ABSTRACT

Title of Dissertation:	A NEW GEOGRAPHIC PROCESS DATA MODEL
	Femke Emma Reitsma Doctor of Philosophy, 2004
Dissertation Directed By:	Professor Jochen Albrecht Department of Geography

Processes, although the subject matter of geography, have not been represented in a manner that aids their querying and analysis. This dissertation develops an appropriate data model that allows for such a process oriented representation, which is built upon a theory of process. The data model, called nen, focuses existing modeling approaches on representing and storing process information. The flux simulation framework was created utilizing the nen data model to represent processes; it extends the RePast agent based modeling environment. This simulator includes basic classes for developing a domain specific simulation and a set of query tools for inquiring after the results of a simulation. The methodology was then prototyped with a watershed runoff simulation.

A NEW GEOGRAPHIC PROCESS DATA MODEL

By

Femke Emma Reitsma

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2004

Advisory Committee:

Dr. Jochen Albrecht, Chair
Dr. Ralph Dubayah
Dr. James Hendler
Dr. Axel Kleidon
Dr. Dagobert Soergel

Acknowledgements

Thanks to Jochen, for believing in me, and for extraordinary mentoring

Thanks to Marco, Shane, and Liz, for distracting me from my dissertation and putting up with my presentation practice sessions

Thanks to the rest of my committee, for this and that

Thanks to other friends, for much besides

Thanks to my family, for long distance love and support

TABLE OF CONTENTS

CHAPTER ONE

Introduction

“Geographies," said the geographer, "are the books which, of all books, are most concerned with matters of consequence. They never become old-fashioned. It is very rarely that a mountain changes its position. It is very rarely that an ocean empties itself of its waters. We write of eternal things.”

The Little Prince by Antoine de Saint Exupéry

1. Entrée

Geography has moved beyond chorology, beyond describing the location of geographic things and their properties (although, most people you talk to would probably still agree with The Little Prince). Nor, in contrast to Kant, is geography the science of space. Geography deals with processes, spatio-temporal phenomena that are in constant flux at one scale or another. It is here argued that if we consider process as the basic organizing concept of geography, as theoretically salient and tenable, then we must develop data models based upon this concept.

The objective of this research is to develop a modeling approach that takes this notion of flux, in the form of process, as its modeling primitive. Such an approach attempts to build from the bottom up, where a method of modeling geographic phenomena will be derived from an appropriate theory of geographic phenomena. As Couclelis has stated, “the *technical* question of the most appropriate data structure for the representation of geographic phenomena begs the *philosophical* question of the most appropriate conceptualization of the geographic world” (Couclelis, 1992: 65, original italics).

However, the author does not intend to enter into any form of metaphysical debate. Rather, it is recognized that our *observations* of constant change in the “things” studied recommends an approach to data modeling that is based on process, which takes change as its core.

The significance of this work comes from the recognition that Object Orientation is not the panacea to modeling spatio-temporal phenomena (Worboys, 2001), as the underlying theories and conceptualizations have not changed; the recent advances in dynamic modeling such as Cellular Automata and Agent Based Modeling continue to reify these same theories; the divide between the spatiality of Geographic Information Systems (GIS) and the temporality of traditional modeling software remains (Clarke *et al.*, 2001). Therefore, this dissertation presents an alternative framework that is grounded in process and is inherently spatio-temporal. Within the methodology developed, geographic processes are modeled as processes rather than inferred from system or object states.

Consequently, it is important to draw apart the technological limitations of representational systems, such as GIS or Agent Based Modeling environments, from the theoretical limitations of the representational system (Raper, 2000). The specification of a geographic process model is critical due to the limitations of current tools to query and analyze the dynamic subject of geographic research, in particular, spatially continuous processes (Worboys 2001). Rather than using concepts developed in different disciplines for different purposes, a modeling theory must be developed specifically for spatially dynamic phenomena in order to appropriately capture the unique nature of these processes. What is needed is a bottom up approach based on a solid theoretical foundation, rather than a top down approach where the tools selected and applied (typically from a narrow range of options) were not developed with geographic phenomena in mind. As was evident in the social critique of GIS more than a decade after its widespread use within academia (see for example Pickles 1995), it is often not until much later that the fundamental assumptions and theory inherent in such tools are considered or questioned.

2. Objectives

The overall goal of the dissertation is to explore a new approach to representing and simulating processes. The objectives for meeting the overall goal are sliced into theoretical, methodological, and application portions. As such, this dissertation attempts to cross the divide between theory and practice, a connection that has been found lacking in Geographic Information Science (Peuquet, 2002). The following three general objectives, including some specific clauses, are as follows:

1. Define a theory of process that considers the dynamic nature of geographic phenomena as its central construct.

    1.1 Develop a conceptualization of a theory of process

    1.2 Specify this conceptualization in an unambiguous manner, forming the basis for implementation.

2. Develop a general modeling and simulation approach that applies the theory developed in the first objective.

3. Test the general theory and methodology, developed in the first two objectives, with the specific case study of watershed runoff as a proof of concept.

From these objectives, the dissertation's structure is developed.

3. Shape

The dissertation is divided into three parts: theory, methodology, and application. Chapter 2, the first chapter of Part I, begins with an overview of the theoretical literature of GIScience and related branches of philosophy used to develop the theory presented in Chapter 3. Chapter 4, formalizes this conceptualization, and is the basis for the development of the general methodology in Part II. Chapter 5, the first of Part II, presents the methodological literature review, as the basis for justifying the methodology developed in the following two chapters. Chapter 6 describes the general methodology without subscribing to a particular language or software framework, followed by Chapter 7, which describes the implementation of the conceptual model. Part III presents the application of the methodology with a watershed runoff case study, beginning with an overview of the literature of watershed modeling in Chapter 8. Chapter 9 describes the model specification and communicates the results of the model, discussing the assumptions, problems, and implications of the simulation. Chapter 10 concludes the dissertation.

CHAPTER TWO

Theoretical Review

For instance, here is a portrait of a man at eight years old, another at fifteen, another at seventeen, another at twenty-three, and so on. All these are evidently sections, as it were, Three-Dimensional representations of his Four-Dimensional being, which is a fixed and unalterable thing.

The Time Machine, H. G. Wells

1. Introduction

There have been persistent calls in the GISc literature for new spatio-temporal ontologies and new theoretically grounded process models (Peuquet, 2001;Worboys, 2001). The significance of developing a process conceptualization and ontology based on apposite theory is in its potential interoperation between process models developed upon the same foundation, its ability to communicate the modeling constructs clearly, and its basis for methodological advances in analyzing and querying processes (to be discussed in Chapter 5). As argued by Raper, an approach that bases the selection of the representation upon the appropriate conceptualization is operationally challenging but “such an approach is perhaps the only way to break new ground in the multidimensional modeling of processes” (2000: 139).

This chapter delves into the literature of Geographic Information Science (GISc), digging for theories of objects, processes, and change, and how such “things” are conceptualized as the foundations of models. An argument is presented that throws a light on the current limitations of our theories, which draw apart structure and process. From this conventional perspective, process is interpolated between states as the process itself is not data modeled or stored, rather the future state of the system. This dichotomy evident in GISc theories plays out in conceptualizations and implementations, as will be further explored in following chapters. Therefore, the objective of this chapter is to review the primary conceptualization of process in the literature and contend for a more apposite theory of process for modeling geographic processes. This theory is based on ideas that have developed within philosophy, namely Four-Dimensionalism and Process Philosophy.

This dissertation does not discuss how reality, or geographic processes occurring therein, is perceived or conceptualized directly; this is in contrast to naïve geography which emphasizes principles from Artificial Intelligence (Mark *et al.*, 1996). Nor are natural language or common sense descriptions of processes of interest here. Rather, the focus is on the scientific conceptualization and representation of geographic phenomena for the express purpose of modeling those phenomena. Within this dissertation, a metaphysical realist philosophy is assumed, which has been suggested as the general perspective taken in GISc (Raper, 2000).

However, it is recognized that perceptions and conceptualizations are implied through the models used, and they impact on how we model and represent geographic phenomena (Peuquet, 2002). Human fiat cannot be escaped altogether. The identity of phenomena emerges through the interaction of socially driven cognitive acts with the heterogeneous structure of the world (Raper, 2000). Although individual human conceptualizations and their variations across cultures are not dealt with, conceptualizations of the human collective are the primary substance of this thesis. This may appear a somewhat more stable foundation upon which to build, yet it remains human fiat, albeit institutionalized fiat. Thus, it will be assumed that categories used to define processes are accepted and well defined, regardless of potential counter examples (Ruelle, 2000).

Furthermore, metaphysical debates on the nature of things such as physical primitives, space, time, and change, are not broached here. The focus of this thesis is the representation of this knowledge in a computational environment. Questions regarding the nature of geographic modeling primitives, modeled space, and modeled time, are those this dissertation addresses. However, there is an undeniable relationship between metaphysical things and represented things, as we cannot represent things we cannot study. New representations provide new windows onto the same subject matter, through which we can see different things.

The rest of the chapter is organized as follows: Section 2 begins the chapter by exploring the treatment of geographic "things" in GISc theory, and considers how they have been used in constructions of space, time, and change. Section 3 discusses work done thus far to develop a theoretical basis for GISc with ontologies. Section 4 presents some fundamental discussions in philosophy, which form some of the philosophical guides to the development of this thesis. Section 5 follows with a consideration of some of the problems of GISc abstractions for modeling processes that are discussed in earlier sections. Section 6 concludes the chapter, leading into the next. Please note that definitions will be developed in Chapters 3 and 4. Here, terms and concepts will merely be explored as they are described in the literature.

2. Geographic *things*

Geography is the study of processes. It is not purely about space, nor about the areal differentiation of objects in that space, rather the complex spatio-temporal nature of geographic phenomena. Regardless of whether research was once chorology framed within a certain spatial or temporal extent (Hartshorne, 1996), or a discipline of place facts congealing to create a landscape (Sauer, 1925), its current interest is in the dynamics or processes that occur within certain scales of interest. Geography's affair with processes is clearly evident when attending any class in the discipline. In an average undergraduate introduction to physical geography, students learn about atmospheric and oceanic circulation, tropical cyclones, earthquakes, erosion, floods, desertification, glaciation, and longshore sediment transport. In a human geography equivalent, students are taught about migration, the green revolution, transportation, urban sprawl, information flows, trade, sustainable development, Fordism, and growth. These are all processes.

Before going any further, it is important to provide at least some informal definitions of important terms that will be used throughout.   In terms of conceptualization, and consequent representation, the terms *object*, *process*, and *change* must be distinguished.

The term *object* is used to describe those conceptualizations of entities that at any moment in time are considered static things, such as cities, forests, and rivers (note that this has nothing to do with implementation in terms of Object-Orientation, which will be discussed in Chapter 5). The dynamics or change in the object is derived from sequences of object states, where attributes are aspects of the object that define its state at a time. The term *process* will be used to describe those entities that at any moment in time are conceptualized as dynamic things, such as erosion, migration, and fluvial deposition. Processes are *not* summaries of object changes but a category to themselves. *Change* expresses the difference between the states of an entity at two instances of time, be that entity an object or process.

2.1 Primitives

GISc has focused upon objects as its primitives. These objects include things we might see on a map, represent as a polygon, point, or line feature in a GIS, or model as an agent in an Agent Based Model. For example, geologic fault lines, census units, and rivers are typically expressed as GISc's objects of study and representation (Raper and Livingstone, 1996). In a classic (and representative) paper by Raper and Livingstone (1995), the objects of study in their research in coastal geomorphology are material or substantive things, such as geomorphologic units or zones and environmental features, all of which are modeled as objects. This is also evident in the ontological work in GISc, as will be further discussed below (Casati *et al.*, 1998;Fonseca and Egenhofer, 1999;Frank, forthcoming;Smith and Mark, 1998;Thomasson, 2001;Varzi, 2001), and in the growing body of literature on spatio-temporal data representation, formalisms, and implementations. Both these theoretical and applied research areas remain focused on the object as the primitive that changes through an absolute or relative notion of time (Couclelis and Liu, 2000). Hence, developments in data models for dynamic geographic entities reflect this focus (for a review of data models see Borges *et al.*, 2001;Tryfona and Jensen, 1998)

Kuhn (2001) notes some of the reasons for such object orientation in geographic and other information systems, including:

- the static roots of GIS are found in cartographic origins;
- an emphasis on attributes and relationships rather than process and change;
- the weakness of logic-based formal languages in dealing with operations and semantics;
- a presumed priority of objects in human (spatial) cognition

This has been further expressed in a recent publication on the foundations of GISc where the geographic information considered as pertinent primitives are boundaries, regions, neighborhoods, and landmarks (Duckham *et al.*, 2003). Qualitative spatial reasoning also centers itself on objects, in particular cognitively salient spatial objects, with point or region based reasoning forming its basis (Cohn and Hazarika, 2001). This focus upon objects has formed the basis of representations of processes in what can be termed “process-form” models (Miller, 2003b;Peuquet, 1994). Processes are then modeled as modifying these objects or their attributes.

The growing body of literature on spatio-temporal data representation, formalisms, and implementations, remains focused on the geographic object as the primitive that changes through an absolute or relative notion of time. Commonly these static features describing the state of a system are conceptualized as objects or fields, discrete or continuous variations in attributes of the system (Couclelis, 1992b). Such objects are defined as an atom that has location specified at a point and associated attribute information (Goodchild, 2003). Hence, developments in geographic data models for spatio-temporal geographic things cater to geographic objects, which define static things (for review of data models see Borges *et al.*, 2001;Tryfona and Jensen, 1998). These objects and their relationships are the predominant primitives of GISc.

However, alternative conceptualizations of the world that focus on the spatio-temporal nature of data are now being developed. The focus of such work is the description of the dynamics of spatio-temporal entities that form the basis of a description of processes. For example, Frank (2001) presents a specification where the objects themselves are time dependent functions, that is, the primitive things are functions in an absolute temporal framework. Although it is not clear how the input and output of these functions is conceptually different from an object to which a behavioral rule is applied, nor has a fully-fledged theory or model been developed. Another, albeit static, representation of a spatio-temporally extending primitive is found in transportation studies that focus on the trip as their unit of study (Shaw and Wang, 2000). Similarly, Galton discusses the nature of event based conceptualizations (2001), and Chen and

Jiang use events as primitives for a land subdivision system that stores changes in land parcels and allows for event queries (Chen and Jiang, 2000).

From this discussion it is evident that one predominant primitive can be drawn out that forms the basis for the majority (if not all) conceptualizations and representations in GISc, namely objects. Unchanging objects that represent substance or mass at an instant of time or over an extended period of time, such as socio-economic units (Frank *et al.*, 2001) or the measurable attributes of things such as land use types (Peuquet, 1994). Change is then derived from these primitive objects by calculating the difference between temporally successive instances of the same object. The more recent event oriented approaches, while useful for exploring existing data, do not appear to be applicable to process modeling.

2.2 Process and Change

There are many different definitions of *process* in GISc, but all have a common interest in the concepts of time and change (as evidenced at the 2002 Research Workshop on Action-Oriented Approaches in GISc: http://www.spatial.maine.edu/~actor2002/). The term event also arises in related discourse, which similarly suffers from a multitude of meanings. However, from the definitions provided in the literature a general consensus may be derived at an abstract level, where a process is considered as something that results in the change of an object (Bian, 2000;Claramunt *et al.*, 1997;Forbus, 1984;Galton, 2000;Thériault *et al.*, 1999). Typically a series of changes represented by a sequence of related states defines the full nature of the process (Worboys, 2001;Yuan, 2001). An event, in contrast, is defined as the occurrence of something significant or of interest with clear temporal boundaries (Worboys, 2001;Yuan, 2001). As explained by Galton, "an event is a chunk of change picked out as an individual from the ongoing flux" (2000: 207). Some argue for the view that events are composed of processes (Claramunt and Thériault, 1996;Yuan, 2001), and others that processes are composite events (Worboys, 2001).

There has been work on expressing processes as taxonomies of changes. These taxonomies describe changes in entities or sets of entities in temporal steps, where the dynamics is interpolated between these steps (such as: Claramunt and Thériault, 1996;Claramunt *et al.*, 1997;Miller, 2003b;such as: Thériault *et al.*, 1999). Other work has considered existential changes through changes in object identity, where it is recognized that the changes that occur between snapshots should be explicitly stored as opposed to temporally interpolated (Frank, 2001;Hornsby and Egenhofer, 2000;Stefanakis, 2003). There has also been brief mention of process as the basis of a conceptual modeling approach, yet without development into a full conceptualization and implementation (Renolen, 2000).

These views of processes are based on prevalent views in past and present theoretical geography, where "study of space-time processes concerns how spatial arrangements are modified by movement or spatial interaction" (Gatrell, 1983: 2). Or more clearly expressed by Thornes and Brunsden, who state that "[w]e have to remember that processes are simply combinations of circumstances which change the state of our system over time" (1977: 27). This dichotomy of static objects or structure and process, is presented as structure explaining process (Abler *et al.*, 1971). For example, from spatial patterns of sediment, stratigraphy, or drainage patterns, we infer certain hydrological events or processes. Traditionally the focus has been to understand process through an analysis of the patterns or structures they produce, trying to establish a relationship between form and process, between what we can measure and what we cannot (Getis and Boots, 1978).

In terms of approaches in GISc to model these dynamic phenomena, the focus has been on time-varying spatial information as spatio-temporal information. Spatio-temporal extensions typically imply the extension of spatial objects through time, thus the focus remains on objects and state changes. For example, "a 'storm' is modeled as a 'moving' object with changing properties (e.g. intensity) and shape over time in an environmental application" (Tryfona and Jensen, 1998: 6). Change is then derived from spatio-temporal interpolation between states of a system, where vectors of movement can be used to describe change between successive states (for example Raper and Livingstone, 1996). As a consequence, processes are inferred rather than directly represented, as expressed by Raper and Livingstone, "four dimensional form and structure may be correlated with the energy inputs to the system" (1996: 9).

3. Ontological Foundations

The conceptual fundamentals of GISc have more recently taken an ontological turn. In what follows ontologies will be explained and their application in GISc discussed. This forms the backdrop for the expression of the conceptual model in an ontology in Chapter 4, and its use for running a simulation in Chapter 7.

3.1 Ontologies for Primitive Modeling

Loosely, an ontology is used to define a set of existing things and their attributes and relationships. These entities in an ontology specify an ontological commitment to a particular conceptualization of the world. What "exists" in our models is defined by our ontology, providing us with our universe of discourse (Gruber, 1993). As a result, if this universe of discourse somehow fails to acknowledge or misrepresents an important part of the metaphysical reality being modeled we may gain spurious results from our models.

In defining *ontology* it is important to distinguish between two main uses of the term, and to define the way it will be used in the rest of this dissertation. The distinction made is between its use in philosophy and computer science or AI (Artificial Intelligence). In philosophy, ontology signifies the body of knowledge and research which is concerned with the investigation of the nature, constitution, and structure of reality, also known as metaphysics (Audi, 1999). Elsewhere, ontology pertains to a categorical system defined by a view of the world that is purposefully restricted, that is, it represents a set of things defined by an individual as their world of primitives. This binary distinction is not universally accepted, for example, Guarino (1998) presents a tripartite classification by further distinguishing ontology as a discipline from the philosophical investigations of those working within the discipline, such as Aristotle and his theory of categories.

However, it provides a useful framework within which to focus upon ontologies. Henceforth, all mention of ontology, unless otherwise specified, refers to its use in AI.

Gruber (1993) defines an ontology as an explicit specification of a conceptualization. The term conceptualization is used in this definition to denote an abstract, simplified view of the world that we wish to represent for some purpose. Every knowledge base or knowledge-based system is committed to some conceptualization, either explicitly or implicitly (Gruber, 1993). Unfortunately this early attempt at a definition of ontology for information systems left room for many possible interpretations, ranging from a catalog (such as its application in Amazon.com) to more expressive and complicated ontologies using axioms of modal logic (Smith and Welty, 2001). Here, ontology will be used as a declarative taxonomy of entities used to represent some part of the world, where what exists is that which can be represented, and thus defines the universe of discourse (Gruber, 1993).

Ontologies may be distinguished at a range of granularity. From a coarse ontology that "may consist of a minimal set of axioms written in a language of minimal expressivity, to support only a limited set of specific services, intended to be shared among users which *already agree* on the underlying conceptualization" (for example Gangemi *et al.*, 2002), to a fine grained ontology which "gets closer to specifying the intended meaning of a vocabulary (and therefore may be used to establish consensus about sharing that vocabulary, or a knowledge base which uses that vocabulary)" (Guarino, 1998: 8). A distinction is also made between upper-level ontologies, domain ontologies, and application ontologies (Brodeur *et al.*, 2003;Guarino, 1998).

- Upper-level ontologies describe very general concepts such as space, time, entities, and relationships, which are independent of a particular domain but are informed by the abstract properties of all of the domains it represents.
- Domain ontologies describe the vocabulary related to a specific domain, such as hydrology, transportation geography, or coastal geomorphology. The terms introduced in the upper-level ontology are refined and focused by the domain ontology.
- Application ontologies describe concepts pertaining to a particular application within a domain, specializing the terms of an ontology further. Examples of applications that might be described by such ontologies include watershed runoff, wave refraction on a headland, or the spatial response of a particular species of cacti to changes in climate.

At any of these levels, an ontology may take a variety of forms. The degree of formality used in defining the vocabulary of terms may vary along a continuum from loose expression in natural language to rigorous formalization of terms with formal semantics, theorems, and proofs, such as first order logic (Uschold and Gruninger, 1996).

The primary reason for developing ontologies is to be able to share knowledge in a manner that aids understanding (Gruber, 1993). For modeling processes, ontologies are needed in order to develop conceptually sound models, effectively communicate these models, enhance interoperability between models developed in different domains, and provide the opportunity for reuse and sharing of model components (Albrecht, 1999;Fonseca and Egenhofer, 1999;Kavouras and Kokla, 2002;Smith and Mark, 1998;Uschold and Gruninger, 1996;Visser *et al.*, 2002). For example, this is particularly pertinent for Earth Systems Science in developing models of large-scale systems, as is typically depicted in the Bretherton diagram of biospheric cycles (Figure 1). Each box in the Bretherton diagram can be further dissected into sub-components. Linking models from different disciplinary domains can be enhanced with a single underlying ontological framework upon which they are all built, thereby allowing these models to easily "talk" to each other. This involves semantic interoperation rather than syntactic, where the meaning of modeling primitives, such as use of space, time, and change, is the same across components. As expressed by Guarino, "even if two systems adopt the same vocabulary, there is not guarantee that they can agree on a certain information unless they commit to the same conceptualization" (Guarino, 1998: 8).

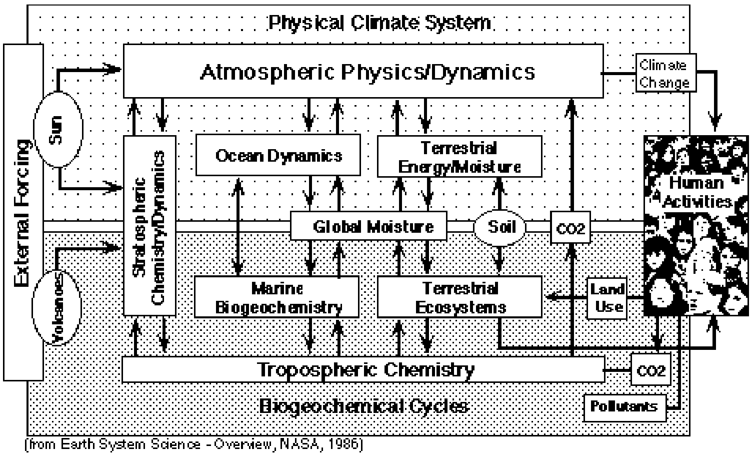


Figure 1. The Bretherton diagram

Source: Earth Systems Science Overview: a program for global change, NASA 1986

3.2 Ontologies Thus Far

Currently ontologies have been and are being developed for representing and sharing spatial knowledge. Research on ontologies is becoming increasingly widespread in computer science, particularly in Artificial Intelligence, Computational Linguistics, and Database Theory (see Guarino, 1998 for a reference list of early application areas). Ontologies are also making inroads into geography via philosophy (Bittner and Smith, 2003;Casati *et al.*, 1998;Smith and Mark, 1998), GISc (Fonseca and Egenhofer, 1999;Fonseca *et al.*, 2002), and Geomatics (Bittner and Edwards, 2001). Ontologies have also recently become a research theme for the UCGIS (University Consortium for Geographic Information Science) (Mark *et al.*, 2002).

