible. For example, the data source could be weekly data rather than monthly. A sophisticated workflow environment (such as Kepler) can understand that the data need to be aggregated into a monthly timeseries, and direct the workflow environment to create a transformation step to perform the aggregation and insert it between the data source and the model. If a transformation cannot be established due, e.g. to incompatible resolutions or excessive transformation error, the reasoner can output an appropriate failure message.

4. CONCEPTUALIZING MODELS

Dynamic models, like data, always conform to a conceptualization, and there is in fact no philosophical difference between specifying data or models when this is done using ontologies (Villa 2001; Villa In press). Any sort of model can be successfully specified as a set of instances of the appropriate ontologies. The main practical difference between data and models is the increased conceptual richness necessary to describe how things change in time and space. This requires at least notions of linkage between concepts with causative or dependency relationships that are normally not necessary when specifying data. It also requires developing ways to *interpret* this causality. The set of abstractions (concepts) that allows conceptualizing and expressing those cause-effect relationships and their results is the adopted **modelling paradigm**, of which examples abound (e.g. ordinary differential equations, stock-and-flow, or individual-based). A modelling paradigm, like any consistent conceptualization, can be captured into an ontology. Most existing modelling software systems (e.g. STELLA, SIMILE) conform to one implicit ontology, which defines their notion of entities familiar to the user such as state variables, flux variables etc. Advanced integrative systems (e.g. IMA) can load different ontologies, which, supplemented by the necessary software, enable them to manipulate models adopting heterogeneous modelling paradigms. Such systems are in the best position to enable integration of independently developed models adopting different paradigms into a higher-level, multiple-paradigm model.

The wording **declarative modelling** has been used to refer to a specification of models that is based on the attributes and semantics of the natural systems rather than the algorithm that calculates their results. Ontologies support declarative modelling by providing, at the same time, schemata for model declaration and meaning for these schemata. In-

stances of ontologies represent declaratively expressed models that refer to concepts laid out in the ontologies. Such declarations contain enough information to enable a software infrastructure to simulate the behavior of the systems represented over a user-defined temporal and spatial extent. Thanks to the rich meaning made possible by ontologies, a workflow environment can properly connect models to data, and feed quantities calculated by simulation to other models in the same environment.

Concept-driven modeling environments can be devised where all concepts used to model a natural system are explicitly defined by standard ontologies, and all technological details related to the calculation of the model are hidden from the user. This approach unifies and outgrows common notions such as "metadata" and "modeling paradigm" and opens perspectives of seamless data/model integration and intelligent, hypothesis-specific database querying. These advanced knowledge-based systems (e.g. IMA, Villa, 2001; Villa, in press) are typically not committed to a particular set of concepts except for a core ontology, carefully designed for generality, paradigm neutrality, and extensibility. Their rationale is the notion that an accurate description of nature's entities is enough information to allow a system to describe data, calculate and integrate models, as long as enough knowledge has been built into the system to allow their description. The uncoordinated extensibility of such environments allows domain experts to produce knowledge describing specific disciplinary contexts. Users can adopt the correspondent concepts to produce representations of natural systems that the system knows how to resolve into numeric states.

The philosophy of declaring a model can be summarized into laying out (1) reference concepts and properties to define the identity of each modelled entity, and (2) the properties that capture the causal relationships that are the key to distinguish a model from data. Causality in conventional models is usually expressed through equations, defined to calculate the value of *variables*. Equations, by naming the values of other variables, implicitly define causal relationships that are viewed as dependencies from a processing point of view. An ontology-based framework can make these dependency relationships explicit, and add semantics to them by means of specialization. So, for example, a generic depends-on relationship can be specialized into a *flows-into* relationship between a state variable and a flux variable (rate), in order to inform the underlying software architecture that the flux must be integrated over time. The notion of variable, so central to conventional approaches, can similarly be enriched and made dependent on the modelled entity. For example, in an individual-based paradigm, variables describe quantitative traits of modelled individuals, but maintain the link to the individual which is the main entity considered. No conflicts need exist between paradigms, whose conceptual boundaries often become blurred when a explicit knowledge-based approach is used, particularly if notions of scale are formally defined (Villa In press).