Thus far, most ontological work in geography, in particular in GISc, has involved describing static spatial objects (Casati *et al.*, 1998;Fonseca and Egenhofer, 1999;Frank, forthcoming;Smith and Mark, 1998;Thomasson, 2001;Varzi, 2001). The units of ontological exploration most commonly ascribed to are objects that can be represented on a map or in a GIS, such as political boundaries, mountains, and islands, whose spatial extent, parthood relations, and spatial relations are explored (Casati *et al.*, 1998;Smith and Mark, 1998;Thomasson, 2001).

> 'What kinds of geographic things are there? Two categories can be distinguished, corresponding to a traditional distinction between physical and human geography. On the one hand there are mountains, rivers, deserts… On the other hand there are socioeconomic units: nations, cities, real-estate subdivisions—the spatial shadows cast by different sorts of systematically organized human activity.'(Casati *et al.*, 1998: 79)

The dynamics of these objects have also been considered through the creation of ontologies for moving objects. For example, Tryfona and Pfoser developed an ontology where the trajectory of an object forms a primitive of the ontology (2001).

Alternatively the world may be described as composed of both static objects and processes, where objects engage in processes. This is a more compelling view with regards to common sense. This ontological approach considers both static objects and processes and the relationship between these types of entities, recognizing two complementary ontologies of the world that form their meta-theory (Bittner and Smith, 2003;Gangemi *et al.*, 2002;Grenon and Smith, 2003;Reitsma and Bittner, 2003). An example of this view is the SNAP/SPAN meta-theory developed by Bittner and Smith (Bittner and Smith, 2003). One ontology is directed towards enduring entities (called SNAP), those entities that are traditionally thought of as objects, such as people, buildings, and mountains, and their qualities, functions, and so forth. The other is directed towards perduring entities, or what they term processes (called SPAN), and is based on the contemporary analytic metaphysics of four-dimensionalism (Sider, 2001). It is argued that to describe processes in an appropriate and complete manner we cannot have a SPAN ontology without a corresponding SNAP ontology and vice versa. For example, the processes of migration and gentrification cannot be understood without the enduring entities of humans, which are involved in these processes. See also Kuhn (2001) who links objects to activities through affordances.

Relevant upper-level ontologies that have developed in other fields include the Process Specification Language (PSL) developed by The National Institute for Science and Technology (Grüninger, 2003;Menzel and Grüninger, 2001), and OWL-S (formerly known as DAML-S). The PSL has been developed to facilitate interoperability of process information among manufacturing systems. At its core, "there are four kinds of entities required for reasoning about processes – activities, activity occurrences, timepoints, and objects" (Grüninger, 2003: 605). OWL-S is an ontology for web services on the Semantic Web which forms an upper-level ontology for processes that service agents can implement. Both PSL and OWL-S are founded upon absolute notions of time and do not yet incorporate spatial concepts (see Chapter 4 for further discussion).

As should be evident, the presence of process in the ontologies discussed above has either been non-existent or as part of a dual ontology of objects and processes, which use or modify those objects. This reflects the theoretical foundations discussed above in Section 2, and presents the opportunity taken in this dissertation to develop a modeling methodology based on an ontology of processes alone. The advantages of such a process oriented approach will be explained below in Section 5.

4. Philosophical Foundations

After considering the "things" that have colonized GISc, it is of value to briefly describe some theses found in the philosophy of metaphysics. These arguments form the philosophical milieu for the thesis presented in this dissertation, namely, considering process as primitive. In particular, the positions of four-dimensionalism and process theory will be sketched.

4.1 Process Philosophy

Process philosophy presents the argument that the fundamental "things" of reality are processes; that is, flux, as opposed to stasis, forms the metaphysical basis. This has been termed one side of the "great divide" in metaphysics, the other being substance or objects (Rosenthal, 1999). In process philosophy, the view is that "not only is everything changing, but all *is* flux. That is to say, *what is* the process of becoming itself, while all objects, events, entities, conditions, structures, etc., are forms that can be abstracted from this process" (Bohm, 1980: 48, original italics). Things simply *are* what they *do* (Rescher, 2000). In contrast, "the commonsense view, enshrined in European language (not all languages), that the most concrete realities to which abstractions are to be applied, the real "subjects" which have "predicates" are things, individuals which change from one actual state to another – a person, a tree, a mountain, a star – not happenings" (Hartshorne, 1998: 397).

Process philosophy is typically associated with pragmatism, its substantial formulation being attributed to Whitehead who proclaimed creative becoming as a universal category (in particular, see Whitehead, 1969). Yet the roots of process philosophy are commonly founded in the fragments of Heraclitus' work, which reorients reality to the flux underlying even what on the surface looks static. Heraclitus shows how the sameness of parts does not guarantee persistence, persistence depends on change (Moravcsik, 1991). This is in contradistinction to classical Greek philosophy, which emphasized a metaphysics of being and substance.

Whitehead created his philosophical system around the generalization of the flux of things, as opposed to the permanence of things. He argued against scientific materialism, which presupposes an irreducible "matter, or material, spread throughout space in a flux of configurations" (Whitehead, 1998: 274). These irreducible things are described as abstractions, with which we fall into the fallacy of misplaced concreteness if taken for concrete realities. Furthermore, with this view of identity as abstraction, to see an object as always the same object we must abstract from what is new in it at each moment (Hartshorne, 1998). As expressed by Mead:

> By taking time seriously, we realize that the seemingly timeless character of our spatial world and its permanent objects is due to the consentient set which each one of us selects. We abstract time from this space for the purposes of our conduct. Certain objects cease to be events, cease to pass as they are in reality passing and in their permanence become the

> conditions of our action, and events take place with reference to them (Mead, 1998: 371)

In contrast, processes are defined as the becoming, the coming into being, of temporally structured “Actual Occasions” (the primitives in Whitehead’s metaphysics) and their relations. For example, rather than considering the movement (predicate) of a particle of soil (subject) to another location (object) as three distinct things, the primitive thing is an instance of the process of erosion, which encapsulates all of the above.

However, process philosophy is an approach, not a consensus or a completed thesis (Rescher, 1999;Rosenthal, 1999). It provides a set of concepts that have not been formalized into a coherent theory. Furthermore, as with all things, there are a range of intermediary positions between the two poles of substance and process metaphysics (for example, Simons, 2000).

4.2 Four-dimensionalism

Orthogonal to the debate over the “things” of reality, is that discussing the temporal nature of these things. That is, how these things change and persist over time through the gain and loss of parts. Where process philosophy is about primitive things, 4Dism (four-dimensionalism) is about how these things persist over time, regardless of whether these things are objects or processes. 4Dism and 3Dism (three-dimensionalism) are two opposing views of temporal persistence that form the primary framework within which the nature of things over time are considered. However, as seemingly with all theories, there are a range of intermediary positions as well as strong and weak forms of 4Dism and 3Dism (Brogaard, 2000;Parsons, 2000;Sider, 2001). For the sake of clarity and brevity only the two main theses will be introduced.

3Dism, also known as endurantism, considers the endurance of things (endurants) over time, where persistence is maintained without temporal parts (van Ingwagen, 2000). Within this view, a physical object is wholly present at all times it exists. Persistence is therefore a matter of trans-temporal identity, where a 3D entity that is wholly present at a certain place at one time is one and the same entity at a possibly (but not necessarily) distinct place at another time. 3Dism deals with changes in objects by temporally modifying the predicate (indexicalism) or the copula (adverbialism) as opposed to the subject itself (Balashov, 2000). Thus change is an instantiation of different properties at different times. As a consequence of this view, one of the main problems of 3Dism is the maintenance of identity over time when an entity gains and loses parts. Solutions range from Chisholm's mereological essentialism, which prescribes the permanence of parts for the maintenance of identity, to weaker forms of 3Dism (Brogaard, 2000). For example, a 3Dist extremist would consider a frontal system as the same frontal system through its preservation of parts such as various air masses and clouds. If at any moment in time those parts change, which it constantly does, its identity changes.

In contrast, 4Dism, also known as perdurantism, tackles the problem of identity through change with perdurants, which have temporal parts. Persistence is a matter of having temporal parts, where a perdurant is only partly present, a temporal part, at each instant or interval of time. Perdurants are four-dimensional objects, which are extended in time and space. 4Dism is a world spread out in time populated by space-time worms, or perdurants. Support for 4Dism lies in diffusing various paradoxes that revolve around the problem of maintaining identity over time (e.g. the ship of Theseus problem). 4Dism does not answer these problems directly but provides a framework for their dissolution into definitional problems (Sider, 2001). For example, a tropical cyclone, while gaining and loosing parts, may be defined as having a number of temporal parts, such as a being a tropical depression and a tropical storm. Its identity depends on the definition of how those temporal parts merge to form the whole.

The contrasting perspectives do not suggest that an enduring view cannot fill a 4D region of space-time, the difference lies in that "the perdurantist will insist that the object does so in virtue of being a 4D entity having extension both in space and time, whereas the endurantist will deny that the object itself is a 4D entity" (Balashov, 2000: 329). In 3Dism, what occupies a 4D region of space-time is the mereological sum of a class of objects that individually occupy its time-like slices, all being occupied by one and the same 3D enduring entity (Balashov, 2000). The difficulty lies in determining what forms the parts of this mereological sum.

As a caveat, the case where endurants and perdurants are considered analogous to continuants and occurrents or objects and processes, respectively (with participation or dependence relationships between each pair of entities), is expressly denied here (cf. Bittner and Smith, 2003;cf. Gangemi *et al.*, 2002). 3Dism and 4Dism can both capture processes and objects. Rather, they deal with the spatio-temporal nature of objects and processes differently.

Although not typically addressed in conceptualizations, the temporal persistence of things tends towards the three-dimensionalist view. There have been some brief, yet undeveloped, discussions of the endurantist versus perdurantist perspectives in GIScience, such as Raper (2000) who argues that most current work is epistemologically dependent on endurantist forms of space and time discretisation, yet it remains focused on objects rather than processes. 3Dsim is particularly evident in the nature of models, where the objects modeled are identified as wholly present at each instant of time, and identity is maintained through the object itself. Processes are inferred from change in objects that sweep through spacetime. For example, defining a process as the succession of actions or events through time requires a logical connection or explanatory mechanism between these changes, which joins them together (Harvey, 1969). For processes, 4Dism is more appropriate, particularly as the process constantly gains and looses parts, and thus cannot be defined by those parts.

5. Geographic Holes

From the above discussion of the underlying conceptual and ontological constructs of GISc and the two philosophical digressions, some holes may be poked where processes are concerned. One of the primary difficulties is the distinction drawn between object (also described as structure) and process. Such "views assume that structure and process are two different things, which they are not; structures of the real world are simply slow processes of long duration" (Blaut, 1961: 4). Coffey (1981) argues for an alternative view, taken up in this dissertation, suggesting that process captures both structure and motion. This conceptual dichotomy between object and process is something this thesis attempts to address by presenting a single primitive as fundamental, rather than two.

Instead of describing the world as a set of objects that undergo change, we may describe the world as a set of processes that embody change. Here "nothing in the physical world is purely spatial or temporal; everything is process" (Blaut, 1961: 2). Change should be at the core of data models of these processes, where representing geographic phenomena as processes is more fundamental than as a collection of objects and relations, the current approach. We cannot effectively model and represent the dynamics of processes with interacting or moving objects. Rather than modeling future system states, future process dynamics should be modeled.

In terms of conceptualizations, absolute views of space and time assume that structure and process are two different things (Blaut, 1961). As has been noted, geography has long been considered the science of spatial relationships between objects, these spatial relationships forming the basis for processes (Chapman, 1977). It appears that upon this foundation most underlying GISc theory lies. What are needed are representations that capture the causal relativity of interacting processes, all within a relative notion of space-time. Worboys comes closest to the work of this thesis, where he states that "it is not *time* that is the key to conceptual modeling of dynamic systems, but *change* and related constructs such as event and process" (Worboys, 2001: 129).

In formalizing a process conceptualization, it is important to specify a salient ontology for processes as “different things become thinkable in different ontologies” (Johansson, 1989: 7). Consequently, different things can be explained and understood within different ontologies. Present and past developments in ontologies for geography are not suited for modeling geographic processes. Most work involving the development of theories and ontologies for change is for discrete mobile objects (for example Hornsby and Egenhofer 2002) or for static entities which fall in the two-dimensional mapping paradigm that is a “restricted projection of a four-dimensional world…[which makes]…it impossible to fit many entities in a geographic ontology” (Raper, 2000:118). Similarly there has been considerable discussion on systems of granularity of discrete and static objects (Bittner and Smith, 2001), and some recent work on spatial and temporal scale relations (Pereira, 2002). Yet for processes, there is a dearth of research on considering them explicitly as the subject of formalization and no evidence of any meta-theory or ontology of process.

The advantage of such a theoretical approach as a basis for modeling the process is that we can track the dynamics of the process, rather than infer it through the interaction of objects. Applying the single primitive of process for modeling theory also adds value by providing leverage for querying processes and analyzing them. For example, in global climate modeling virtually the same future state of increased temperature can be modeled as a result of two very different changes to the model, an increase in solar luminosity or an increase in CO2. It is not immediately obvious which processes, such as heat transport or a change in cloud optical depth, caused these results. Representing the processes in operation at each time step would enhance our ability to understand the results of the model and presents an alternative view of the world. Such a theoretical foundation allows us to address questions that are not directly answerable with current object centered formulations, which focus on the state of a system that results from the process; it allows us to ask questions such as:

- Where is a process operating at a particular instant of time?
- How has the process changed over time?

Currently questions of the object centered approach are restricted to two basic types "what is at a specific location?" or "where is a certain attribute?", the composition of which define the realm of possibilities (Goodchild, 2003). Furthermore, in modeling processes we may gain insight into their causal relations by storing their interactions. Questions regarding how the rules of the process affect the dynamics of the process may be better explored.

6. Teleporting to Conceptual Worlds

The difficulties of current conceptualizations and ontologies in accommodating geographic processes forms a springboard for the consideration of an apposite upper-level ontology for modeling geographic processes based upon a germane conceptualization. The next chapter develops the conceptualization, which is followed by the ontology in Chapter 4. Such an ontology will form the basis of a domain ontology; the first step towards a realization of a concrete implementation in a computational model (Smyth, 1998).

The development of a conceptualization and ontology prior to the model makes clear the assumptions underlying the implementation. Furthermore, such a specification of the needs of the model is unfettered by implementation requirements. Beginning with an apposite conceptualization avoids the representational inabilities of software environments from driving the nature of the model (Raper and Livingstone, 1995). However, limitations will build as the chapters progress towards the general methodology and the specific case study.

CHAPTER THREE
Conceptualizing Process

By cosmic rule, as day yields night, so winter summer, war peace, plenty famine.
All things change.

Heraclitus. Fragments.
Translated by Brooks Hamilton, 2003

1. Introduction

This chapter presents a conceptualization of geographic phenomena that is based on the single primitive of process. Conceptualizations are necessary because even though two models might adopt the same vocabulary, there is no guarantee that they will interoperate unless they commit to the same conceptualization (Guarino, 1998). Moreover, "[i]t is likely that models built on the basis of significantly different conceptual frameworks will produce unreliable results when coupled together"(Couclelis and Liu, 2000: 2).

The objective of this chapter is to develop a conceptualization that will form the theoretical basis for an implemented model. Where an ontology is language-dependent, a conceptualization is language-independent (Guarino, 1998). The language employed for this conceptualization is the most expressive available, that of natural language. In this case, that language is English. This is purposefully done in order to present the theory with as few limitations as possible, thus it ignores any restrictions that may arise from the expressivity of ontology languages, the representational capacity (or incapacity) of the computer, or the software tools available for implementation.

In the conceptualization developed, some of the theories of formal ontology are considered, such as mereology (Simons, 1987) and granular partitions (Bittner and Smith, 2001;Smith and Brogaard, - to appear), and hierarchy theory is introduced (Allen and Starr 1982, Ahl and Allen 1996). This conceptualization will be specified in an ontology in the following chapter, forming the basis for a modeling approach developed in Part II, and applied in Part III. This notion of process follows from the literature review of Chapter 2, where the limitations of a worldview that takes static entities as its primitives provides impetus for an alternative view. As expressed by Sowa, "[t]he choice of representation can have a major effect on the way the reasoning is carried out and on its ultimate success or failure" (2000: 245).

The rest of the chapter is organized as follows: Section 2 characterizes the single primitive of this conceptualization, namely process. Section 3 follows by defining the parts of the process, which leads to Section 4 in discussing the relationships among processes and between processes and their parts. Finally, Section 5 leads on to the development of an ontology in Chapter 4.

2. Process

Processes are the fundamental entity of this conceptualization, the single primitive. It must be noted that static entities or objects do not exist within this conceptualization. Such static entities might include sediment, humans, a body of air, a stand of trees, or a transportation network. Objects are considered abstractions of processes, that is, processes with dynamics that are ignored. Thus, in contrast to typical approaches that oppose structure and process, where process operates upon structure, structure is considered as merely slow process.

What then is a process? Process is difficult to define without introducing other concepts such as space, time and change, which are inseparably intertwined. As with any mathematical or linguistic system, primitives are hard to define, they simply are. Given this difficulty, the next best thing is to specify some of the properties of processes so we may at least know one when we see it (Munsat, 1969). Thus, in what follows, the characteristics of geographic processes are considered.

2.1 Process Undefined

According to the Oxford Companion to Philosophy (Honderich, 1995), a "process is a series of changes with some sort of unity, or unifying principle, to it". How, then, do we define this unity, or delineate processes? The act of defining a process is necessarily a bounding action, that is, we are cutting a continual flux into parts, spatio-temporal parts. Can we simply draw lines around processes in the same manner as static entities or map-able things? Should we consider the differences between processes that are more evidently spatio-temporal jumps in the continuum of flux, such as the edge of a flood at any moment in time, than others that are not so clear, such as the border of a mountain?

These problems reflect those of static entities, where bona fide objects, those that exist independently of human cognitive acts, have been distinguished from fiat objects, those that do not exist independently of human cognitive acts (Smith, 1995). One or the other extremity of this dichotomy has been argued for, where either every object is a result of human fiat or every object is bona fide (Smith, 1995). It may appear that physical processes, such as tornadoes, volcanoes, and the fluvial deposition of an alluvial fan, align more closely with bona fide processes, and human processes, such as globalization, gentrification, and the diffusion of agricultural innovation, with fiat. However, processes, as with static entities, do not fall so clearly into these two classes.

In limiting the scope here to a conceptualization for modeling geographic processes, it is argued that the distinction between bona fide and fiat is not useful in this context nor does it form a hurdle to process definition. The bona fide nature of spatial processes is defined by a disciplinary act of fiat that delineates the thresholds or bounds of the process, that is, spatial and temporal grain and extent. In a sense, we have institutionalized processes through peer review, bona fide processes through consensus. Such an act of fiat is based on perceived discontinuities in the continuum of flux. There are plenty of arguments against such a clear cut set of processes, for example, consider defining the extent of globalization. However, for the sake of modeling these processes it is impossible to scientifically model, analyze, and compare processes without such bounds. Whether it be the wind speed of a tropical storm defined as ranging from 35-63 knots (Christopherson, 2001), or the size of a sampled population in more qualitative research, the bounds of these processes are defined and accepted (at least in the research applying the technique). Thus all processes are considered a result of disciplinary fiat, which ossifies or concretizes our bona fide objects of study.

Focusing on processes does not imply that merely a switch has been made from object or structure to process, from material substance to laws of flux, as in the dichotomy described in Chapter 2. Rather process is here defined to encapsulate both material substance and rules or physical laws that specify the behavior of the process. As the single modeling primitive, process encapsulates both matter and movement into one "thing".

This representative entity forms the data modeling primitive for processes, which can be expressed in tuple form as: (*x1*, *y1*, *x2*, *y2*, *st,* {*a1*, *a2*,...}, {*r1, r2, …*}), or graphically as a (node,edge,node) triple as illustrated in Figure 1 below. Each (node,edge,node) will be henceforth referred to as a *nen*. The location of the process is identified by x1, y1, x2, y2, which expresses the spatial extent of the process. The s*t* represents the spatio-temporal granularity of the process, which may be a function of the amount of energy that initiates the process. For example, given some threshold breaking push, the spatio-temporal granularity expresses how far and over what time period the process will operate in response to that push. The set {*a1*, *a2*, ...} defines the set of attributes of the process. The set {*r1, r2, …*} defines the set of rules of the process that govern its dynamics and interaction with other processes. For example, a set of rules for modeling the process of sediment transport in the longshore may define the spatio-temporal extent of an instance of that process as 5m/hour, depending on various relationships it holds between other processes operating in the nearshore.

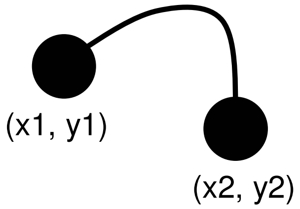


Figure 1. Process representation

Note that this is only a representation of a point process, which might best represent processes such as runoff in a watershed. It can also be extended to areal or linear feature and into the third spatial dimension.

2.2 Process Properties and Behavior

A process may have a range of properties and functions pertaining to its domain. The properties or attributes define the nature of the process at any instant of time. The functions describe the behavior of the process, defining the limits of the process' operation, which may be spatially or temporally based. For example, a process may depend on the operation of nearby processes which provide energy or matter for its initiation, such as the process of Hortonian overland flow requiring the process of precipitation.

Processes, although represented as a discrete unit, are conceptualized as continuous in their spatio-temporal dynamics. As expressed by Ahl and Allen, "[a]rguments emphasizing either continuity or discontinuity turn on the usefulness of the characterization of the phenomenon, not upon ontological assertions about nature being truly discrete or otherwise. One can always narrow the measurement grain or extent, or change focal definitions, so as to find continuity where before there had been apparent discontinuity" (Ahl and Allen, 1996: 135).

Each process contains a set of behavioral rules, which determine its instantiation, dynamics, and end. These rules define the thresholds of operation in relation to other processes and the consequent action when such thresholds are exceeded. Therefore, at any instant of time, the system that is being modeled is characterized by a particular set of processes in operation. The advantage of representing models as systems of rules and constraints is the ability to undertake non-deterministic reasoning, which "can be more explicitly and robustly represented than with purely mathematical models"(Peuquet, 1994: 59).

2.3 Process Identity

A process is composed of patterns of sub-processes, which form the basis of process identity. We may take a leaf from the work of Heraclitus (what little there is), where identity is defined by patterns of change, not by unchanging characteristics of objects. A process persists if it maintains a certain pattern of change, a certain set of sub-processes. For example, if the parts of the process El Niño no longer operate, parts such as shifting atmospheric wind patterns and blocking of upwelling in the Eastern Pacific by warm surface water, then El Niño is no longer.

Furthermore, in contrast to most theories of temporal identity, process identity is based on the gain and loss of its material parts. The identity of a process is maintained by changing of parts, not by unchanging characteristics of objects or by the sameness of parts. For example the process of migration does not depend on the same human individuals; rather they enter and exit the process. Likewise air molecules enter and exit the process of a tropical cyclone, and yet the tropical cyclone persists.

The research domain of the process provides the definition that allows us to identify its beginning, persistence through time, and end. For example, tropical depressions are defined as wind speeds of up to 34 knots, and hurricanes are defined as wind speeds greater than 65 knots (Christopherson, 2001). These institutionalized processes are taken as the primary mode of process identification.

2.4 Process Space-time

The spatio-temporal nature of a process is relative to other processes. In contrast to more common notions of time where processes are thought to take time (Munsat 1969), for modeling geographic processes it is impossible to draw apart space and time, hence the definition of a process that is spatio-temporally extended. Space-time is a measure of change, rather than the tick of an absolute clock monitoring spatial change. For example, the dynamics of a tornado are measured in space-time units, such as km/hour. However, there is no temporal change without space, nor is there spatial change without time. Processes create space-time through change.

Instead of populating an absolute temporal framework with states, events, and processes (Galton, 2000), processes define the spatio-temporal framework. Processes create a relative space-time through their interaction. Absolute space and time are imposed frameworks, processes do not decide when to operate based on some spatial or temporal marker, they occur based on an interaction with some other process. Thus, time and space are not exogenous variables; they are encapsulated within the process instance through definitions of spatial and temporal extent and granularity and through behavioral rules of interaction.

The spatio-temporal boundaries of processes to consider may be organized into four classes, spatial grain, spatial extent, temporal grain, and temporal extent. Extent is concerned with the spatial size of a process, in (x, y, z) dimensions, or the temporal length or duration over which those phenomena operate (Lam and Quattrochi, 1992). For example, continental glaciers operate over a much larger extent, both spatial and temporal, than thunderstorms. Grain is used to refer to the finest distinction represented in the model, often referred to as resolution (Albrecht and Car, 1999). Temporal grain refers to the observed frequency of behavior, where frequency is defined as the number of cycles a phenomenon completes within a specified time interval. For example, the temporal grain of the tide is approximately once every twelve hours. In the context of this dissertation, fast behavior is defined by high frequency and slow behavior by low frequency (Ahl and Allen, 1996). For example, the movement of a glacier occurs at a much lower frequency than an ephemeral cusp formation at a beach. It is assumed that thematic or attribute granularity naturally follows from the formulation of these four classes. The delineation of the spatio-temporal boundaries of a process depends on thresholds of the process. For example, a process of erosion may be a function of exceeding certain thresholds of rainfall duration and average intensity at a point or over an area.

3. Process Parts

Now that the unity of an individual process has been described, the relationships between processes and their parts must be delineated. These relationships are divided into three basic types: temporal parts, spatial parts, and spatio-temporal parts. Their description is essential as they form the basis for query and analysis of processes in implementations of this conceptualization.

Traditionally, the primary tool for describing such relationships is mereology, which describes the relation between part to whole (Simons, 1987). This theory applies in every domain, from ordinary objects such as people, chairs, and mangos, to processes such as frontal systems, information flows, and erosion. It may also be applied to abstract entities such as classes and properties. However, mereology, as defined by its central proponent Simons (1987), has thus far remained three-dimensional in scope (Sider, 2001). As a consequence, spatio-temporal parts are not considered, which are essential for a full explication of processes. Therefore, what follows is an outline of a mereological extension that considers the part-whole nature of four-dimensional processes.

The simplest expression of this part-whole relation is given by ≤, where $x \leq y$ is read as 'x is a part of y'. For example, if a tidal cycle (a spatio-temporal entity) is considered a whole, its sub-processes such as currents and waves are parts of that whole. The relation ≤ includes both the case of proper parthood (<) and equality. The core axioms of mereology define the part-whole relation as reflexive, transitive, and antisymmetric, that is, as a partial ordering (Simons 1987).

This mereological part-of relation behaves differently between 3Dism and 4Dism. 3D things have only spatial parts. 4D entities, in contrast, have both spatial and temporal parts. Processes modify the part-of relation further, where we have spatial, temporal, and spatio-temporal parts. Informally three types of parts may be distinguished for spatial processes: temporal parts, spatial parts, and spatio-temporal parts.

3.1 Temporal Parts

Temporal parts are parts that divide the process along the temporal dimension. Due to the difficulty of visually representing four dimensions, a process is represented here by three dimensions. The three spatial dimensions of a process are collapsed to two spatial dimensions, and the third dimension represents time. The temporal part of a process is displayed in Figure 2 as the gray shaded prism. The dark arrow represents the temporal dimension (*t*) and the rectangular prism represents the spatial and temporal extent of the process.

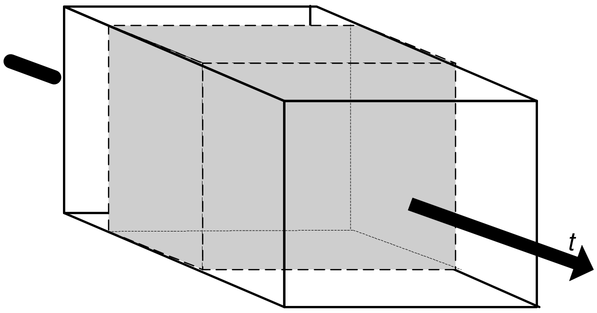


Figure 2. A temporal part of a process

Unfortunately it is difficult to represent schematically all aspects of the relationship, in particular spatial and temporal grain. The temporal part of a process is defined as having:

1. the same spatial extent as the process
2. the same spatial granularity as the process
3. the same temporal granularity as the process
4. a temporal extent that is smaller than the process

For example, a temporal part of a process such as gentrification may be the temporal interval over which rent is increased by a certain amount; a second temporal part of that same process may be the temporal interval over which inhabitants are evicted. Another example is tropical depression Isidore, which from the 14/09/02 – 17/09/02 was tropical depression Ten, 18/09/02 - 19/09/02 (till 4pm) tropical storm Isidore, and from 19/09/02 – 23/09/02 hurricane Isidore (see http://weather.terrapin.com for more detailed hurricane tracking). Tropical depression Ten, tropical storm Isidore, and hurricane Isidore, are all temporal parts of the one process.

3.2 Spatial Parts

Processes do not have spatial parts that endure through time in the metaphysical sense (Sider, 2000). This is a result of the transient nature of spatial parts which may enter and exit a process. Take for example tropical depression Isidore, a spatial part of it on the 14th September at the time it was rated a tropical depression, such as a certain mass of air particles, will no longer be a spatial part on the 21st September when it was rated a hurricane (Figure 3).

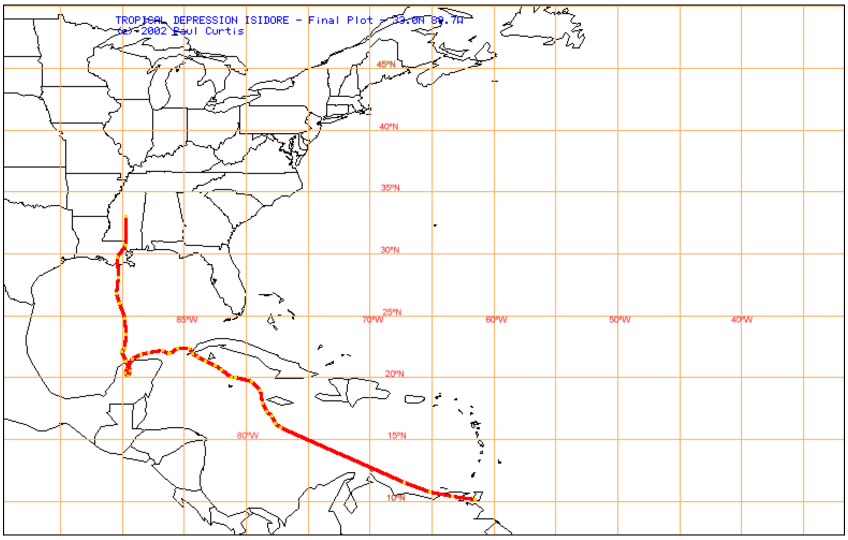


Figure 3. The path of hurricane Isidore (from http://hurricane.terrapin.com/)

Rather, we may only discuss the spatial parts of a process if we consider the process at an instant of time (Figure 4). The spatial part of a process is defined as having:

1. a spatial extent that is smaller the process
2. the same spatial granularity as the process
3. no temporal granularity
4. no temporal extent

Thus, a spatial part exists at an instant of time, but in this conceptualization it is not an entity that endures or perdures through time. For example, spatial parts of urban sprawl at an instant in time may include the inner-city from which sprawlers are moving and the outer suburbs to which they are heading.

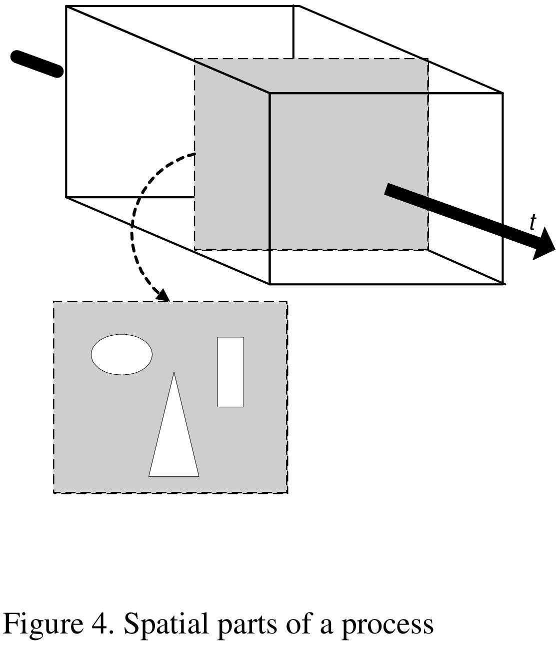


Figure 4. Spatial parts of a process

3.3 Spatio-temporal Parts

The third class of parts, spatio-temporal parts, is the type of part of most interest to the development of this conceptualization. Spatio-temporal parts are termed here sub-processes and are defined as having:

1. a spatial extent that is equal to or less than the process
2. a spatial grain that is equal to or less than the process
3. a temporal grain that is finer than the process
4. a temporal extent that is equal to or finer than the process

In Figure 5, spatio-temporal parts are visually expressed as gray boxes within the larger transparent box. Take for example the process of global warming, which has as its spatio-temporal parts processes such as heat transport and the latent heat flux, which in turn have as their parts processes such as evapotranspiration and thermal radiation from land.

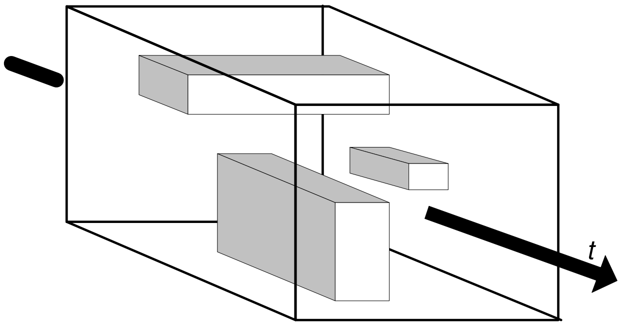


Figure 5. Spatio-temporal parts of a process.

3.4 Process Granularity Trees and Scale

Spatio-temporal parts of processes can be organized into granularity trees or hierarchies in order to clearly define the scale of those spatio-temporal parts. There have been several attempts to classify hierarchies into categories, none of them entirely successful because the categories unavoidably overlap. Thus one can broadly distinguish between "structural" hierarchies, which emphasize the spatial aspect (anatomy, topology) of a system, and "functional" hierarchies, which emphasize process in time. As proposed, structure and function cannot be separated, and represent complementary aspects of an indivisible spatio-temporal process; but it is often convenient to focus attention on one or the other aspect. All hierarchies have a "part within part" character, but this is more easily recognized in "structural" than in "functional" hierarchies (Koestler 1968: 59).

In descriptions of the process class of urban sprawl we can move from the level of granularity defined by processes at the neighborhood level to that level defined by processes observed at the metropolitan statistical area level. Or, in the case of climate processes, we can describe them at the granularity of microclimates or at a level of large-scale phenomena such as the El Niño weather pattern. The spatio-temporal part-whole nature of these processes defines the granularity tree that they compose. Granularity trees, although traditionally defined on spatial objects (Bittner and Smith, 2001), may be extended in the consideration of processes. However, the rules that define the organization of processes into granularity trees are not the same as those for objects. The organization of processes into a granularity tree, also termed hierarchy, is based on a number of ordering principles derived from hierarchy theory (Ahl and Allen, 1996;Allen and Starr, 1982), which were recently investigated by Reitsma and Bittner (2003).

Consider the tidal cycle in the Bay of Fundy as a granularity tree (Figure 6). Note that this is a class of processes that can be applied to any instance of a tidal cycle in the Bay of Fundy. One tidal cycle occurs over almost a 13-hour period, the parts of which may include various currents, such as current A and current B, which in turn may include various waves, such as wave 1 and wave 2, and eddies as their parts or sub-processes. Thus, we can describe processes at varying levels of granularity, the composition of which defines our granularity tree. Note that this example is rather rough and as such is provisional, a consequence of few geographers describing processes hierarchically.

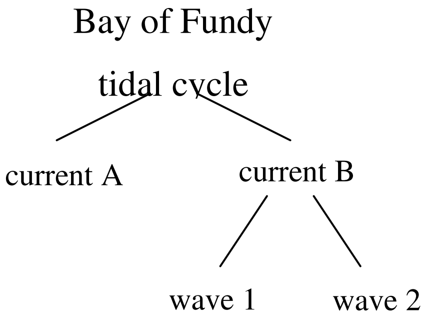


Figure 6. The Bay of Fundy tidal cycle as a granularity tree

In the context of a granularity tree, these processes may be organized into a tree according to their temporal granularity or frequency, beyond the requirements of spatial and temporal extent and spatial granularity. That is, processes of low frequency, which recur over a long duration of time, are higher in the hierarchy than processes of high frequency, which recur over a short duration of time. This relationship between processes at a higher level in the system of granularity and patterns of individual processes at a lower level is repeated down through the system of granularity. For example, the temporal grain or frequency of atmospheric phenomena are typically classified as micro-scale: seconds to minutes, meso-scale: minutes to days, synoptic scale: days to weeks, and macro-scale: weeks and greater (Ahrens, 1991). Likewise in the example above (Figure 6), the tidal cycle in the Bay of Fundy has a lower frequency than currents, which have a lower frequency than their sub-processes of waves.

Thus in the domain of processes, granularity is a spatio-temporal notion. The organization of processes into a tree of granularity must therefore consider both their spatial and temporal extent and their spatial and temporal grain. However particular attention is paid to temporal grain. Furthermore, the levels of processes in a granularity tree are involved in dependence relations, where "different levels can contain completely different types of laws – and these cannot be reduced to each other" (Johansson, 1989: 22).

4. Process Relationships

There are a number of relationships to be considered with regards to processes beyond the description of part-whole relation between processes and sub-processes. Granularity trees (Bittner and Smith, 2001) and other related hierarchical forms such as partonomies

(Tversky, 1990) and those found in hierarchy theory (Ahl and Allen, 1996;Allen and Starr, 1982) do not express measurable characteristics of the relationship between parts and wholes and how they can be organized systematically in a hierarchical fashion. With processes, the added temporal dimension brings with it new questions as to how these parts and wholes relate to one another. For example, is there a causal, controlling, or constraining relationship between parts and wholes? Furthermore, what of the relationships among processes at one level of granularity?

Within this conceptualization of process, intra and inter process relationships are specified within the definition of the process itself. Relationships do no form entities themselves and cannot be discussed independently of processes. Abstractly, all types of interaction can be defined as an exchange of mass or energy; as a consequence they influence the operation of each other in some manner. For example, a landslide process may provide the material for the process of fluvial sediment transport in a river.

All relationships between processes are interactions of some form. This precludes the necessity of developing a typology of spatio-temporal relations that extends the topological relations found in qualitative spatial reasoning research such as the 9-Intersection model (Egenhofer and Herring, 1990) and the RCC (Region Connection Calculus) (Randell *et al.*, 1992). Furthermore, processes can be spatio-temporally co-located, relationships such as overlap have no meaning in this conceptualization unless it is specified as the basis for some kind of interaction, as described in the behavioral rules of each process.

5. On to Ontologies

In sum, the conceptualization developed in this chapter presents a theoretical basis for a model that is founded upon a single primitive, that of process. This elementary thing encapsulates both substance and dynamics, defining its spatio-temporal nature and potential for interaction with other processes within its rules of behavior. From this conceptualization of process an upper-level ontology will be developed in the following Chapter, in a sense, forming the base class from which process models may inherit. This aims at making explicit the conceptualization, minimizing ambiguity, and therefore making clear any assumptions lying therein.

CHAPTER FOUR

Flux Ontology

A theory of categories…draws the line around the thinkable…such a line is not immutable, as with science and common sense, it changes…as it has in the past.
(Johansson, 1989: 5)

1. Introduction

This chapter attempts to address an ontological gap by developing an upper level ontology for modeling spatial processes. The ontology to be developed in this chapter is based on the conceptualization expressed in Chapter 3. Rather than considering the spatial shadows of these dynamic phenomena, such as the patterns produced by processes, the heart of spatial processes is aimed at, as is expressed in modeling such processes. Hence, the ontology's domain is geographic processes, and its aim is to capture the abstract nature of these processes for the purpose of modeling. The advantage of developing an upper level ontology of geographic processes is that it clearly communicates the primitives of the modeling theory in this dissertation and provides the basis for developing a modeling methodology. Furthermore, at such an abstract level we are able to talk of processes in both physical and human domains of research, a divide that is rarely crossed.

Some of the design criteria the ontology attempts to meet includes extensibility, minimal ontological commitment, and minimal encoding bias (Uschold and Gruninger, 1996) The upper-level ontology aims to capture a shared vocabulary in order to be extendable.

A minimal ontological commitment involves making as few claims as possible about the modeled world; that is, it endeavors to commit to the most basic terms and thereby broaden the net capturing potential domains. The concept of minimal encoding bias describes the independence of the ontology on a particular symbol-level encoding or language. Furthermore, the specification must be consistent, complete, and match the problem domain.

The following parts of this chapter are structured as follows: Section 2 provides an overview of relevant ontology languages and tools, selecting the most appropriate for the purpose of modeling geographic processes. Section 3 presents the ontology, entitled flux. Section 4 concludes the chapter and Part I.

2. Ontology Languages and Tools

The language used to express the ontology has a direct impact on what can be said, hence the importance of careful selection. If we make the mistake of using or relying on natural language as our window onto reality, then we will be stuck with its semantic vagaries and variations among cultures. The primary criteria for selection of an ontology language are the level of expressivity, a formal specification of the language's semantics, and the potential for implementation. The level of expressivity needed includes the ability to express:

- Classes - defining the processes in the ontology
- Predicates - defining both the relationships between processes and their properties

- Rules - defining the thresholds of process creation, death, and change
- Methods - defining the behavior of the processes

An inadequate specification of the semantics will lead to inconsistent interpretations and uses. As expressed by Winter and Nittel:

> "Specifications describe the *what* of pieces of a task ('what are the actors?', 'what are their relations?', 'what kind of actions have to be taken?'), not the *how* ('how do the actions have to be executed?"). Hereby, specifications do not only name the actions but also describe the restrictions, result and meaning of these actions, i.e. the semantic aspects of an action."(2003: 724).

The third criterion, potential for implementation, is a consequence of foresight regarding the need to translate the specification into executable Java code.

What follows is an overview and evaluation of the most pertinent linguistic tools for expressing the ontology of processes. This discussion leads to a language selection based on the above criterion and the requirements of the theory and conceptualization expressed in Chapters 2 and 3.

2.1 Ontology Editors

Ontology editors are the obvious first choice in the development of ontologies as they provide the support for the rapid development of ontologies. There are a range of free editors available, such as the online ontology editor Ontolingua, developed by the Stanford Knowledge Systems Laboratory [1] (Farquhar *et al.*, 1996), and Protégé-2000. These two examples also have the capability of exporting the ontology into various

[1] http://www.ksl.stanford.edu/software/ontolingua

target languages to be directly inserted into external software; in particular Protégé has a Java plugin (beta version) that translates the ontology into Java classes, as well as externally developed extensions such as OntoJava (See: http://www.i-u.de/schools/eberhart/ontojava). Furthermore, Ontolingua is able to declare functions, which has advantages for geographic information (Mota *et al.*, 2002).

However, not all aspects of the ontology can be expressed in these ontology editors; in particular, methods describing the behavior of the processes cannot be formulated. Anachronistically, it was also found that such editors are very restrictive in terms of having direct control over the mapping to Java code for implementation. They do not allow one to "get under the hood" and use the ontology to map to an implementation or extend the ontology model, nor can all aspects of the model ontology be converted to Java.

2.2 UML

The Unified Modeling Language (UML) was also explored for its potential in defining the base ontology (Rumbaugh *et al.*, 1999). UML does not have formal semantics, which results in multiple interpretations and potential implementations (Winter and Nittel, 2003). For this reason, spatio-temporal extensions to UML, such as STUML (spatio-temporal UML), are not useful for the purpose of this research either (Price *et al.*, 1999). Prior to the release of UML 2, UML was not directly executable beyond the creation of the class and property structure of a model. The recent development of Action Semantics in UML2 provides the potential for fully executable UML. However, it does not have a notation or syntax, it is a purely semantic standard and therefore its use is dependent on the implementation (Rumpe, 2002;Sunyé *et al.*, 2001).

2.3 PSL

The Process Specification Language (PSL) is a formalized language that has been developed for the interoperability of business and manufacturing processes, exchanging process information among systems (Grüninger, 2003). The PSL defines a process as a collection of activity roles, a process specification therefore being a set of activity role specifications (Menzel and Grüninger, 2001). The PSL presents useful constructs in a formal language, yet these constructs are defined on an absolute notion of time. The model theory requires the linear ordering of processes to be pre-specified[2]. In addition, the PSL maintains the process - object dichotomy that this dissertation is aiming to avoid with Activities and Objects.

2.4 OWL and extensions

OWL (Web Ontology Language) is a language used for describing information that needs to be processed by computer where its formal semantics allows for meaning to be expressed[3]. It is specifically designed for material to be represented on the World Wide Web, and forms the basis of the vision of the Semantic Web (Berners-Lee *et al.*, 2001). However, it has much wider potential than the web as it has formal semantics that are

[2] http://www.nist.gov/psl

[3] http://www.w3.org/2001/sw/WebOnt

serialized in XML, a burgeoning developer and user community, and is recommend for use as a basis for ontology specification of information that is not web based. Jena 2.0, a Java library of classes for dealing with OWL ontologies, has also been developed, which allows for the development of computer applications that utilize ontologies[4].

2.4.1 OWL-S

OWL-S (Web Ontology Languages for Services), formerly DAML-S, provides a high level ontology that is intended for modeling web services[5]. This extension to OWL is based on research in workflow management and programming languages and is similar in nature to the PSL (Grüninger, 2003). These web services are described as having "processes" as one part of their upper level ontology, which are described in a "process model". It expresses many useful notions regarding processes, time, and conditional constructions, such as if-then-else statements. As with PSL, the key class (Process) and its related classes and properties are based on an absolute notion of time, as expressed in the DAML-Time ontology, which is not useful for the development of the modeling ontology based on the conceptualization presented in Chapter 2 (Pan and Hobbs, 2004). Furthermore, OWL-S also incorporates objects and processes, where processes involve agents, such as clients and servers.