There are limits to the declaration of models in current ontology frameworks, particularly if reasoning capabilities must be preserved. These limitations stem from the fact that any dependency other than linear requires second-order logics statements to be fully formalized, and the handling of second-order logics is beyond the capabilities of existing reasoning systems. The dominant paradigm, Description Logics, can only operate on first-order statements. Even with these limitations, reasoning can be profitably used to enforce correct design and consistent definition of models. As an example, the biodiversity ontology used in the IMA to model a community states that coexistence of populations in a community (e.g. as captured in the *coexisting-population* relationship) also implies coexistence in space and time. Software implementations of the approach can be taught to automatically check that the specification is consistent with coexistence in both space and time before the model can be accepted or calculated. This prevents users from defining inconsistent models and helps retrieval of compatible data sources from databases when the model is applied. Reasoning of such kind can be used to assist proper design and application of a model by enabling a software system to enforce model design disciplines, transcending the mere engineering realm, and therefore facilitating the use of complex simulation models by non-scientists such as decision makers, and ensuring their correct application at the same time.

5. PERSPECTIVES

Uniform, paradigm-explicit, software-independent, and knowledge-rich representation of data and models opens both obvious and less obvious perspectives. Among the obvious ones, models become dependent on the ontologies that provide their meaning, not on the software that calculates them, so that can be executed by different infrastructures, translated, and integrated with data and other models. Current efforts using semantics to enable such integration are at the forefront of research in several fields: SEEK (SEEK 2004) in ecology and biodiversity, GEON (GEON 2005) in geology, GBIF (GBIF 2004) in taxonomy,

SEAMLESS (SEAMLESS 2005) in agriculture. In the following, I will explore two less-obvious perspectives opened by modelling nature at the conceptual level. The first case is the seamless extension of metadata semantics to declarative modelling, enabling integrated specification and storage of data and models. The second case, a direct consequence of the first, is the use of abstract model structures as powerful search tools to hunt for significant regularities in distributed data collections.

5.1. Uniformity of data and model representations

The common "hare and lynx" predator-prey system evokes a nonlinear differential equation system or a discrete "stock and flow" model in the mind of most ecologists. Figure 1 explores this model in a knowledge-based framework: in 1A, the "identity" of the system is captured in a particular instant of time, while in 1B the knowledge necessary to infer a dynamical model is added.

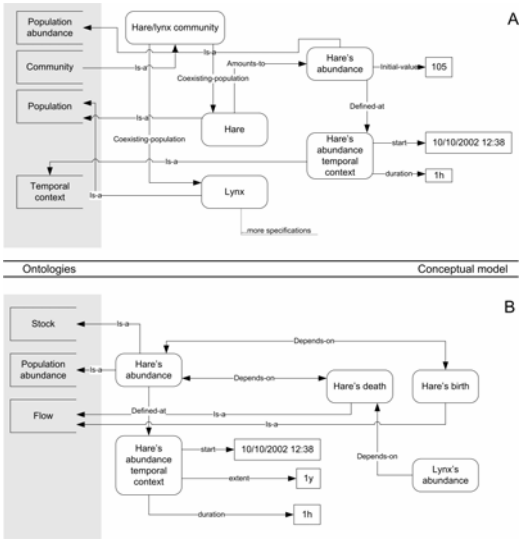

Figure 1. Portions of a possible conceptualization of a predator-prey community. In (A), the system is conceptualized as static. In (B), conceptual details from a modeling ontology are added that allow a system to infer a calculation workflow for the system.

The system shown uses concepts from biodiversity ontologies that describe populations and communities, coupled with a taxonomy ontology that allows defining species unambiguously by referring to and reinterpreting data from known repositories such as GBIF (GBIF 2004). The knowledge model in 1A is essentially a dataset: it states that at a certain time two populations coexist in a community, and have specified numerical abundances. Yet, it's much richer than a typical dataset with conventional metadata, because of the added knowledge expressed in the ontologies. System 1A is the equivalent of two state variables with initial conditions and no dynamic information. The specification of the population abundances can be extended to inform the system of how they change in time. For example, adopting a dynamic system paradigm based on "stocks" and "flows", we can state that the abundance of the lynx population is not only a numeric abundance, but also a *stock*, and refer to a suitable dynamic systems ontology to provide a formalization of the concept. Its time specification can say that while its value in the hour following measurement is the initial value of 1A, the extent now expresses its state over a one year temporal extent. Corresponding *flows* are added and linked to the existing model to express the factors that influence the change in the stocks. A system can analyze the declaration and decide that in order to know the numeric state of the system over the given extent, the stock and flow identities require difference equations to be defined, and the initial state must be extended in time by integrating the flows. The result should preserve the original identities, producing a system very similar to 1A, only with different abundance values per each time step. The system will automatically define a process in terms of *variables*, *inputs*, *outputs*, concepts that users won't need to manipulate unless they want to. The paradigm-specific information can be absorbed in the knowledge base of the framework, and users only need to concentrate on accurately describing the system using known, well-documented concepts. The dynamic specification, shown partially in 1B, is considerably more verbose, but still recognizable as an extended version of the previous "static model". It is noteworthy how metadata, units, and other information are the same: storage, query or inference can use the same infrastructure for either model.