2.4.2 SWRL

One of the primary deficiencies of OWL for modeling is the inability to express thresholds in the form of rules. However, recently there have been developments of a

[4] http://jena.sourceforge.net/

[5] http://www.daml.org/services/

rule standard in the form of SWRL (Semantic Web Rule Language), which extends OWL to the expressivity of full first order logic[6]. SWRL is based on RuleML (Rule Markup Language), a standardized XML syntax catering for the expression of rules and allows them to be encoded in a standard yet informally semantic way. The RuleML does cater to reaction rules in the form of a set of premises resulting in an action, however its translation to SWRL has thus far only been a restricted part of this abstract rule type, the Derivation rules, which assert a conclusion when certain conditions hold.

3. Flux Ontology

Following the discussion above, OWL was selected as the ontology language to be used in the following explication of an upper level process ontology. The upper level ontology namespace or URI is: www.glue.umd.edu/~femke/ProcessModel/flux#. The namespace name is: "flux", and it can be found in Appendix A .

3.1 Classes

The single primitive class, Process, is expressed as follows:

```
<owl:Class rdf:ID="Process"/>
```

An example of a domain specific subclass of Process is expressed as:

```
<flux:Process rdf:ID ="Runoff">
      <rdfs:label>Runoff</rdfs:label>
</flux:Process>
```

[6] http://www.w3.org/Submission/SWRL/

The subclass relationship is transitive, defining a class hierarchy of processes and sub-processes.

3.2 Properties

Each instance of a Process has the minimal set of properties summarized in Table 1 below. All of these properties are owl:Datatype properties, mapping an object to a datatype.

Property	**OWL Label**	**Domain**	**Range**
Spatial X Extent	spatialXExtent	Process	Integer
Spatial Y Extent	spatialYExtent	Process	Integer
Spatial Z Extent	spatialZExtent	Process	Integer
Spatial Grain	spatialGrain	Process	Integer
Temporal Extent	temporalExtent	Process	Integer
Temporal Grain	temporalGrain	Process	Integer
X1	x1	Process	Integer
X2	x2	Process	Integer
Y1	y1	Process	Integer
Y2	y2	Process	Integer
Z1	z1	Process	Integer
Z2	z2	Process	Integer
Value	value	Process	Integer

Table 1. Properties of a Process

These properties are defined in the flux ontology in the following manner:

```
<owl:DatatypeProperty rdf:ID="spatialExtent">
      <rdfs:label>spatialExtent</rdfs:label>
      <rdfs:domain rdf:resource="#Process"/>
      <rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>
```

Spatial X Extent and Spatial Y Extent: The Spatial X Extent and Spatial Y Extent of a process define the spatial boundaries of the Process in granular units defined by the model. These properties are properties of a subclass of Process.

Spatial Grain: The Spatial Grain of a Process defines the spatial distance a Process traverses when it operates. It is in granular units defined by the model. Spatial Grain may be relationally defined as a rule (see discussion of rules below) or a constant. Spatial Grain is a property of a subclass of Process.

Temporal Extent: The Temporal Extent of a Process is the duration over which the Process operates. Temporal Extent is a property of a subclass of Process.

Temporal Grain: The Temporal Grain of a Process is the temporal duration of an instance of Process when it operates. Temporal Grain may be relationally defined as a rule (see discussion of rules below) or be a constant. Temporal Grain is a property of a subclass of Process.

X1, X2, Y1, Y2, Z1, Z2: The X1, X2, Y1, Y2, Z1, Z2 properties of a Process define the spatial location or coordinates of the Process at any instant of time. They can only be defined on individuals of a domain Processes, not on subclasses of a Process.

Value: The Value of a Process is a number that expresses some attribute of the Process used to define its behavior. It is only defined on a Process individual.

An example of a subclass of Process:

```
<owl:Class rdf:ID ="Runoff">
      <rdfs:label>Runoff</rdfs:label>
      <flux:spatialXExtent>100</flux:spatialXExtent>
      <flux:spatialYExtent>100</flux:spatialYExtent>
      <flux:spatialGrain>1</flux:spatialGrain>
      <flux:temporalExtent>20</flux:temporalExtent>
      <flux:temporalGrain>1</flux:temporalGrain>
      <rdfs:subClassOf rdf:resource="&flux;Process" />
</owl:Class >
```

An example of a Process individual:

```
<Runoff rdf:ID="runoff432">
      <flux:x1>20</flux:x1>
      <flux:x2>21</flux:x2>
      <flux:y1>4</flux:y1>
      <flux:y2>5</flux:y2>
      <flux:z1>2</flux:z1>
      <flux:z2>1</flux:z2>
      <flux:value>10</flux:value>
</Runoff>
```

3.3 Rules

The rules that define the thresholds of change or process behavior in the process model are expressed in SWRL. These rules define relationships among processes.

3.3.1 Rule Syntax and Semantics Overview

Rules are expressed in the form of an implication between an antecedent (body) and consequent (head). Informally, the intended meaning of an implication is read as: if the conditions specified in the antecedent hold, then the conditions specified in the consequent must also hold. Both the antecedent and consequent consist of conjunctions of atoms, which may refer to individuals, data literals, or variables.

For example, the meaning of the following rule is: if variable x has a wind speed greater than 65 knots, and x is located in a place (represented by variable y) called the “Western Pacific”, then x has the label “Typhoon”.

```
<swrl:Variable rdf:ID="x"/>
<swrl:Variable rdf:ID="y"/>

<owl:Impl>
<owl:body rdf:parseType="Collection">
      <swrl:dataValuedPropertyAtom>
                  <swrl:propertyPredicate
                  rdf:resource="#hasKnotsGreaterThan"/>
                  <swrl:argument1 rdf:resource="#x"/>
                  <swrl:argument2>65</swrl:argument2>
      </swrl:dataValuedPropertyAtom>
      <swrl:individualPropertyAtom>
                  <swrl:propertyPredicate
                  rdf:resource="#isLocatedIn"/>
                  <swrl:argument1 rdf:resource="#x"/>
            <swrl:argument1 rdf:resource="#y"/>
      </swrl:individualPropertyAtom>
      <swrl:individualPropertyAtom>
                  <swrl:propertyPredicate
                  rdf:resource="&rdfs;label"/>
                  <swrl:argument1 rdf:resource="#y"/>
            <swrl:argument1 rdf:resource="#Western Pacific"/>
      </swrl:individualPropertyAtom>
</owl:body>
<owl:head>
      <swrl:individualPropertyAtom>
                  <swrl:propertyPredicate
                  rdf:resource="&rdfs;label"/>
                  <swrl:argument1 rdf:resource="#x"/>
            <swrl:argument1 rdf:resource="#Typhoon"/>
</swrl:individualPropertyAtom>
</owl:head>
</owl:Impl>
```

3.3.2 Rule Classes

There are three types of rules expressed in the flux ontology:

- Create Process Rule (CreateProcessRule)
- Change Process Rule (ChangeProcessRule)
- Destroy Process Rule (DestroyProcessRule)

The CreateProcessRule defines the threshold for creating a new process in the model. The ChangeProcessRule specifies when a process individual changes one or more of its properties. The DestroyProcessRule identifies when the process individual will be removed from the model. Only the abstract classes are defined in the flux ontology, their full expression can only be defined in a domain ontology. For example:

```
<owl:Impl rdf:ID="CreateProcessRule"/>
```

In the following domain example, the threshold for the creation of a runoff Process is defined, where if Precipitation has value *a*, and *a* is greater than 2, a new Process individual is created. The creation of the process is defined by a method, which is described in Section 3.4 below.

```
<swrl:Variable rdf:ID="a"/>
<swrl:Variable rdf:ID="p"/>
<owl:Impl rdf:ID ="createRunoffRule">
      <rdfs:label>createRunoffRule</rdfs:label>
      <rdf:type rdf:resource="&flux;CreateProcessRule"/>
      <owl:body rdf:parseType="Collection">
            <swrl:individualPropertyAtom>
                  <swrl:propertyPredicate rdf:resource="&flux;value"/>
                  <swrl:argument1 rdf:resource="#Precipitation"/>
                  <swrl:argument2>#a</swrl:argument2>
            </swrl:individualPropertyAtom>
            <swrl:dataValuedPropertyAtom>
                  <swrl:propertyPredicate
                        rdf:resource="&sumo;greaterThan"/>
                  <swrl:argument1 rdf:resource="#a"/>
                  <swrl:argument2>2</swrl:argument2>
            </swrl:dataValuedPropertyAtom>
      </owl:body>
      <owl:head>
            <!–
                  The rest of this example is continued below in
                  Section 3.4 on Methods
            -->
      </owl:head>
</owl:Impl>
```

3.4 Methods

If the threshold defined in the rule is passed, a method is used to implement the behavior of the process. A separate methods ontology was developed to be used in conjunction with the flux ontology, in order to form the basis for its implementation. The URI of the methods ontology is: www.glue.umd.edu/~femke/ProcessModel/methods#; it has a namespace of: “methods” and can be found in Appendix B.

3.4.1 Method Syntax and Semantics Overview

OWL does not yet have the capacity to express methods. Following initial exploration into developing a language extension for methods, an alternative approach was selected based on the concept of built-ins developed in SWRL as it fell better within the time constraints of the dissertation and the experience of its author. SWRL built-ins have been developed for future extensions of the language, and are essentially a call out to an external method or program that returns information required to evaluate the SWRL statement[7].

3.4.2 Method Classes and Properties

The SWRL Syntax expresses the head and body of an argument as a collection of atoms, such as the IndividualPropertyAtom and DatavaluedPropertyAtom. Within the method ontology a MethodAtom class has been declared, which is a subclass of the generic SWRL Atom[8]. Within a declared MethodAtom, taking the place of SWRL property

[7] http://www.w3.org/Submission/SWRL/#8

[8] http://www.w3.org/2003/11/swrl#

predicates, are two property classes: MethodObjectProperty and MethodDatatypeProperty. These property classes are subclasses of MethodProperty, a subclass of rdf:Property. A method of type MethodObjectProperty returns an object from the operation of the method. A method of type MethodDatatypeProperty returns a datatype from the operation of the method.

There are three basic method types defined in the flux ontology:

- createProcess - which results in the creation of a new process instance
- destroyProcess – which results in the destruction of a process instance
- changeProcess – which results in the change of a process instance, such as its movement or change in property value

An example of a method subclass in the flux ontology is:

```
<method:MethodObjectProperty rdf:ID="changeProcess">
      <rdfs:label>changeProcess</rdfs:label>
      <rdfs:domain rdf:resource="#ProcessModel"/>
      <rdfs:range rdf:resource="#Process"/>
</method:MethodObjectProperty>
```

An example of a method individual, which is continued from the first part presented in Section 3.3.2:

```
      <owl:head>
         <method:MethodAtom>
             <swrl:propertyPredicate rdf:resource="#newRunoffProcess"/>
                  <swrl:argument1 rdf:resource="#RunoffModel"/>
                  <swrl:argument2 rdf:resource="#p"/>
         </method:MethodAtom> >
      </owl:head>
</owl:Thing>
```

4. Concluding Part I

The above ontology, founded in the conceptualization of Chapter 3, provides the groundwork for the methodology to be developed in Part II. As will become evident in forthcoming chapters, there will be changes to the ontology in order to accommodate the things needed in a process model and the requirements of a domain ontology.

An interesting conundrum arises in how to deal with future possibilities. That is, we are stuck with current things defined in the ontology, but what if new things emerge? This involves questions of structural evolution of a system rather than simply system dynamics. It seems that it is currently impossible to develop new classes of things as they arise in the operation of a process model beyond the specification of a metamodel that captures a wider range of rules, which delays rather than solves the problem.

CHAPTER FIVE

Methodological Review

Time is a means of relating distinguishable changes to each other.
(Lippincott, 1999: 246)

1. Introduction

There are many different methodologies for modeling dynamic geographic phenomena, such as mathematical models or agent-based models, all of which are implemented within a computational environment. Any of these approaches to modeling processes assume a certain conceptualization of the entities they are concerned with, whether it is explicitly formalized within an ontology or implicit in the underlying assumptions of the model. This chapter considers methods of modeling geographic processes in the light of their conceptual underpinnings. Its objective is to review the methodology and present an argument for an alternative modeling conceptualization that is founded in the theory developed in Part I. A consequent modeling conceptualization follows in Chapter 6, and its implementation is described in Chapter 7.

The chapter is structured as follows; Section 2 considers how geographic "things" have been modeled thus far, relating these methods back to the theory discussed in Part I. The primary methodologies that will be discussed are Geographic Information Systems (GIS), Cellular Automata (CA), Agent Based Modeling (ABM), and Equation Based Modeling (EBM), which capture the dominant (if not all) approaches to modeling dynamic geographic phenomena. Additionally, research on database management systems for spatio-temporal data is also reviewed. Section 3 follows with a discussion of the advantages and significance of the methodology developed in this dissertation, which takes process as its primitive construct. Section 4 concludes the chapter.

But first, two caveats.   In what follows, reference to an object in terms of object-oriented implementation will be clearly stated in order to avoid confusion with the use of the term object to represent a static primitive. And, although seemingly inane, it is assumed here that the objective of all process models is to model processes, a point which will be returned to towards the end of this chapter.

2. Modeling Geographic *things*

The following review of modeling approaches in geography centers on their treatment of space, time, change, and the primitive things that are modeled. These four abstractions form the focusing lens through which three methodological clusters are viewed, namely Geographic Information Systems (GIS), computational simulation in the form of Cellular Automata (CA) and Agent Based Modeling (ABM), and Equation Based Modeling (EBM). In addition, database management systems for spatio-temporal data and the practical application of ontologies is discussed.

2.1 GIS and Extensions

GIS are the most prominent compendium tool for research involving geographic data. They are tools for the import, manipulation, management, analysis, display, and export of spatial data. The typical data model primitives available to the user are points, lines, polygons, and pixels. These primitives are used to represent static geographic data, in 2, 2.5, and 3-dimensions. Points, lines, and polygons, are typically referred to as vector data models, and pixels compose the raster data model, collections of which form the object view of the world. Both raster and vector representations have an absolute datum, that is, a coordinate system. Regardless of whether one is space filling or not, both treat space in the same manner. This is evident in their integration in a unified data model by Cova and Goodchild (2002), an amalgamation that captures the advantages of both raster and vector representations.

From their earliest days GIS were not designed or pre-conceptualized as dynamic modeling tools. There are two main approaches to extending GIS for representing dynamic phenomena: temporally extending GIS, and coupling GIS to environmental models. Temporal extensions to GIS involve either snapshots, where each layer represents an instance in time, or amendments vectors, where each entity is associated with a list that contains information regarding each change in the entity (Langran, 1993;Peuquet, 1994). The “snapshot” data model, one of the earliest representations of time in GIS, organizes space over time, where each raster layer is used to represent a state of the world at a point in time (Wachowicz, 1999). A collection of those spatio-temporal snapshots is used to represent a 4-D space-time cube, where at each time step there is a tuple of object id, space, and time (Peuquet, 2001). This snapshot practice is conceptually intuitive, convenient, and easily adapts to available data sources such as satellite imagery, hence it remains prevalent due to its simplicity (e.g. Chen and Jiang, 2000). Problems of large-scale data redundancy, where phenomena do not change everywhere at all times, produced an alternative, the base-state with amendment model. This model updates states from the initially complete snapshot for only those objects that undergo change (Langran, 1993). For both these approaches, change is interpolated between consecutive system states, whether it be between system states or object states.

Incorporating time into the raster and vector data models is seen as the obvious solution to representing dynamics. However, as argued by Peuquet (1994), time and space exhibit important differences that do not comply with the neat addition of dimensions. Recognition that "simply extending a spatial data model to include temporal data, or vice versa, will result in inflexible and inefficient representations for space-time data" has produced a slew of spatio-temporal substitutes (Peuquet, 2001: 15). Alternatively, time can be represented by space, as has been developed in time geography, which implements Hägerstrand's classic model of temporal phenomena (Hägerstrand, 1967). Computational implementations of time geography represent the potential path of an individual as a spatial extent which changes over time as the individual moves through space over time (Miller, 2003a).

For linking GIS and dynamic models, particularly environmental models, there have been a variety of coupling solutions ranging from loose, in terms of file import and export, to tight coupling, that is, integrated environments (Bernard and Kruger, 2000). For example, Feng and Sorokine (2001) mapped hydrological model classes to the OpenGIS consortium's abstract specification[9], and Pullar (2003) developed an integrative development language called MapScript. However, there are many limitations to coupling of GIS to physical models, which have been well documented (Waters, 2002). As Kemp (1997: 232) notes, "GIS manages static and discrete data while environmental models deal with dynamic and continuous phenomena…In order to fully integrate the two we need to add dynamics and continuity to our understanding of spatial data and spatial interaction and functionality to the environmental models".

Object-orientation has been hailed as a solution to integrating environmental models and GIS, or as a new basis for representing environmental processes (Bian, 2000;Raper and Livingstone, 1995;Wachowicz, 1999). The development of object-orientation programming languages has engendered much research in object-oriented GIS, modeling, and databases. Advantages of the object-oriented approach include its conceptual consonance, software independence, and that the identified entities structure the representation rather than the geometry structuring the representation, in contrast to GIS which depends on geometrical primitives (Bian, 2000;Raper and Livingstone, 1995). Hence new modeling techniques, both conceptual and implemented, are predominantly being developed within this object-oriented paradigm (Borges *et al.*, 2001;Frihida *et al.*, 2002;Hamre *et al.*, 1997;Raper and Livingstone, 1995).

[9] http://www.opengis.org

Object orientated approaches typically handle time by time-stamping objects or their attributes (Stefanakis, 2003), which neatly parallels the 3Dism approach to handling the persistence of objects over time as discussed in Chapter 2. Typologies of modeling change have also been developed, analogous to the theoretical classifications introduced in Chapter 2, where change is represented as a new state with a new time stamp (for example Yuan, 1996;Yuan, 2001). These developments draw apart the temporal, spatial, and attribute dimensions, reducing change to a variety of distinct forms. For capturing change in spatial objects, various temporal interpolation methods have been proposed for determining geometric changes of spatial objects (Zhang and Hunter, 2000).

Such advances have lead to what has been termed dynamic GIS, which introduce environmental modeling techniques within a GIS (De Vasconcelos *et al.*, 2002). Here the lines between the traditional fields of GIS and automata based simulation are rapidly blurring, with both the increasing integration of GIS data structures into computational simulation tools and the converse import of simulation tools into a GIS environment. For example, De Vasconcelos *et al.* (2002) present a dynamic GIS which is based on a *geounit*, a CA like data structure which extends that simple formalism to any form of spatial structure and is combined with scheduled and event based actions. A further example of the integration of computational simulation and GIS is the development of CA within a GIS, for example, van Deursen developed a spatially distributed hydrological model in PCRaster (an open source GIS developed at the University of

Utrecht), which is essentially a CA (1995). Similarly, ABM are also being coupled to GIS for importing spatial data (Gimblett, 2002).

2.2 CA and ABM

Discrete computational models for spatial processes typically take the form of Cellular Automata (CA) or Agent-based Models (ABM). The goal of both CA and ABM is to model emergent phenomena that are not self-evident in the capabilities of individual units. Such emergence is found in patterns of observables. They both model the same notion of underlying absolute space and utilize the same types of time, falling into what Zeigler terms discrete time or discrete event systems, depending on the modeling approach taken (Zeigler *et al.*, 2000). In brief, the primary distinction between CA and ABM is the conceptual primitive used to represent phenomena. In CA, this primitive is a static cell or pixel, a collection of which composes a layer of cells. Its dynamics involves each cell transferring information to its neighboring cells. An ABM, in contrast, is composed of distinguishable objects, the same geometric primitives of point, line or polygon data models found in GIS. Furthermore, an agent has the added advantage of being mobile.

Other approaches to simulation, such as qualitative simulation are not considered in depth here. However, it is of relevance to note that work in qualitative simulation maintain the distinction between object and process, where the application of processes to objects results in future system states. For example, Simmons (1983) models processes in geologic interpretation, and their application to observable states.

2.2.1 CA

CA are a modeling framework for spatially continuous phenomena (Langton, 1986), such as landscape processes or urban sprawl (Box, 2002;Haff, 2001;Silva and Clarke, 2002). They are simple models used to represent the diffusion of things such as matter, information, or energy, over a spatial structure. In its most simple form, a CA is composed of a uniformly tessellated surface (typically a grid) whose cells may exist in a finite number of discrete states. As such, CA can be considered a dynamic extension to raster GIS (Bian, 2000). Each cell has an identically sized neighborhood consisting of nearby cells, and a rule set defining how each cell changes based on the state of its neighborhood. These changes can be either a function of relative or absolute models of time, absolute time being where the scheduled tick of the model clock defines the change, and relative time expressed as a cascading process of event-based changes from one cell to the next. With these component parts, the model is initiated and run where each cell in the CA checks its neighborhood and changes its state based on the rules defining its behavior. Despite the simplicity of construction, the dynamics of a CA model can produce complex results. For example, O'Sullivan measures change as the record of the time-series evolution of a measure of spatial pattern (2001).

However, CA are limited when it comes to modeling dynamic spatial phenomena. The most important limitation is that the structure of the tessellation is typically static. Although, there has been some promising experimentation with mutable CA in urban modeling (Semboloni, 2000), and self-modifying rules to capture non-linear behavior have been used (Silva and Clarke, 2002). Yet there remains little scope for feedback and consequent self-organization of the cellular structure.

2.2.2 ABM

Agent-based modeling (ABM), synonymous with individual based modeling in ecology (Bian, 2000), is a simulation methodology focused on mobile individuals and their interaction. It is based on the development of Multi-Agent Systems (MAS), which were created in the field of distributed artificial intelligence, a sub-field of artificial intelligence (Gilbert and Terna, 2000). In a sense, ABMs are the dynamic object (vector) counterpart of the dynamic field (raster) representation and implementation of CAs. The primitive in ABMs, the agent, can conceivably be used to represent anything of interest to the modeler. Agents are typically used to represent human actors of some form, which interact with each other and/or with their environment (Brown *et al.*, forthcoming;Epstein and Axtell, 1996). However, they can also be used to represent physical environments such as those represented by GIS layers (Box, 2002), or objectives such as crises management or prevention and control (Weber, 1998).

The observables or attributes of an agent (including spatial location) are measurable characteristics of the agent that change over time (Van Dyke Parunak *et al.*, 1998). These observables describe the state of the system at any one time and are the primary output of an ABM. ABMs develop histories of system states, where, as with temporal extensions to GIS, change is handled by storing the system state at each time or by storing vectors of events for each agent. The focus of ABM is to understand the emergent outcome of each model, where emergent “denotes the stable macroscopic patterns arising from the local interactions of agents”(Epstein and Axtell, 1996: 35). In terms of spatial ABMs, this is the spatial pattern of observables (for example Parker and Meretsky, 2004).

2.3 EBM

An Equation Based Model (EBM), in its simplest form, is a function that can be applied to some observable. These observables are measurable characteristics of interest that may change over time. EBM are based on a set of equations that express relationships among observables, their evaluation producing the evolution of the observables over space and time. In contrast to CA and ABM, future states are not directly specified, rather a derivative function is used to specify the rate of change of the state variables (Zeigler *et al.*, 2000). As further explained by Zeigler, “[a]t any particular time instant on the time axis, given a state and an input value, we only know the rate of change of the state. From this information, the state at any point in the future has to be computed” (2000: 49). However, in terms of simulation, there is evidence to suggest that the same results can be gained by either the computational approach, such as ABM, or the EBM (Brown *et al.*, forthcoming;Van Dyke Parunak *et al.*, 1998).

EBMs can be developed in a range of spatial dimensions. For example, global climate change models may be one horizontal (varying with latitude) or vertical dimension (varying with altitude), such as energy balance models or radiative convective models respectively. Alternatively they may be created as two dimensional statistical dynamic models, varying with both latitude and altitude. Or full three dimensional models, within both the atmosphere and the ocean (Henderson-Sellers and McGuffie, 1987), may be formed to represent the global climate system. In terms of two or more dimensional spatial models, EBMs are composed of sets of linked partial differential equations.

Typically EBMs are developed with spatially continuous data, as might be represented in two dimensions with a tessellated surface such as the raster data model. In this case the spatial modeling primitives are pixels, which may have one or multiple attributes, and where each layer is associated with an instant of time. As such, these representations suffer from the same problems in representing the dynamics of the model as GIS, as expressed in Section 2.1. The vector field, another form of tessellated surface, represents both direction and magnitude at each instant of time; for example, wind and flow fields. This comes much closer to the data model represented here. However, vector fields are utilized to represent the movement of some mass or energy as opposed to the processes that are involved in that movement, which incorporate a set of associated rules and attributes as expressed in the *nen* data model introduced in Part I.

2.4 Databases and Query Languages

Outside of the geography community there has been work in the development of spatio-temporal Database Management Systems (DBMS), where spatial formalisms have been temporally extended (Abraham and Roddick, 1999). Traditionally spatio-temporal DBMS development involved extensions of the relational data model (Peuquet 2001), yet of late there has been a transition from relational data models to object models (Griffiths *et al.*, 2001). However, there are as yet few examples of truly spatio-temporal database systems, and "most lack support for changes to aspatial data" (Griffiths *et al.* 2001: 11). One developing example is the Tripod project, which seeks to develop a complete spatio-temporal database system that supports the storage, management and querying of entities that change over time through the notion of a history (Griffiths *et al.*, 2001).

The focus of spatio-temporal data modeling is on objects and their relationships, such as their spatio-temporally extended entity-relation model (STER) (Tryfona and Jensen, 1998). These objects and relationships are temporally extended and have histories that define their changes, where either the object or the attribute is time stamped (Huang and Claramunt, 2002). The movement or history of spatial objects are usually stored as trajectory vectors in 3D space (Peuquet, 2002). For example, MOD (Moving Objects Database) systems are designed for applications such as tracking delivery vans, taxicabs, or military vehicles (Peuquet, 2002).

In terms of change, there are two types typically evident in a database: schema evolution and data evolution (Libourel, 2001). For data evolution, most spatio-temporal database modeling emphasizes the snapshot view, where change can be interpolated between time slices of system states or object states (Erwig *et al.*, 1999). These changes have also been used in constraining the evolution of objects represented in a database, defining permissible and prohibited evolutions in the database where evolution or change is modeled as a temporal relationship between two states (Claramunt and Parent, 2003). More recently Mountrakis *et al.* (2002) developed a change-oriented data model for the storage and querying of spatio-temporal information. Their approach allows them to store the change between time slices that represent objects such as buildings or cadastres, and query those changes at multiple levels of abstraction.

The integration of space and time in databases largely deals with geometries changing over time, that is, change in the three spatial object abstractions: point, line, and polygon. As a consequence of developments in DBMS, the types of allowable queries thus focus on geometric or attribute change (where attribute includes location), For example, how did the political boundaries of Europe change over time? Or, when did the last 100 year flood occur which exceeded by 10% the average spatial extent of 100 year floods? With the three dimensions of space, time, and attribute, a spatio-temporal query is expressed where one is fixed, the other controlled for, and the third to be measured (Frihida *et al.*, 2002).

In order to express these queries for spatio-temporal databases, spatio-temporal query languages have been developed. These query languages have predominantly been developed through extensions of SQL (Structured Query Language) for relational databases, such as STQL (Spatio-Temporal Query Language) (Erwig and Schneider, 1999), or extensions of OQL (Object Query Language) for object oriented databases that are based on the ODBMG standard (Object Database Management Group), such as STOQL (Spatio-Temporal Object Query Language) (Huang and Claramunt, 2002) and Tripod-OQL (Griffiths *et al.*, 2001).

2.5 Ontology Based Modeling

Thus far, the author is not aware of any research on utilizing ontologies for modeling. However, there has been research into ontology driven GIS. For example, mapping ontologies to class structures as a basis for implementation has been operationalized, such as the work by Tryfona and Pfoser (2001) who automatically generate object classes, such as city or river, from ontologies. Similarly Fonseca et al. generate classes from ontologies for semantic interoperability (2002).

There is also a growing body of work directed to converting semantic web languages to running code. For example, Kalyanpur *et al.* describe a general approach for mapping OWL classes to Java classes (Kalyanpur *et al.*, 2004), and OntoJava automatically converts RDF Schema and RuleML sources into a set of Java classes (Eberhart, 2002).

3. Advantages and Significance of Process

As expressed in the introduction, it is assumed that geographic process models do just that, model geographic processes. However, it is argued here that this is precisely not what the modeling methods discussed above in Section 2 do. In what follows, four arguments are presented for a methodology that takes process as its primitive, namely, that processes should be modeled rather than future system states, the need for storage and query of process information, the potential for process analysis and uncovering causality within models, and the utility of the process construct as the basis for interoperability and greater query and analysis efficiency. These arguments are not predicated on what cannot be done, rather, on what is not being done in dominant approaches to modeling geographic processes due to the focus on modeling future system states.

3.1 Modeling Processes

Every knowledge base or knowledge-based system is committed to some conceptualization, either explicitly or implicitly (Gruber, 1993). Similarly, modeling methods are also constrained by an explicit, or more commonly, implicit conceptualization. The modeling approaches discussed above are committed to conceptualizations that focus on simulating future system states rather than processes. As expressed by Claramunt *et al.*, "[c]urrent spatio-temporal models are oriented toward the representation of the evolution of spatial entities. However, none of them provides basic constructs to specify the underlying knowledge describing processes occurring in the real-world"(1997: 16).

In current approaches to modeling, processes are specified by the rules or functions that translate one state of the system to another. Between state time slices, amendment vectors, CA state changes, and agent movements, the nature of the process, although implicit in the behavioral rules or mathematical functions of model, is not explicitly modeled or recorded. As expressed by Fotheringham, "inference plays a key role in any quantitative study. In any study, data are collected to *infer* something about an underlying process or situation" (Fotheringham *et al.*, 2000: 184 author's italics). In stating that process is not modeled, what is meant is that the modeling system is not focused on representing the spatio-temporal operation of processes. While processes are specified as rules or equations in traditional approaches, there are no data models or data structures that represent process dynamics, regardless of whether they can be derived by reevaluating the rules between time slices.

In terms of the methods described above in Section 2, all of them embody this problem and have added problems in representing dynamic phenomena. GIS are committed to an implicit conceptualization based on static objects or system states, where temporal representations are mainly concerned with the states and changes of states of these objects or fields (Yuan, 1996). As a consequence, temporal extensions to GIS are lacking in their ability to reason about and model dynamic phenomena (Clarke *et al.*, 2001;Frank, 2001;Raper and Livingstone, 1995;Worboys, 2001). The divide between the spatiality of GIS and the temporality of traditional modeling software is not only found in computational limitations, but is a reification of the respective atemporal and aspatial theories the software embodies. The inadequacies of current GIS to support processes is due to a lack in theoretical foundation (Kavouras, 2001).

CA and ABM, although dynamic, are still based on system or object states at instants of time. As expressed by Epstein and Axtell, “[e]ach agent has internal states and behavioral rules. Some states are fixed for the agents life, while others change through interaction with other agents or with the external environment” (1996: 4). Process is typically presented as the relationship between the current and future states of cells or agents, defined by a set of behavioral rules. Processes are therefore implicit to the model, embedded in the rules of the agent or cell, yet they are not explicitly modeled, nor can they be directly inferred from changes between recorded system states. For example, in an ABM of urban sprawl, each agent may have a set of behavioral rules defining their movement and interactions. At each time step, the system state is logged in the form of agents and their attributes. However, whether the future system state of sprawled urban form is a direct result of processes such as rent increases in the inner city or increases in crime, is not represented or stored. The extent of ABM's ability to discuss process is to link the initial model setup or specification with the output through some form of spatial pattern metric (Parker and Meretsky, 2004;Rand *et al.*, 2003).

Similarly, EBM also focus on system states and their update. The equation itself represents the process, but its operation is typically not represented or recorded in the results. As with ABM, in EBM there are ad hoc solutions for determining the path of a process and which process is operating where, but no general solution or data model which addresses this directly. The modeling approach developed within this dissertation focuses on the representation and storage of processes expressed in current models, but with a process oriented data model. This approach avoids the loss of information through the cracks of time, such as through the imposition of an inappropriate temporal granularity that misses changes, as it requires operation at the level of the defined process.

3.2 Storing and Querying Processes

We can mine data for process information, or classify collected data into process types automatically or manually (Merz and Blöschl, 2003;Yuan, 2001), however current approaches to storing model output do not allow for easy querying of process information, as noted in Section 2.4. For example, Figure 1 below expresses this difficulty. Here the location of the black point moves from time one (t1) to time two (t2), yet given knowledge of the system state at each of those times, the process by which the point moves is not stored. Our ability to determine the process typically depends upon an in-depth knowledge of the model and the system it represents, and has the potential to result in the wrong process. In order to accurately determine the processes in operation the model must be rerun, applying the rules or equations over again. However, there are currently no common data models for representing processes, therefore extraction of this process information leaves us with no way to analyze or query it.

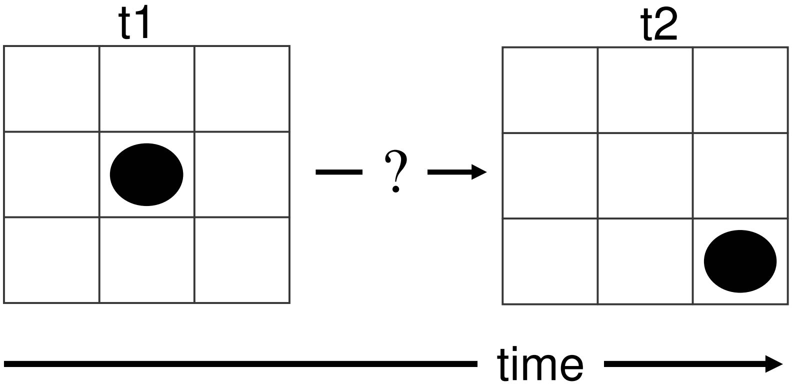


Figure 1. Process inference

Nor can we determine what processes are occurring at an instant of time, because in the traditional theoretical framework process by definition is something that occurs between time or system states. That is, process is the translation between system or object states at different times, therefore it cannot be represented in one time slice. Consequently, queries about where a process is occurring at an instant of time cannot be expressed with current methods.

Common approaches to modeling primitives are evident in the two basic types of queries on spatio-temporal representations, that is, state or changes in state. For querying states with current data models, only two basic types of queries may be asked: “what is at a specific location?” or “where is a certain attribute?”, the composition of which define the realm of possibilities (Goodchild, 2003;Peuquet, 2002). With the dynamic extensions of ABM, CA, and EBM, these queries are temporally qualified, yet there remain the two fundamental types of queries that can be asked. For example, given a specific agent, what are its associated properties at time x? Or, given a specific set of cells (i.e. location), what are its associated properties at time x? In terms of change queries, attributes and entities are queried as to if and when they changed by interpolating between these states. As a result, spatio-temporal databases are designed to store historical, present, and possible future data (e.g. for planning purposes), “they are not designed to record which processes activate a change” (Claramunt *et al.*, 1997: 7).

To understand, query, and explain processes, processes must be modeled. Claramunt *et al.* recognize that “there is a need to model dynamics behind changes in order to test hypothesis [sic] about their action. This problem must be addressed at the process level to discover how things happen and how entities are related into spatio-temporal networks” (1997: 2), yet maintain the object/process dichotomy. How or why questions cannot be easily asked or answered with methods based on current approaches focused on what, where, and when questions.

3.3 Process Analysis and Causality

Modeling a process is not just tracking and storing the movement of some object, such as an agent. Recording change does not equal process. Clearly we can track change, but with current data models we cannot hunt the processes that caused the change. We can associate outputs with various changes to the model structure or initial conditions, but cannot easily explore the causal mechanisms among processes that cause these results.

Analyzing the interaction of processes is important if we wish to see how various processes propagate through the system over time, and to determine which spatio-temporal points in the model to tweak. In simulating processes we may gain insight into their causal relations by storing their interactions. Questions regarding how the rules of the process affect the dynamics of the process (rather than the pattern produced by the process) may be better explored by modeling and storing processes.

3.4 Efficiency and Interoperability

As expressed in the beginning of Section 3, the proposed modeling approach largely addresses what is not being done rather than what cannot be done. However, the lack of a data model or structure to represent processes brings processes, in terms of modeling methods, back into view. In querying or analyzing processes, an argument can be made for the inefficiency of attempting to recreate processes each time in order to query the results for where certain processes caused changes in system states. The proposed methodology of explicitly storing process information overcomes this problem, allowing for queries similar in nature to current system state queries, but for processes, for example, querying for the location of processes, their attributes, or their change over time. Furthermore, state information can be derived from this modeling approach, so there is no loss of information. For example, in modeling the process of coastal erosion, the various eroded states of the system can be directly extracted from the process model.

Given the argument of Part I that all things are process, where structure or stasis are slow processes, process forms a more basic primitive. The proposed approach of modeling and simulation with process as the single primitive provides a common basic construct, which if applied to models of different domains would facilitate interoperability between models. Common representations of space-time, which has been one of the key problems of integrating GIS and environmental models, potentially allow interoperation at the process level rather than the model level, removing the effort required in translating between models. As was expressed in Chapter 2 with the

Bretherton diagram, this could be an important boon to modelers of complex systems deriving their model components from different fields of study.

4. Closing Comments

The primary methodologies for modeling geographic processes have focused on generating future system or object state representations and analyzing these system or object states and the differences between them; as expressed in a recent modeling text, “environmental models are focused upon change” (Mulligan, 2004: 29). The alternative proposed in this dissertation reformulates this tactic such that process information is explicitly represented and stored. This has the advantage of allowing for exploration into the dynamics of process interactions, explanation of those dynamics, and ultimately of presenting a new epistemological window onto the subject matter. Consequently, as a novel way of modeling the geographic phenomena studied it may provide new insights into how those geographic phenomena operate.

CHAPTER SIX

Conceptual Model

> Nobody can understand the full meaning of a theory and a set of data without first having grasped the fundamentals of the chosen mode of discourse.
>
> (Olsson, 1975: 11)

1. Introduction

This chapter describes the conceptual model for the implementation of the process model. As such, it informally maps the ontology developed in Chapter 4 to an abstract implementation description, while ignoring the details of implementation and the details of a domain specific application. However, the restrictions imposed by basic computational methods are recognized, such as the principles of object orientation and the discrete nature of computation. To place this chapter in context, in the same way that Chapter 3 forms the theoretical conceptualization for the implementation of the ontology in Chapter 4, so this chapter forms the methodological conceptualization for its implementation in Chapter 7. Hence, the reification of the following conceptual model will be discussed in the next chapter.

2. Tightening Some Conceptual Screws

For the sake of modeling, a new basic construct is introduced that extends the ontology from the single primitive of process, to a type of restricted process, termed here a parameter. Parameters are instituted due to the inability to define a complete system of processes, typically representing the external input to the model. A process can be modeled as a parameter in the sense that it is an encapsulated process, where none of the internal workings of the process are evident in the parameter, merely a representative value. Although this is an application problem derived from the difficulty of any domain to completely define and model its research subject, and the problem that we cannot model absolutely everything, it is considered here as it has a general impact on the methodology that crosses most domains.

Parameters are practical abstractions for modeling geographic processes that are purposefully defined by the researcher in two scenarios. First, parameters are defined when we do not want to or cannot model the whole process, for reasons such as minimizing the complexity of the processes modeled or restrictions imposed by software, hardware, or other external influences. Second, parameters are defined when the observed temporal grain of the phenomenon exceeds the temporal extent of the model. For example, in the first case, to model the process of runoff in a watershed the process of precipitation must be included; however, we may not want to model the whole process of precipitation. Precipitation can then be included in the model as a parameter, represented as a value at a point or over some area to be used by the runoff process model. Extending this example to the second case, the geomorphology of the watershed may be considered a parameter in the runoff model. Changes in geomorphology are measured with a temporal granularity that exceeds the temporal extent of the process model, that is, geomorphologic changes are observed to take longer than the time the model takes to run, yet they are included because geomorphology has an impact on runoff processes.

Parameters impact on the processes being modeled and can be modified by those processes. However, they have no behavior of their own. Parameters influence processes whereby the process registers its presence and value at a specific location. Parameters are modified by processes when their values are changed by a process. For example, in a model of erosion, the erosion process will affect the geomorphology, and the geomorphology will influence the dynamics of the erosion process. Yet, geomorphologic change is outside the temporal extent of the model and therefore geomorphology has no defined behavior of its own.

One further point of note is the use of object-orientation (OO) in modeling. Regardless of the conceptual saliency of computational objects to perceived geographic objects, they should not be confused. A computational object, as a programming implementation, can be used to represent anything, including a process. For example, Wachowicz (1999) uses an object to represent events. The same model can be programmed in many different ways, yet at runtime the results are the same. The cognitive consonance of objects, in terms of OO, has been overly stressed in some cases, where the conceptual objects limit the implementation OO objects. Limiting implementation objects to these conceptual objects maintains the current focus on cognitively salient things for modeling geographic processes, where the objects, attributes, and methods of object-orientation form modeling primitives (for example Bian, 2000).

3. Process Model Structure

A process model is conceptually structured following the principles of object orientation, that is, abstraction, encapsulation, inheritance, and polymorphism. This provides the basis for a generic description without having to align with a specific language. As such, a specific language does not limit or constrain the structure of the conceptual model.

The process model consists of three base classes from which domain specific models may inherit methods and properties, namely: process, parameter, and model. This extends the ontology of Chapter 4, which merely contained a process class, through recognition of some of the restrictions imposed by the modeling domain and computational environment. The model class forms the modeling environment for the processes and parameters; it is incorporated in order to define operational aspects such as the initiation of the model, its display, and parameter scheduling. The process and parameter classes define the common properties and methods that all inheriting process and parameter instances implement. All aspects of the model are conceptually encapsulated within these three classes.

3.1 Model Class

The model class only contains methods pertaining to the setup, scheduling, and recording of the processes and parameters. The setup method creates the processes and parameters that initiate the model. The scheduling method iterates over the parameters and specifies the creation of the process instances based on the thresholds defined in the process class methods.

3.2 Process Class

The basic process class contains a set of properties and methods that all subclasses inherit. The methods incorporate rules for the interaction between process types and between processes and parameters.

3.2.1 Properties

The properties of a subclass of a process follow those defined in the ontology, namely spatialXExtent, spatialYExtent, spatialZExtent, spatialGrain, temporalExtent, temporalGrain. Instances of the process include location properties of x1, x2, z1, y1, y2, z2, and a property defining some value of the process, such as energy or mass. An additional property, ID, is added to provide a unique identity code for each process instance.

3.2.2 Methods

Get and set methods for all of the properties are defined in the class. Methods for the creation and destruction of other processes are also specified, as well as methods defining its own behavior.

3.3 Parameter Class

The parameter class contains a set of properties and methods that types of properties inherit. In contrast to the process class, the parameter class is not spatially dynamic, that is, it does not have a changing set of x1, x2, or y1, y2 properties. Rather, it is located at a point or over an area. This conforms to the classic data models of point, line, polygon, and pixel.

3.3.1 Properties

The parameter contains the following properties: temporalGrain, temporalExtent, spatialGrain, spatialExtent, and inputFile. The temporalGrain of the parameter defines how often it is updated; for example, precipitation as a parameter may be updated hourly. The temporal extent defines the total number of times the parameter is updated. The spatial grain and extent, although typically implicit in the input file of a raster or vector layer, is specified as it may form the basis of the spatial extent of the process. The inputFile property defines the input file(s), which contain information on the spatial location and value of the parameter.

3.3.2 Methods

Parameters have no methods other than get and set methods defining their properties.

4. Model Behavior and Output

Conceptually, processes were defined in Chapter 3 as creating a process space-time manifold. However, for modeling processes, an absolute reference frame of space and time is recognized for its utility. Three notions of space-time are subscribed to, absolute, relative, and relational. In absolute space-time the four axes of space-time are used as a measurement framework, describing the relationships among processes through time, and dictate the update of input parameters. This forms the basis for the initiation of the model.

Within this absolute spatio-temporal reference framework, processes and events create a relative space-time through their behavioral rules and properties. This internal time relative to processes' internal dynamics, defines their temporal extent with reference to the absolute framework. This second notion of time has been termed "real" time by Couclelis and Liu (2000). Thirdly, each process experiences relational space-time when other processes or parameters influence it. For example, the relative space-time of a process could change in response to synergistic forces with other events, in response to changes in the relational space-time of the process.

In creating this spatio-temporal manifold, the behavior of a process is defined by a set of rules. These rules not only define the dynamics of each process in relation to parameters, but the interaction among processes. Whenever a process changes, it records its identity and properties to an external database, which can then be queried.

5. Process Queries

The output of the process model is used to query processes for their state or their dynamics at an instant of time or over an interval of time. These two base types of queries can be applied to properties or attributes of the processes, which includes spatial location. Given the nature of the process data model, the spatial character of a process includes: direction, location, and extent.

5.1 Process State Query

Process state queries characterize the state of the modeled system at an instance or over an interval of time. For example, questions such as “Where is a process over an interval of time?” or “What process is operating at an instant of time?” can be asked based on the process’ attributes or spatial characteristics. The results of process state queries at an instant in time or over an interval of time can be represented as a table of process instances or represented visually as a static display of the processes within the space defined by the model, for example, the distribution of infiltration processes within the space defined by a watershed parameter. Additionally, in the case of a query over an interval of time, a graph can be produced that represents some attribute or a count of the selected processes (y-axis) over the interval of time (x-axis).

5.1.1 Process Attribute State Query

A process attribute state query involves a search for a process that has a particular attribute at an instant of time or over an interval of time. For example, “What processes have 25 Joules of energy at 2’oclock on Saturday the 12th of May?”. Or, “Select the processes that are transporting sediment of greater than 2mm in diameter between the 35th and the 45th unit of time”.

5.1.2 Process Spatial State Query

Querying for spatial properties of a process at an instant of time or over a time interval is termed a process spatial state query. For example, "Search for processes that are located within the spatial region defined by the bounding box (x1, y1, x2, y2) at time 56". Although this is a type of attribute or property query, it is drawn out as the geography of a process is often of special interest.

5.2 Process Change Query

A process change query involves the search for patterns of change that define the dynamics of the process at an instant of time or over an interval of time. As with process state queries, the three outputs of table, display, and graph, also apply to process change queries.

5.2.1 Process Attribute Change Query

The attribute change of a process over an interval of time can be queried in a number of qualitatively different ways. For example, find processes that have changed an attribute:

- from value a to value b
- from positive values to negative values
- from greater than a to less than b
- from the range a to b to the range c to d
- by percentage or absolute change

More complicated expressions can then be built up from these simple primitives, defining complex patterns of change.

5.2.2 Process Spatial Change Query

The spatial change of a process is based on the location attributes of the process, x1, y1, z1, x2, y2, z2. With the *nen* data model, the basic form of query is defined as a change in location; a higher level form of query of change in orientation is also included as it is a useful qualitative abstraction that has meaning in models of processes where direction is important. The change of location of a process can either be defined with a specific (x1, y1, x2, y2) location or with a region, such as that defined by a bounding box. Thus there are four basic combinations: from specific location to location, from specific location to region, from region to specific location, or from region to region. For example, in Figure 1 below, a query can be expressed that searches for processes that moved from the dashed square at time one (t1) to the dashed square at time two (t2).