Modeling at the conceptual level allows users to employ a language that's tailored to the knowledge domain of reference, adding the necessary dynamic information to the definition so obtained, and let the system figure out an appropriate computing workflow. The advantage is that, as each modeling paradigm can be described by a set of ontologies with corresponding software, a system can be extended so that, for example, the flow/stock model can coexist with others in the frequency domain or with individual-based models. The details of the scheduling and the interactions between different calculation workflows can be sorted out automatically. Another important consequence is that when space and time become part of the allowed semantics, there is no need for specially tailored tools for spatial modeling, because such functionalities can be invoked as neces-

sary by the knowledge-based system. It is for example conceivable to make a non-spatial model spatially explicit by simply describing one or more of its components as distributed in space (Villa, in press). A properly configured system can propagate the notion of space in one concept to the whole conceptual network, or mediate competing representations by operating transformations, e.g. to propagate coarse polygon data over a fine-resolution grid.

5.2. Declarative modelling for knowledge discovery

In a distributed database context, knowledge-based modeling opens novel and exciting perspectives such as "model-driven query" (Villa In press). An appropriately general version of a model can become a powerful discovery tool that can be used as a constraint over a distributed knowledge base or semantic web in order to discover new knowledge in a fully automated way. For example, by defining an instance of a food web as the realized mapping of a semantic relationship (e.g. "feeds-on") on the concept "natural-population", a system can automatically translate this definition into a query, and launch an iterative process that identifies all potential food webs represented in the population data stored in a semantically annotated database. It is easy to imagine the power deriving from matching an abstract model structure to a distributed database that's semantically annotated. By describing patterns of interest in terms of ontologies, repositories of ecological knowledge with sufficient semantic information can be used to automatically discover patterns and relationships that have traditionally taken lengthy investigations to find, even when only the necessary data are present in a database.

6. CONCLUSIONS

While by no means trivial, the software aspects of knowledge-based systems are within reach and prototypes of knowledge-based modeling systems are in use today. The major challenge is the development of the extensive knowledge base necessary to enable large-scale usage of these approaches. This challenge is not only technical: even when the difficulties of developing, storing and maintaining large ontologies are successfully addressed, their recognition and acceptance will remain difficult. Knowledge based computing will require a paradigm shift in the way ecologists think about ecological modeling. For now, knowledge based approaches are a little-understood black box in the minds of most scientists. It will require a few, compelling demonstration projects for the advantages of knowledge-based approaches to become apparent to the larger ecological community. Even then, it will require community agreement on the knowledge base itself (ontological commitment). A common resistance in the scientific community to the use of ontologies is the fear of committing to a specific conceptualization that may not fully reflect one's scientific view. Ontologies exist for eminently practical reasons, and concerns such as these are easy to answer with adequate discussion. Nevertheless, the time, training, and discussion necessary to adequately dispel them should not be underestimated.

7. ACKNOWLEDGMENTS

This publication has been partially funded under the SEAMLESS integrated project (European Commission, DG Research, contract no. 010036-2).

8. REFERENCES

GBIF (2004). Global Biodiversity Information Facility. Internet: http://www.gbif.org

GEON (2005). GEON Cyberinfrastructure for the Geosciences. Internet: http://www.geongrid.org

ISO (2004). ISO/TC211 Geographic Information Metadata standard. Internet: http://www.isotc211.org

Kepler (2004). The Kepler Project. Internet: http://www.kepler-project.org

Ludaescher, B., A. Gupta, et al. (2001). Model-Based Mediation with Domain Maps. 17th Intl. Conference on Data Engineering (ICDE), Heidelberg, Germany.

SEAMLESS (2005). SEAMLESS IP home page. Internet: http://www.seamless-ip.org

SEEK (2004). Enabling the Science Environment for Ecological Knowledge, Partnership for Biological Informatics. Internet: http://seek.ecoinformatics.org.

Villa, F. (2001). Integrating modelling architecture: a declarative framework for multi-paradigm, multi-scale ecological modelling. Ecological Modelling, Vol. **137**, No. 1, pp. 23-42.

Villa, F. (In press). A semantic framework and software design to enable the transparent integration, reorganization and discovery of natural systems knowledge. Journal of Intelligent Information Systems.

W3C (2004). OWL Web Ontology Language Guide, W3C. Internet: http://www.w3.org/TR/owl-guide