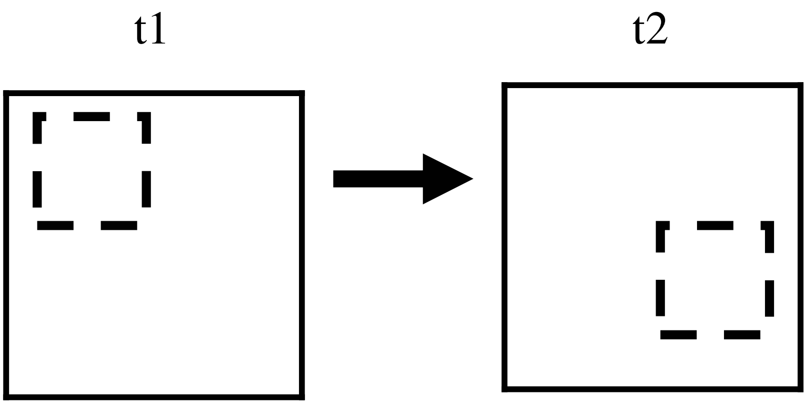


Figure 1. Example of a spatial change query

For orientation, the query involves specifying the change in the relationship between the x1 and x2 and/or y1 and y2. The relationships are specified by the three relational operators: equals (=), greater than (>), and less than (<). For example, Figure 2 illustrates the following query: select processes that have changed in orientation such that the process attribute $y2_{t2} > y2_{t1}$.

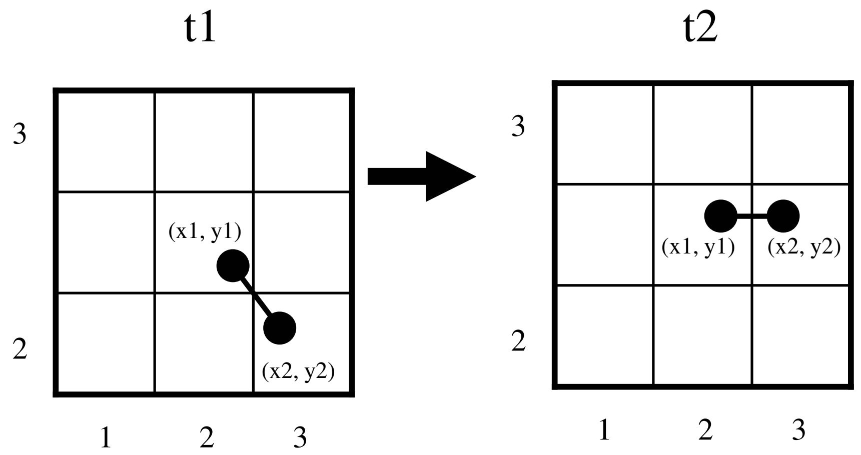


Figure 2. Example of a process spatial change query for orientation

Beyond the simple process query, which is a basic analytic device, what quantitative measures can be derived from the process model that allows for comparison between models? This and other analytical questions go beyond the scope of the dissertation, but form the obvious next step towards a better understanding of the operation of processes.

6. Towards Implementation

The general structure of the simulator has been presented in a high level form in order to extract the methodology from the restrictions imposed by the implementation language or implementation tools. From this abstract discussion, an implementation can be developed which applies this general method. In the next chapter, this implementation will be described.

CHAPTER SEVEN

Prototype Implementation

1. Introduction

The implementation of the conceptual model lies in a field of possibilities. Varying the approach taken to implement a conceptual model, although a technical issue, will also have implications for the results of the model (Gulyás, 2002). While recognizing this conundrum, one must begin somewhere. In what follows the approach taken will be described in detail, including some of the design issues and assumptions in the development of the process simulation tool. As will be described further, the ontology was considerably extended in order to accommodate limitations arising from the tools available.

This chapter begins by describing the simulation environment used to create the model in Section 2. Section 3 outlines the simulation framework, describing the main class structure, which is followed by a description of the simulator in Section 4. Section 5 presents the query tools developed for the simulation results and the different ways to display those results. Section 6 describes a trial implementation of an ontology based simulation. Section 7 concludes the chapter.

2. Simulation Environment

Given the discussion of space, time, change, and substance at the conceptual level in Chapter 5, it becomes evident that the realities of the implementation environment available limit the implementation of the modeling approach. From the discrete confines of the computer to the imposed structure of object-orientation, technologically the model is constrained to a particular framework. The straitjacket of choice is Java, including the incorporation of the RePast (Recursive Porous Agent Simulation Toolkit) library, an open source agent-based modeling environment created by Social Science Research Computing at the University of Chicago[10]. RePast is primarily used for its display and scheduling classes, and also has the advantage of containing Java classes for importing GIS raster data (ESRI ASCII raster files). As a caveat, the agent-based environment is not used to do agent-based modeling per se; rather, its classes are used in order to simulate process as the primitive modeling construct. In the terms of object-oriented implementation, a process forms an object or class of objects.

3. Simulator Structure

The simulator, called flux, inherits and extends a number of basic operating classes from Repast, namely scheduling classes, display classes, and a base model class. The objects developed in the flux package in turn form the base set of classes for a domain model (Figure 1). Note that in what follows only the main simulation objects will be discussed,

[10] http://repast.sourceforge.net/

ignoring a number of objects developed to deal with the more mundane aspects of the model operation, such as extending RePast for display and recording of processes. A UML class diagram of the main modeling classes developed in the flux package and their relationship to RePast classes can be found in Appendix C.

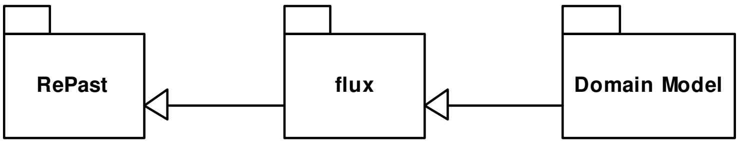


Figure 1. Model Inheritance Structure

The flux model contains a set of interfaces and default classes that define the basic structure of the process model, including methods that must be implemented by an inheriting domain model. The objective was to develop as much generic functionality within the flux classes, thereby minimizing the code to be developed within the domain model. The general class structure of the modeling primitives in the flux package is presented in Figure 2 below; a modified UML class diagram.

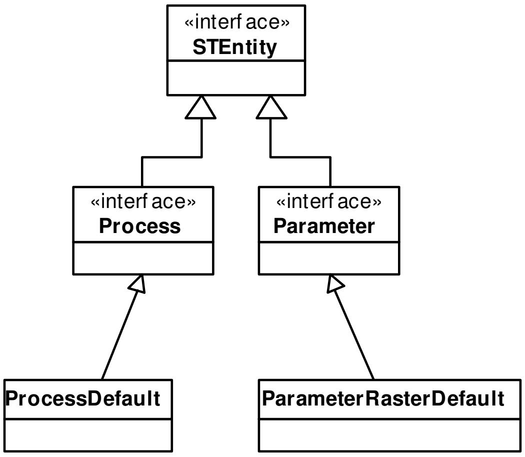


Figure 2. Model Class Structure of Primitives

The STEntity is the top-level interface that specifies the methods that any inheriting process or parameter instance, such as ProcessDefault and ParameterRasterDefault, must implement. For example, these methods include set and get methods for the properties: temporal grain, spatial grain, temporal extent, and spatial extent.

The Process interface extends the STEntity interface with added methods that an inheriting process is required to implement. For example, set and get methods for properties defining the location of the process, that is, the x1, y1, z1, x2, y2, and z2. The ProcessDefault class implements the Process interface with a set of generic properties and methods that are widely applicable to processes in other domains. For example, methods that take care of the display of the process as a node-edge-node triple and the recording of the process are included in this interface.

The Parameter interface specifies various get and set methods for a parameter, such as its ID and Value. The ParameterRasterDefault is but one implementation of Parameter, and extends RePast's RasterSpace class to incorporate added functionality such as a generic method for raster coloring, and a method that allows for searching the Moore neighborhood at a range of sizes.

4. Simulation and Results

In order to simulate the model, it was necessary to introduce two new classes: ProcessController and ParameterController. These two classes were implemented in order to control their respective process and parameter classes and instances, providing a useful intermediary between the process model and the process classes. These two classes are defined in the flux package, where the ParameterController is an interface with methods to be implemented, and the ProcessController forms an abstract class with a few generic methods. Figure 3 below presents the basic set of model classes to be used in a process model. Initial explorations into the possibility of programming each process as a thread as an alternative to these two controlling classes, suggested that it would be too computationally intensive for the number of processes to be represented and more difficult to develop and control for the scheduling.

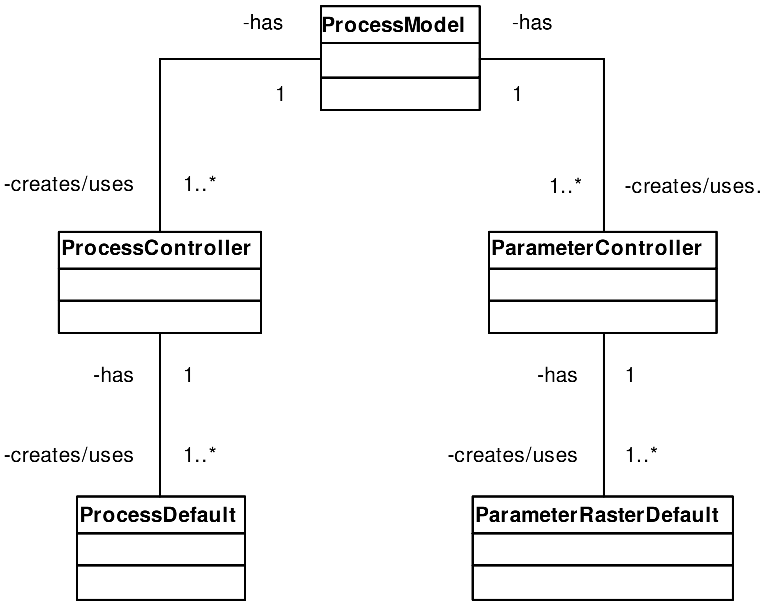


Figure 3. Basic model operation classes

A sample operation of the model is depicted in Figure 4 below as a UML activity diagram. At the initiation of the model a series of setup methods are implemented, such as the creation of the ProcessController and ParameterController and the display surface. The model then iterates over a set of commands that update any of the parameters needing to be updated, calls the ProcessController to operate its processes, updates the display, and then calls a method that records the results of each process in a text file at the end of the model run. When the Process controller is called to operate, it iterates through each process until the process runs out of energy. This property of process energy is used to calibrate the relative and relational spatio-temporal extents of the process with the parameter defined model update. Each time a process instance is created or changed it is recorded in a text file containing all records of the class of processes it belongs to. Currently the ID, location, energy, and value of the process are recorded. However, this can be extended to any property of the process.

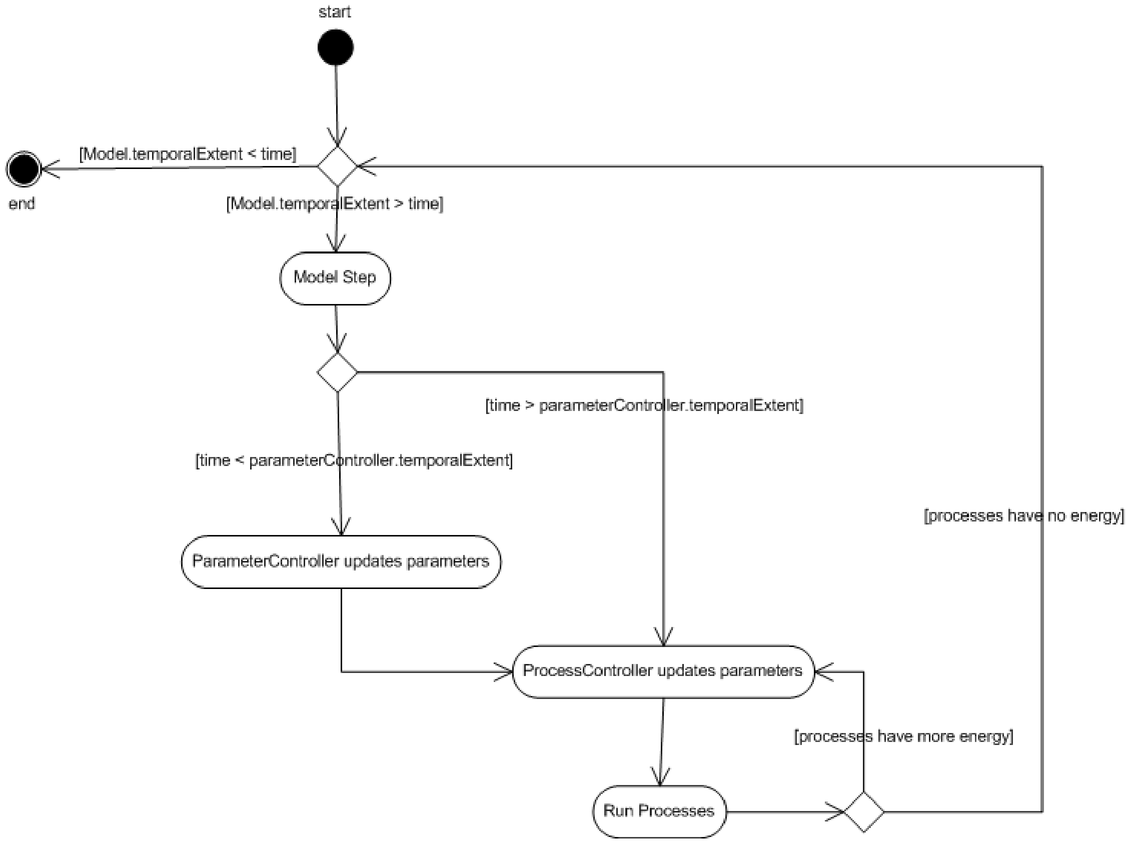

start
[Model.temporalExtent < time]
end
[Model.temporalExtent > time]
Model Step
[time > parameterController.temporalExtent]
[time < parameterController.temporalExtent]
[processes have no energy]
ParameterController updates parameters
ProcessController updates parameters
[processes have more energy]
Run Processes

Figure 4. Sample simulation diagram

As expressed in Chapter 6, the scheduled time forms the absolute framework within which relative and relational notions of time are implemented. The scheduled time is typically defined by an input parameter, such as the hourly input of precipitation; the relative time of associated processes is specified by the operation of the process; and the relational time is defined by its interaction with other processes. Each operation or interaction requires a certain amount of energy, which is relative to the absolute time defined by the scheduler.

The simulation can be run in batch or GUI mode. GUI mode allows for visualization of the simulated processes and the ability to step through simulation runs. Figure 5 below illustrates a sample representation of a process, where the nen represents each process instance. The process displayed in blue in Figure 5 below, is a sample of overland runoff over a digital elevation model represented in green; higher elevation is represented by lighter tones of green.

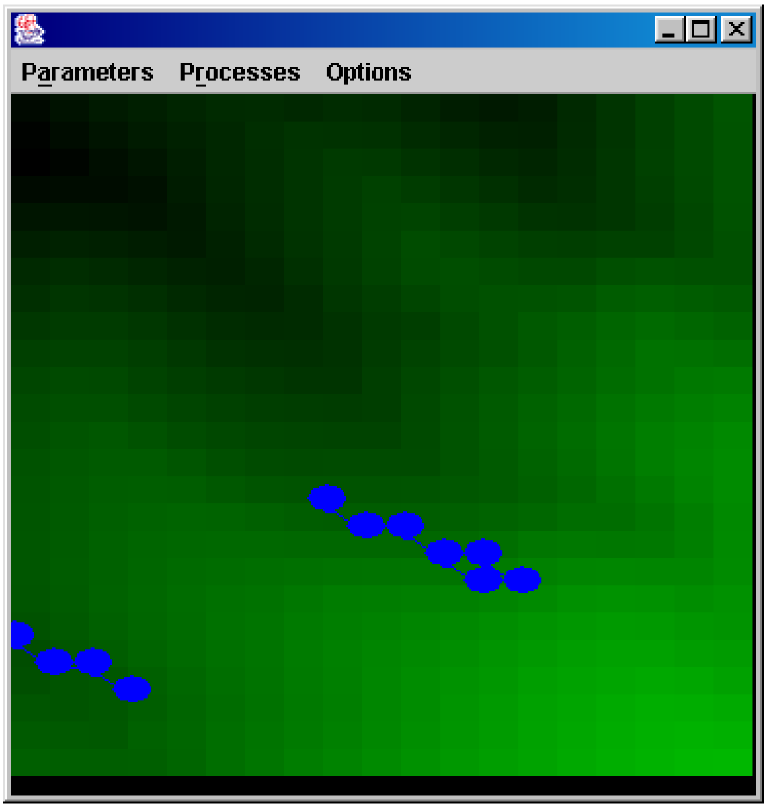


Figure 5. RePast GUI with process in operation

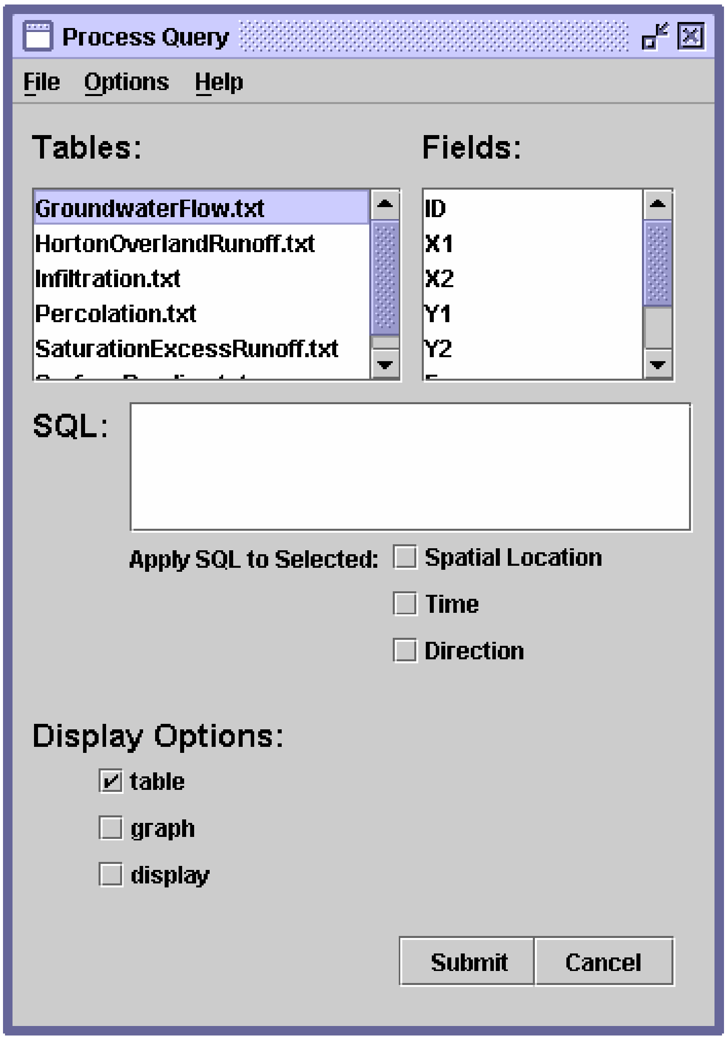
5. Querying the Model Results

The output from a simulation can then be queried for the state and changes of the modeled processes. The results of a query are then displayed in a table, chart, or display, as will be illustrated below.

5.1 Queries

Regardless of the limitations of SQL (Structured Query Language) for querying spatio-temporal data (Egenhofer *et al.*, 1999), for the purposes of this prototype SQL proved useful and powerful for querying the results of the model. With the results stored in text files, the JDBC (Java Database Connectivity) API was utilized to access and query this text file as a database via an ODBC (Open Database Connectivity) interface to connect to the database[11]. The types of queries expressed in the conceptual model formed the basis of a GUI (Graphical User Interfaces) that allows the user to query for attribute and spatial states and changes of the processes stored in the text files (Figure 6). All of the specified kinds of queries can be expressed in SQL, however, some functionality has been added to simplify querying, namely tools for delineating spatial and temporal bounds, as well as direction. These tools are found under the Options menu and implemented by ticking the appropriate checkbox. To date only the space and direction tools have been implemented.

[11] The JDBC API is available from http://java.sun.com



Tables:

- GroundwaterFlow.txt
- HortonOverlandRunoff.txt
- Infiltration.txt
- Percolation.txt
- SaturationExcessRunoff.txt

Fields:

- ID
- X1
- X2
- Y1
- Y2

SQL:

Apply SQL to Selected: □ Spatial Location

□ Time

□ Direction

Display Options:

☑ table

□ graph

□ display

Figure 6. Process Query tool GUI

When the Process Query tool is run the file names found within a specified model output folder are imported into the Tables combo box. Selecting one of these tables populates the Fields combo box with the fields of the table, which can be used in the SQL query to be entered in the SQL text area.

5.1.1 Process Attribute State Query

An example of the SQL syntax for a process attribute state query is as follows:

SELECT *

FROM table

WHERE attribute_id = some_value;

5.1.2 Process Spatial State Query

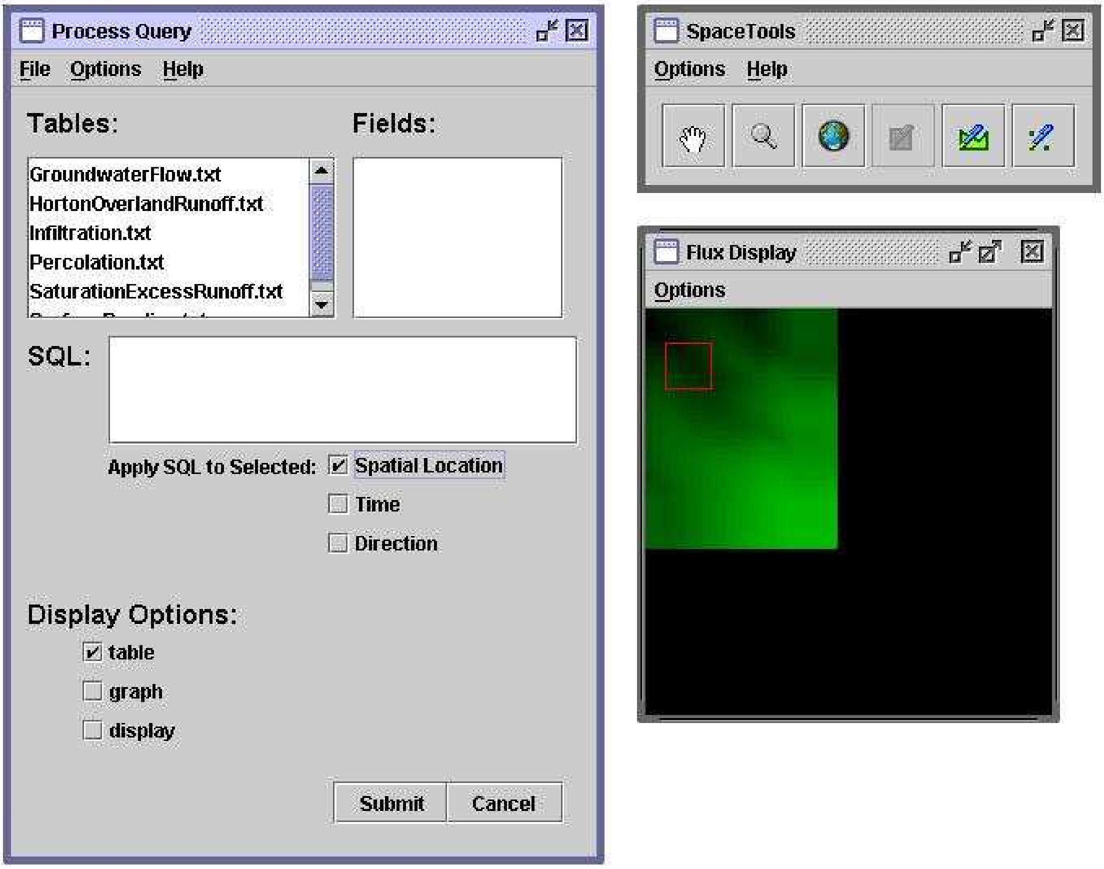
An example of the SQL syntax for a process spatial state query is as follows:

SELECT *

FROM table

WHERE X1 **BETWEEN** boundaryX1 **AND** boundaryX2 **AND**
X2 **BETWEEN** boundaryX1 **AND** boundaryX2 **AND**
Y1 **BETWEEN** boundaryY1 **AND** boundaryY2 **AND**
Y2 **BETWEEN** boundaryY1 **AND** boundaryY2 **AND**;

The values of location may also be specified as particular values of X and Y or with any other integer operators. Two tools were created in order to simplify this process, SpaceTools and DirectionTools. SpaceTools allows the user to load their modeling backdrop, such as a DEM, and select either a spatial area with a rectangle drawing tool or a point location with a point drawing tool (Figure 7). Thus far these are the only two tools that are functional of the six tools displayed. By ticking the appropriate checkbox on the main Process Query GUI, this area or point location is used to select only those processes within the bounding box or at that point when the query is submitted.



Tables:

GroundwaterFlow.txt
HortonOverlandRunoff.txt
Infiltration.txt
Percolation.txt
SaturationExcessRunoff.txt

Fields:

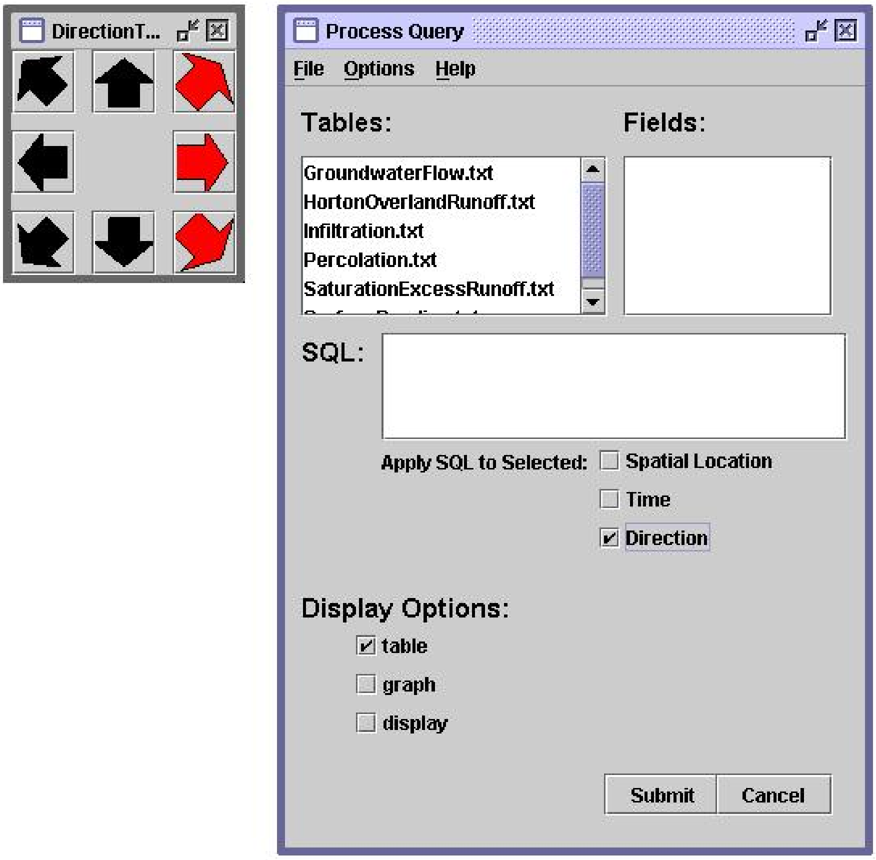
SQL:

Apply SQL to Selected: ☑ Spatial Location
□ Time
□ Direction

Display Options:
☑ table
□ graph
□ display

Figure 7. Query using SpaceTools

The DirectionTools, simplifies the specification of a direction based query by allowing the user to select processes operating in a direction of interest (Figure 8). For example, in Figure 8 all processes operating in the North East, East, and South East directions will be selected if the appropriate check box is selected before submitting the query.



Tables:

GroundwaterFlow.txt
HortonOverlandRunoff.txt
Infiltration.txt
Percolation.txt
SaturationExcessRunoff.txt

Fields:

SQL:

Apply SQL to Selected: □ Spatial Location
□ Time
☑ Direction

Display Options:

☑ table
□ graph
□ display

Figure 8. Query using DirectionTools

5.1.3 Process Attribute Change Query

An example of the SQL syntax for this query is as follows:

SELECT *

FROM table t1

WHERE t1.attribute_id = some_value **AND** t1.process_ID **IN** (

SELECT t2.process_ID

FROM table t2

WHERE t2.attribute_id = some_other_value);

5.1.4 Process Spatial Change Query

Because the location information is stored in the same manner as an attribute, the form of the query is the same as for an attribute change query. An example of the SQL syntax for this query is as follows:

```
SELECT *
FROM table t1
WHERE t1.X1 = some_value AND t1.process_ID IN (
                SELECT t2.process_ID
                FROM table t2
                WHERE t2.X1 = some_other_value);
```

The query can be modified to incorporate any of the location attributes and any integer operator.

5.2 Displaying Results

The results of the queries may be displayed in a chart, two-dimensional display, or text file, depending on the query type. For example, displaying results in a chart only applies to queries for a certain quantity over time, such as the value of an attribute from time step 5 to time step 45. A sample chart output is displayed below in Figure 9, where time is the x-axis, and a count of processes from a dummy simulation is the y-axis. The chart display utilizes the JFree Java library, which includes classes for plotting charts.

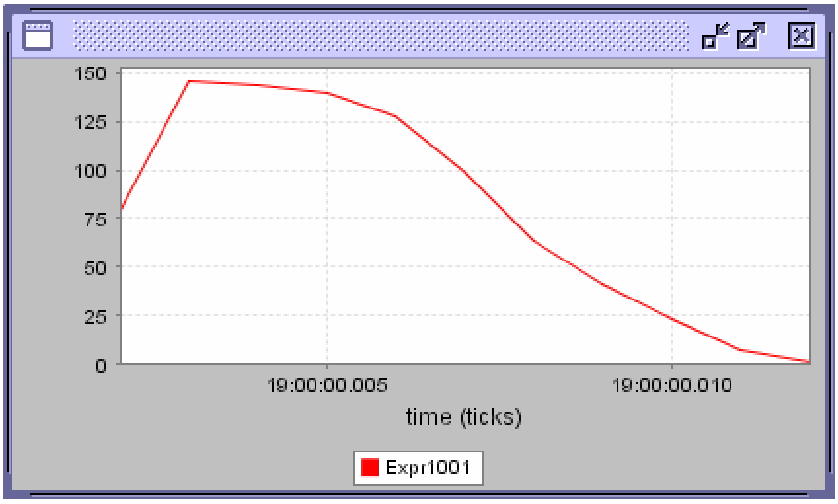


Figure 9. Sample chart output

The display output recycles some of the RePast and flux code to present the spatial distribution of a process. Figure 10 below, provides a simple example of a query that selects one process. The text output is simply a selected subset of the original results text file.

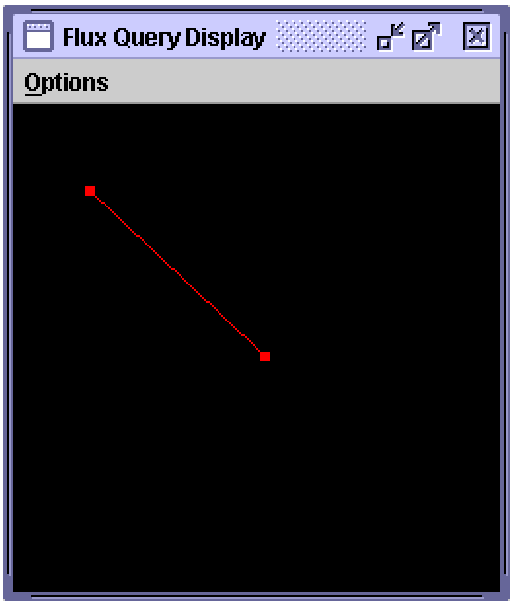


Figure 10 Sample display output

6. Ontology Based Simulation

The simulation framework discussed above is for a generic implementation where the domain model is specified and developed in Java. This environment is what is used in Section III for the domain application of watershed modeling. A prototype was also developed for an ontology driven simulation in order to provide a proof of concept and to bridge the modeling gap from theory to implementation.

The simple model used for development involved a single process: overland runoff, and two parameters: precipitation and a digital elevation model. The model ontology is provided in Appendix D, and defines two rules, the creation of the process and the change of the runoff process. A process is created when the intensity of precipitation exceeds a constant. The behavior of a process is governed by a simple rule where runoff occurs in the direction of lowest elevation if a lower elevation exists.

In order to implement the ontology based model, the flux ontology was also modified (Appendix E). One of the more significant changes was to incorporate a range of built-ins, that is, built in methods that were predefined in the flux modeling environment. Much of the original flux framework was also modified in order to utilize the ontology, in particular an OntologyReader class was developed, which was responsible for reading the ontology, generating the appropriate process and parameter objects described in the ontology, and evaluating the rules. The results from the model were generated and stored in the same manner as the flux simulator and the running simulation was presented in the same display panel as depicted in Figure 5 above.

Although there is much potential for future developments towards model interoperability with an ontology based system, there are few benefits at this stage of development. Due to the way it was implemented, there was significant computational overhead in checking through the process rules in the ontology at each time step. Furthermore, defining the rules in SWRL with built-ins was very labor intensive and limited in expressivity.

7. Concluding Part II

This final chapter of Part II has presented the general implementation of the theory and conceptual model developed in previous chapters. The base set of modeling classes that compose the flux framework and a simple query tool has been described, and forms the basis for the application model of watershed runoff to be discussed in the next chapter, the first of Part III. Thus far, a basic prototype for an ontology driven simulation has also been developed, as described in Chapter 6, but is limited due to the computational complexity incurred in running a simulation in this manner. As yet, the analysis of processes with the proposed representation has not been addressed, only a basic querying framework, which forms the first steps towards analysis. The development of analytical measures for the nen data model falls outside the scope of the dissertation, but forms a likely point for extension. A probable avenue for initial exploration includes recent work on vector field operations (Li and Hodgson, 2004).

CHAPTER EIGHT

Watershed Modeling Review

1. Introduction

Watershed modeling has been selected as the testing ground for the theory and methodology created in this dissertation. It was chosen because it is a well researched area of geography, where processes are explicitly considered and specified. The purpose of this chapter is to give an overview of the most common methods of watershed modeling in order to provide some background and justification for the application of the proposed methodology. In what follows, Section 2 presents the status quo, including research frontiers in data models developed in the hydrological modeling community studying watershed dynamics; Section 3 follows with a brief consideration of these models in the light of the proposed methodology discussed in Part II; and Section 4 concludes the chapter.

2. Modeling Watershed Hydrology

There are many ways of classifying models both in general and within the field of hydrology. For example, models may be classified by conceptual type, such as empirically or physically based models, or by spatial type, such as lumped or spatially distributed models (Beven, 2001;Grayson and Blöschl, 2000;Mulligan, 2004;Singh, 1995). The slightly different approach taken here is to consider them according to modeling primitives, in order to clearly state the case for the value of an alternative modeling primitive based on process. In particular, distributed models are the focus of this chapter as they explicitly model the spatial nature of hydrological processes.

2.1 Equation Based Models and GIS Connections

Distributed models traditionally come in two basic forms, distributed modeling and spatially distributed modeling. Distributed modeling divides the watershed into discrete spatial units, computes the response of each unit to inputs such as precipitation, and then combines them to give the response for the entire watershed, such as the SHE model (Abbott *et al.*, 1986a;Abbott *et al.*, 1986b). This approach does not capture the spatial interaction of processes at or between spatial or temporal scales, "[i]n some respects the distributive mechanism means that the distributed model is essentially a 'lumped' model at grid scale" (Ward and Robinson, 2000: 348). In contrast, spatially distributed modeling explicitly deals with interactions among neighboring spatial units. Such models are typically used to route water flow over a landscape using flow direction algorithms, the simplest of which results in sending water to a neighboring downslope element that has the greatest elevation decrease, known as the D8 method.

Most advanced spatially distributed rainfall-runoff modeling are based on the classic, physically-based, distributed model blueprint designed by Freeze and Harlan (1969). This design describes a basic framework for numerical modeling, with a set of partial differential equations operating over a set of points arranged in a three dimensional grid representing the watershed. The more recent blueprint provided by Beven does not break the data modeling mold of the original, rather considers how different models and their parameters might fit into a model space (Beven, 2002). This basic data modeling approach is ubiquitous in spatially distributed models, that is, the use of the traditional pixel, point, line, or polygon primitives, which at each instant of time in the model are described by a set of attributes. Pixels are used in grid representations of the watershed, specifying a value such as elevation at a specific location; points can be similarly used to represent a continuous field of data, or may be used to represent specific data collection points such as lysimeters or piezometers; lines are typically used to model flow networks; and polygons are used for representing larger areas of interest such as hydrological response units. The underlying general data structure for all of these primitives is defined by a spatial location *x*, *y*, *z*, a time point *t*, and a set of attributes $a\colon\{x, y, t, a_1 ... a_n\}$. Regardless of whether the equation is physically based, empirically based, or stochastic in nature, the underlying representational devices, the data models, remain the same.

Consequently, the types of output available to the model user, which lead to analysis and querying techniques, also remain the same. Although substantive output may vary from model to model, such as whether sediment or chemistry is modeled (Borah and Bera, 2003;Borah and Bera, 2004), the structure of the information provided is consistently the state of that output at each instant of time. For example TOPMODEL is a spatially distributed model that uses an index of similarity called the topographic index to define its spatial units, and uses a flow routing algorithm to direct water through these units (Beven, 2001). The output of the model predicts watershed discharge and the spatial distribution of saturation at any instant in the simulation or as a cumulative output at the end of the simulation.

Since the mid 90s, there has been an increasing amount of research and development on the integration of hydrological models and GIS (Feng and Sorokine, 2001;Romanowicz and Beven, 1993;Streit and Wiesmann, 1996). This linkage of GIS and hydrological modeling has ranged from loose coupling, simply the transfer of data from one program to the other, to hydrological models embedded within a GIS, such as the LISFLOOD model developed within PCRaster by De Roo *et al.* (2000). This integration has aided modeling by easing problems of spatial data input and by tapping into the data management and analytical tools of GIS. Yet, as with earlier models, the underlying data models remain the same, as expressed by Maidment who specifies six basic data structures used in these models, namely three basic (point, line, polygon), and three derived (grid, TIN, network) (Maidment, 1993).

For example, the Automated Geospatial Watershed Assessment tool (AGWA) is a hydrological watershed modeling tool that integrates a GIS and the existing hydrological models of KINEROS2 and SWAT. In their description of the AGWA tool, Miller *et al.* (2002) give examples on change detection in water yield, that is, supporting the visualization of increase or decrease in the spatial distribution of water runoff over time; yet this does not provide any insight into the processes that cause these changes. A further example of the difficulty of relating process to form is provided by Gurtz *et al.* (2003), who implemented and analyzed the results of two models for the same catchment, namely WaSiM-ETH, a physically based and grid modeled water balance model, and PREVAH, a conceptual model based on hydrological response units. Despite that these two models are very different, assigning water flow to different processes, both models simulate watershed discharge realistically in comparison with observations.

The results of such models can be classified as temporal, such as the hydrograph, spatial, such as the accumulated spatial distribution of runoff, or spatio-temporal, such as the change in the spatial distribution of runoff over time. These results are used to compare and validate models (Veith *et al.*, 2003). Such a traditional data modeling focus leads to query and analysis of the state of entities existing in their totality at an instant of time, or for the difference between states of the entity at different time instances (for example van Oosterom *et al.*, 2002). For an example of the latter, Gao *et al.* use sequences of frames to show change in the distribution of attributes as physical fields (Gao *et al.*, 1993).

2.2 Other Computational Simulation Environments

As introduced in Chapter 5, Cellular Automata (CA) presents an alternative approach for modeling spatially continuous phenomena. Recent advances in modeling geomorphology change use CA, which extend the spatially distributed modeling approach (Coulthard *et al.*, 2000;Favis-Mortlock *et al.*, 2000;Haff, 2001;Pullar, 2003). Not only do the grid cells interact, for example, excess energy in one cell may be transported to a neighboring cell based on a range of cell characteristics, but CA allow for the interaction between the structure of the landscape and the processes operating over it.

However, as with the earlier models of Section 2.1 in this chapter, the data model used remains state based. Each cell in an application of CA contains information about the state of that cell at an instant of time. Thus the resultant dynamics of the model can only be interpolated between time slices.

3. An Alternative Data Model

The problem with interpolation is that the wrong process may be interpolated. Take, for example, the case provided by Baird who finds two quite reasonable yet distinctly different explanations of a pattern found in the output of a hydrological model (Baird, 2004). Baird observes that the temporal pattern of high initial flow rates in the soil, followed by a steep decline after a precipitation event, can be explained by two different processes, one being the importance of macropores in a model utilizing a combination of Darcy's law and the Richards equations, the other by the entrainment of air bubbles which over time coalesce and block the flow of water. Representing processes explicitly provides the opportunity to explore which processes are dominant and whether our descriptions of those processes are correct. The proposed modeling approach provides a laboratory for testing process descriptions rather than a laboratory for testing state descriptions.

As expressed by Mulligan, “there are still areas in which the complexity of hydrological processes is so great, or the information so little, that we do not understand the processes well enough to develop reliable models” (2004: 117). If the processes are not understood, how can they be modeled, visualized, and explained in a model? This raises the question of whether we are capturing the right kind of data, whether current methods of measurement can record process information. The proposed approach, while not solving the difficulty of a lack of process data, allows for testing hypotheses about descriptions of processes. It permits the exploration of rule spaces rather than parameter spaces. An advantage of a rule based approach rather than equation based is the easier inclusion of qualitative information, particularly for defining thresholds, such as expert “non-encoded” knowledge (Seibert and McDonnell, 2000).

4. Review Conclusion

In conclusion, the modeling methodologies found in watershed runoff models mirror the standard approaches evident in dynamic spatial modeling in general. As such, these methodologies do not explicitly represent the processes that are embedded in the model. Without representation, the processes cannot be analyzed or queried; rather, they can only be deduced from the modeled states of the system, a process that is fraught with the potential for error. The alternative methodology proposed in this dissertation attempts to alleviate such problems of process interpolation and provide new scope for different forms of query and analysis.

CHAPTER NINE

RCEW Runoff Simulation

Different conclusions merely reflect the different ontologies embedded in the chosen definition of identity.

(Olsson, 1975: 94)

1. Introduction

The objective of the model implementation is to test the theory and methodology described in Parts I and II of this dissertation. As such, it aims at a representation of watershed runoff that captures the main processes discussed in the literature, yet remains simple enough to be completed within the time frame of the dissertation and enable testing and exploration of the model. Unfortunately beyond working with a subset of the data for development purposes, computation constraints limited the models application to a spatially and temporally restricted area, as will be further explained below. The rest of this chapter is structured as follows, Section 2 describes the source of the data and the modeled subset; Section 3 explains how the model parameters were derived; Section 4 specifies the processes modeled; Section 5 presents the results and explores their implications; and section 6 concludes the chapter.

2. Data Source

The dataset used to develop the watershed modeling test case is from the Reynolds Creek Experimental Watershed (RCEW). This is a high-quality long-term dataset that was recently released to the research community; it is available via anonymous ftp: ftp.nwrc.ars.usda.gov, and is maintained by the U.S. Department of Agriculture Agricultural Research Service, Northwest Watershed Research Center in Boise, Idaho, United States (http://www.nwrc.ars.usda.gov). The RCEW dataset covers a 35 year period, from 1962-1996, incorporating a range of variables as summarized in Appendix F.

The following description of the RCEW is a summary of Slaughter *et al.* (2001) and Seyfried *et al.* (2001a), for a detailed description please refer to these papers. The RCEW is 239 km^2, ranging in elevation from 1101m to 2241m above mean sea level. It is located in the Owyhee Mountains of South-western Idaho, United States (Figure 1).

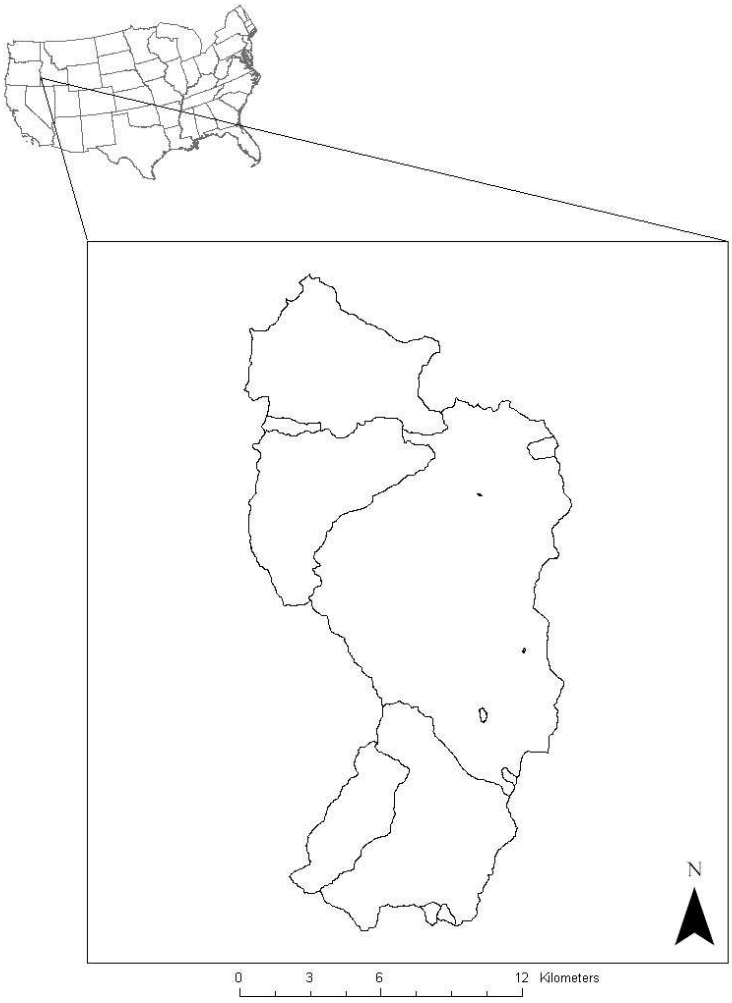


Figure 1. Location of the RCEW

Reynolds Creek, the stream draining the watershed, is a third-order perennial stream that drains north to the Snake River. Approximately 77% of the watershed is under public (federal or state) ownership, with the remainder being privately owned and utilized for livestock grazing with some irrigated fields along the creek at lower elevations. Within the RCEW there is large variation in local climate, geology, soils, and vegetation (see the special issue of Water Resources Research introduced by Marks, 2001 for a full description).

The spatial extent of the RCEW dataset was subset due to computational limitations; it also proved an easier test bed for development. Upper Sheep Creek, a small sub watershed, provided such a subset to test the modeling approach (Figure 2). The primary characteristics of Upper Sheep Creek is a drainage area of 25.9 ha (DEM calculated), an elevation range of 1839-2017m, and an intermittent streamflow regime. It was selected because it is the only small sub-watershed that can be best approximated by a rectangle, necessary due to the current limitations of RePast; it has an intermittent regime rather than ephemeral, therefore it should produce more runoff; and it was included in a study that contains summary statistics on evapotranspiration, which were used in the model (Hanson and Wight, 1995).

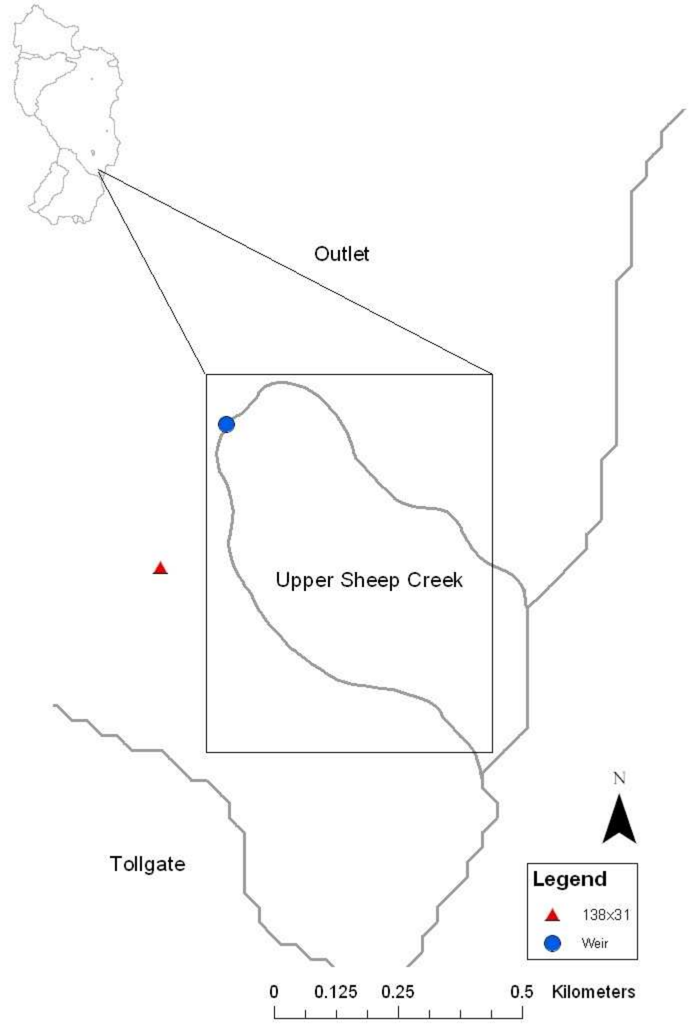


Figure 2. Upper Sheep Creek watershed

The temporal simulation interval was restricted by precipitation records, the availability of streamflow data, and selection of a precipitation event that could be clearly mapped to a discharge event. The precipitation data are continuous records available for 12 sites, 20-32 year records available for 8 sites, 10-19 year records available for 25 years, and 4-9 year records available for 8 sites; a total of 53 sites (Hanson, 2001). The data for precipitation was initially subset to the interval 04/06/1972 – 12/29/1975 in order to capture the full spatial distribution of continuous records, thereby generating the best interpolated surface and minimizing the volume of data for the maximum number of sites. This temporal interval captures 49 of the precipitation measurement sites, excluding sites 138x22, 138x33, 138x44, 098x97 (see Hanson *et al.*, 2001 for site identification). This proved acceptable as these sites were all very close to each other with one used site remaining that represented the area. The data was further subset as the streamflow records ended on 08/06/1975, and an event that clearly registered on both precipitation gauges and discharge measurements was needed for the purposes of comparison with the simulation output. This final selection limited the data to be used in the model from 5/1/1974 - 6/30/1974, as depicted in Figure 3 below. Note that the precipitation follows the discharge slightly as the nearest precipitation measurement site was to the Southwest of the Weir that measured the discharge of Upper Sheep Creek (Site 138x31 in Figure 2), and the precipitation event moved in from the Northeast.

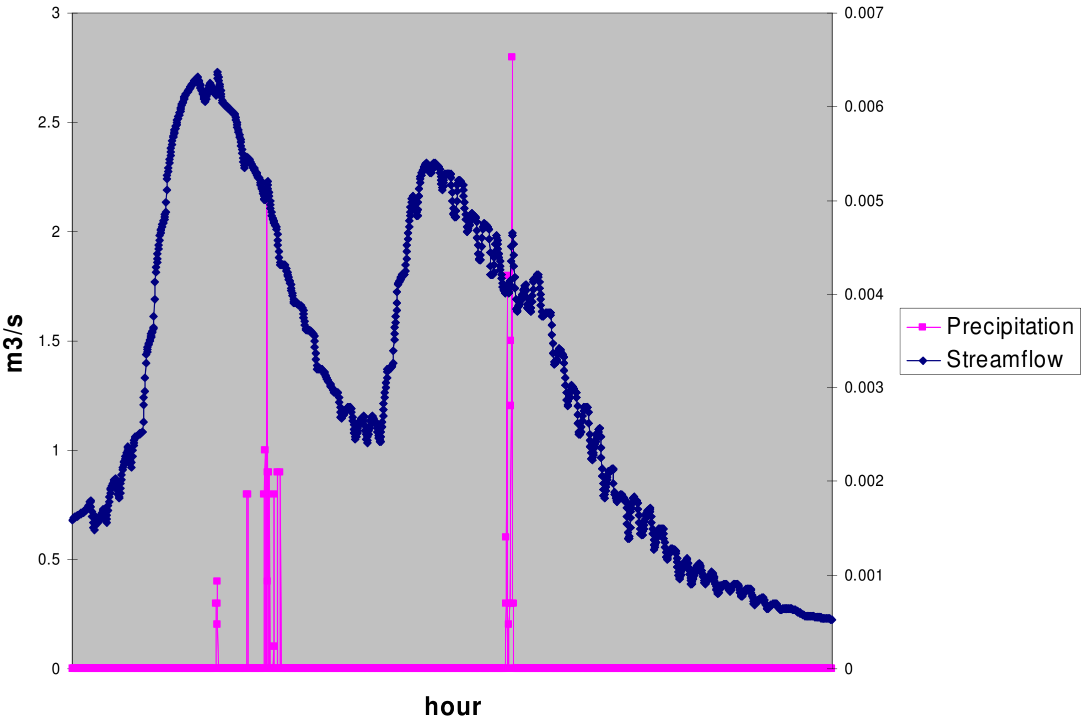


Figure 3. Precipitation, measured at site 138x31, and streamflow for Upper Sheep Creek from 05/1974-06/1974

3. Parameter preparation

Each of the following parameters, barring evapotranspiration, were created for the RCEW as a whole before being clipped to the Upper Sheep Creek watershed.

3.1 Precipitation

There are three measures of precipitation available, unshielded precipitation, shielded precipitation, and computed wind-adjusted precipitation; computed precipitation values were used for the model. Following restructuring of the data, a surface was interpolated for the whole of the RCEW from the 49 points of precipitation. Each hourly set of measurements, within the temporal extent selected, were interpolated with universal kriging, which incorporated the DEM as the single trend component (Pardo-Iguzquiza, 1998). GSTAT[12], available as an extension to R[13], was used to iterate over all of the files and generate an asciigrid of precipitation for RCEW for each time slice (Pebesma and Wesseling, 1998).

3.2 Elevation

The relief of Upper Sheep Creek is modeled by the 30m digital elevation model (DEM), provided in the RCEW dataset, and was converted to an asciigrid for the simulation. Typically digital terrain analysis methods for creating the flow surface prior to modeling include the removal of “spurious” pits in order to create a continuous downward slope. However, it is difficult to determine whether such topographic features are spurious within a 30m squared area, for example, such low points may be sinkholes or dolines. Therefore, they were not removed as the process of surface ponding was included in the model which can handle the scenario of water accumulating in such depressions.

3.3 Bedrock

The soils data in the RCEW dataset included a field describing the depth to bedrock, which was used to generate a bedrock surface. However many of the values are

[12] http://www.gstat.org

[13] http://www.r-project.org/

unknown as they are deeper than investigated. In these cases a value of –x is given, meaning the bedrock is deeper than the specified value x. As a temporary fix, 5m were added to these absolute values. A bedrock layer was created by subtracting these depths from the DEM.

3.4 Evapotranspiration

Values for evapotranspiration in the Upper Sheep Creek were determined from a paper by Hanson and Wright (Hanson and Wight, 1995). This provided a simple solution for defining evapotranspiration rather than calculating it by the Pennman-Monteith equation. They divided Upper Sheep Creek into two parts A and B, based on two types of vegetation, Grass-Low sagebrush and Grass-Mountain big sagebrush. Based on the vegetation layers, the two values of evapotranspiration were assigned to the different parts of Upper Sheep Creek (A and B).

3.5 Saturated Hydraulic Conductivity

Using data from Rawls *et al.*(1982), definitions from the Soil Science Glossary provided by the Soil Science Society of America[14], and corresponding data in the RCEW dataset, saturated hydraulic conductivity was approximated from the soil texture class. Unfortunately Rawls *et al.*(1982) did not include silt in their categorization, thus it was

[14] http://www.soils.org/sssagloss/

approximated as being between silt loam and sandy clay loam, that is, as equal to 0.56 cm/h.

3.6 Infiltration Capacity

The soil moisture data was measured at five sites, only three of which fall within the selected precipitation time frame. Three points are not enough to generate a surface therefore this data can only be used to calibrate or test the model. The Soil Hydrologic Group data was used to specify infiltration capacity, which is the National Resource Conservation Service classification for estimating overland flow[15]. As defined in the National Engineering Handbook (NEH-4), each hydraulic soil group is associated with an infiltration capacity. An added class was specified for the case where no infiltration could take place, such as on rock terrace escarpments, namely class E of 0 mm/hr.

3.7 Watertable

The watertable was created purely for the purposes of the model application, and is not expected to accurately represent the watertable in RCEW as it is unknown and the data is not available. The generated watertable took the streams within RCEW as its base, such that the cells at the location of perennial streams were assigned a value of 0 meters below the DEM, and the intermittent stream cells were assigned a value of 1 meter below the DEM. All other cells were assigned a value based on an increasing function

[15] See the National Soil Survey Handbook produced by the National Resource Conservation Service (NRCS) - http://soils.usda.gov/technical/handbook/download.html

based on distance from these cells with a script written in Java using the Jama library[16], a Java matrix library. This layer of values was then subtracted from the DEM, and the maximum value of this layer and the bedrock layer was taken as the watertable in order to assure that the water table was always above the bedrock layer.

4. Process Specification

In what follows, an outline of the behavior of each process represented within the model will be described in pseudo code. The processes included are infiltration (I), percolation (P), groundwater flow (GF), Hortonian overland runoff (HO), saturation excess runoff (SE), and surface ponding (SP). The spatial extent of all processes is defined by the DEM, that is, by the selected rectangular area that represents Upper Sheep Creek. The spatial granularity of each process is also defined by the DEM, where each process operates over a 30m² area. The temporal extent of each process is defined by the model extent, that is, from 5/1/1974 - 6/30/1974. The temporal granularity of the processes is a function of the forcing parameters, which in the case of this application is the hourly update of precipitation.

Any lateral movement in the x and y direction, whether it be above or below surface, is defined by the D8 method, whereby the minimum elevation value in the 8 cell neighborhood of a cell is taken as the direction of flow. Although the D8 algorithm has been assessed as a poor descriptor of the spatial distribution of flow (Endreny and

[16] http://math.nist.gov/javanumerics/jama/

Wood, 2003), it was used in the model as it provided the simplest approach for implementation.

Following a precipitation event, processes are created as follows:

```
if (precipitation – evapotranspiration > infiltration capacity)
        if (there is a neighboring point of lower elevation)
                create HO
        else create SP
else    if (watertable is the same elevation as DEM)
                create SE
        else create I
```

4.1 Infiltration (I)

The infiltration process converts directly to a percolation process at the following time step.

4.2 Percolation (P)

Percolation processes are generated following infiltration and result in water flowing down in the z direction through the soil matrix towards the watertable. The rate of downward flow is defined by the hydraulic conductivity parameter, and in the x and y direction according to the DEM surface.

```
if (the watertable has not been reached)
    if (there is a lower neighboring elevation)
        percolate in a direction depending on the surface slope at a rate dependent
        on the hydraulic conductivity of the soil and the mass of water.
    else    percolate straight down at a rate dependent on the hydraulic conductivity
            of the soil and the mass of water
else    convert to G
```

4.3 Groundwater Flow (GF)

Groundwater flow occurs once percolation has reached the water table.

```
if (there is a lower neighboring cell based on watertable elevation)
    if (watertable >= DEM elevation)
        create SE
    else    continue flowing in direction of lowest watertable elevation
else    if (watertable >= DEM elevation)
            create SE
        else    add to the water table by elevating it
```

4.4 Hortonian Overland Runoff (HO)

Hortonian overland runoff is generated when the rate of precipitation exceeds the infiltration capacity of the soil.

```
if (HO mass > infiltration capacity)
    if(there is a lower neighboring cell in DEM)
        continue HO runoff in direction of lowest neighbor
    else    create SP
else    if (watertable >= DEM elevation)
        if(there is a lower neighboring cell in DEM)
            create SE
        else    create SP
    else    create I
```

4.5 Saturation Excess Runoff (SE)

Saturation excess runoff is generated when under precipitation the watertable is equal to or exceeds the elevation of the DEM.

```
if (SE mass > infiltration capacity OR watertable >= DEM elevation)
    if(there is a lower neighboring cell in DEM)
        continue SE runoff in direction of lowest neighbor
    else    create SP
else    create I
```

4.6 Surface Ponding (SP)

Surface ponding results when precipitation less evapotranspiration is greater than infiltration capacity and there is no neighboring cell of lower elevation.

```
if(watertable elevation >= DEM elevation)
        continue SP
else    create I
```

5. Results and Discussion

The specification presented above in Sections 3 and 4 outlines the model that was implemented within the flux modeling framework (described in Section II). The simulation of the model produced text files for each process class, which store information on the dynamics of each process instance.

5.1 Hydrograph Simulation and Extensions

The most obvious result to provide is the hydrograph, which maps to the output of traditional modeling approaches. This is possible within the methodology developed as it captures both state and process information. Unfortunately, reproduction of watershed discharge over time is not particularly difficult, nor does it imply that the processes in the model have been adequately modeled (Bevan, 2000: 218). Figure 4 below is the initial hydrograph of the simulation results that is generated using the query tools described in Part II. As is clearly evident when comparing it against the measured hydrograph depicted in Figure 3 above, the modeled output closely follows the precipitation pattern, but does not reflect the measured discharge very well at all. There are two key reasons for this problem; first, the model does not take into account baseflow that results from spring snowmelt; second, the spatial resolution of the model strongly influences the rate of discharge over time, where at each hourly time step a process occurs over a 30m grid cell.

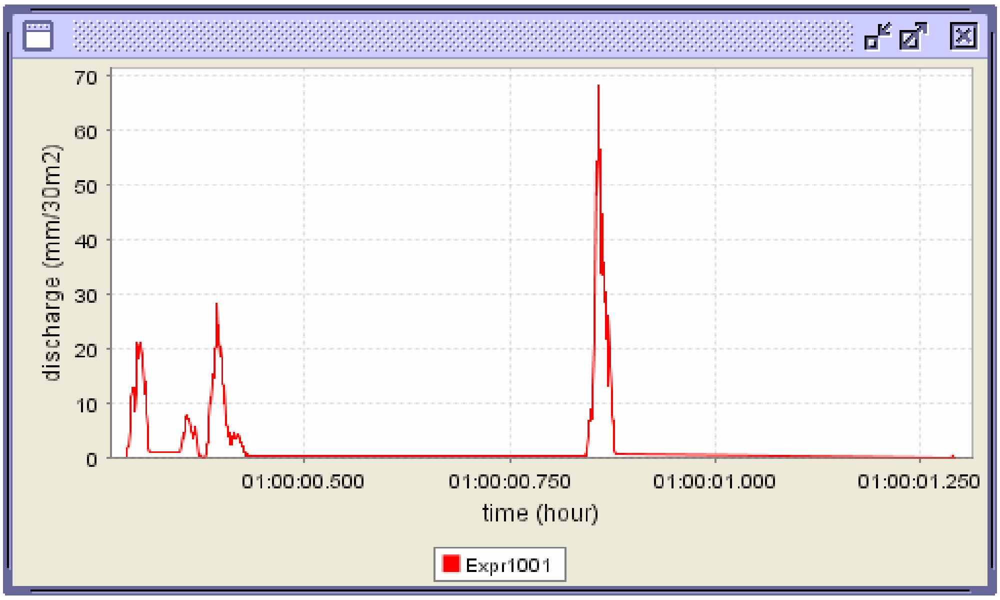


Figure 4. Simulated hydrograph of Upper Sheep Creek 5/1/1974 - 6/30/1974

The spatial resolution is predefined by the 30m DEM. As with other models such as TOPMODEL (Quinn *et al.*, 1995), grid resolution has implications for spatial predictions. In future this might be solved by re-sampling all of the grid layers to a finer spatial resolution, which will aid in correcting the timing of discharge. An alternative temporary solution was attempted that slows the groundwater flow down by inserting a timer function. One of the more promising results from this experiment is shown in Figure 5 below, which presents a pattern of discharge slightly closer to the measured output.

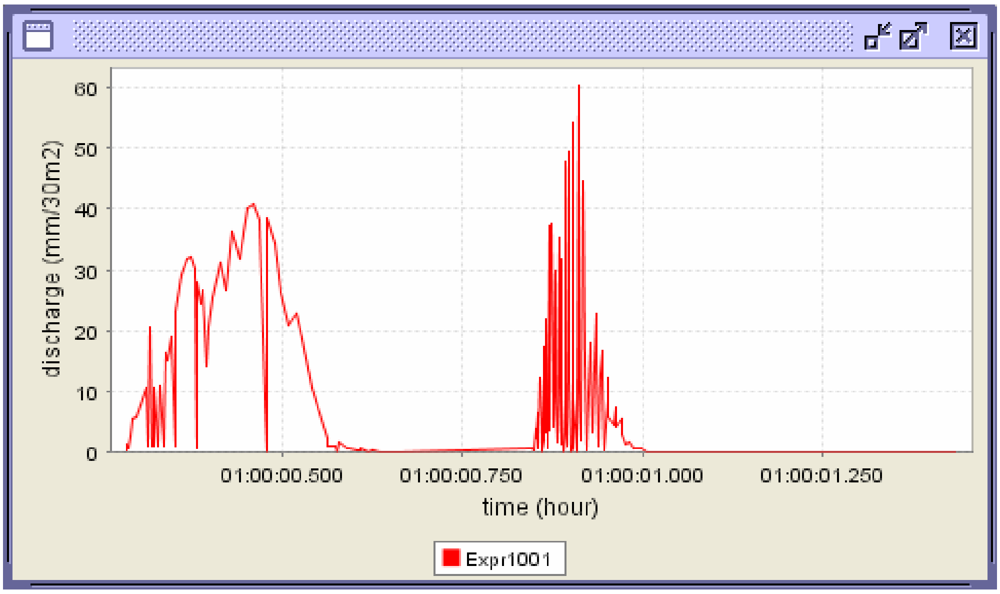


Figure 5. Tweaked hydrograph of Upper Sheep Creek 5/1/1974 - 6/30/1974

However, poor spatial resolution also results in the loss of detail in the variation in spatial attributes such as elevation, saturated hydraulic conductivity, and precipitation. Consequently, processes at finer spatial resolutions are not modeled.

5.2 Exploring the Results

The advantage of the process data model, however, is that we can move beyond the hydrograph as our main form of validation and start exploring how the processes defined in our model are expressed at runtime; what their spatial, temporal, and attribute characteristics are. The first and most obvious result to consider is the spatial dynamics of the modeled processes. Figure 6 below presents two process time slices, displaying the spatial distribution of the processes at hour 402 and hour 403 over the DEM. The green nens represent groundwater flow, blue nens: Hortonian overland runoff, orange nens: percolation, and yellow nens: infiltration. This display allows users to compare their process descriptions in the model with qualitative knowledge of where those processes occur in reality.

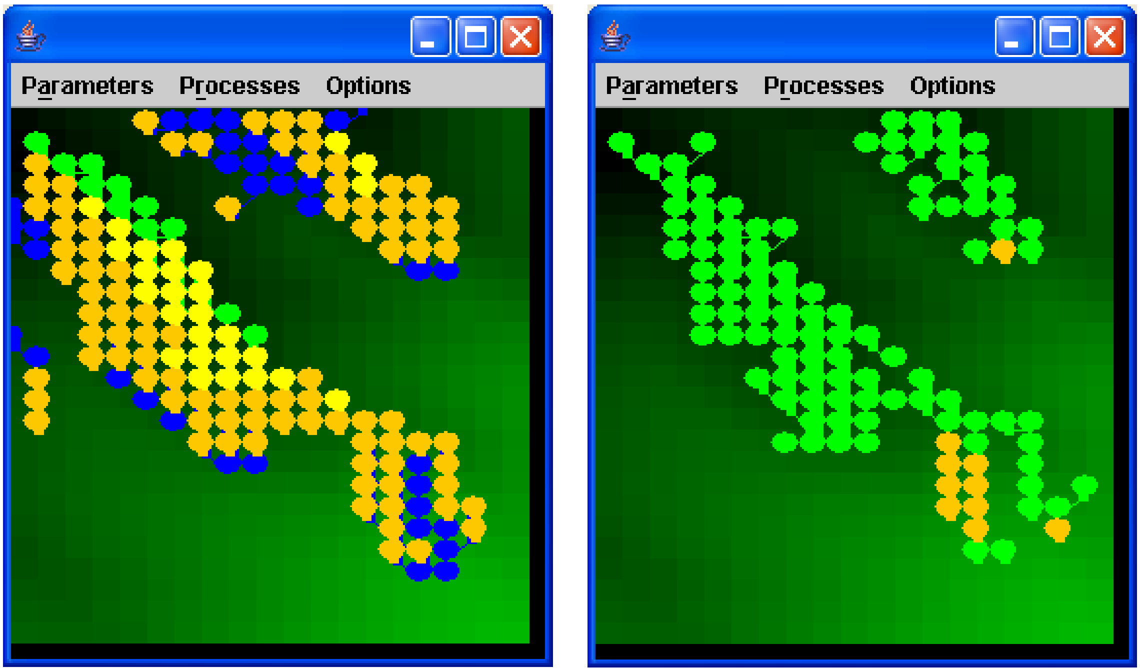


Figure 6. Simulation display output for hour 402 and hour 412

Beyond the spatial qualities of the results, any other aspect of the process that is stored in the data structure may be queried for. For example, the query tool created for the flux modeling framework also allows for direction based querying; however, this would perhaps be more useful for other types of processes such as atmospheric processes. Although not yet implemented, in a model that incorporates the interaction of processes at different scales, representing the process information will allow for novel queries such as selecting spatio-temporally coincident or interacting processes or tracing the dynamics of individual process instances. The results of such queries will form the basis for further analysis.

In terms of the original theoretical conceptualization of Chapter 3, process information exploration can be classed as querying for temporal, spatial, and spatio-temporal parts. In the case of the application presented in this chapter, a temporal part of the overall process of watershed runoff might include querying for the initial response of the watershed to precipitation, such as within the first hour. Queries for spatial parts might include selecting processes that occur along a riparian zone or other areas of interest such as agricultural areas within the watershed, as demonstrated by the spatial query tool extension described in Part II. Spatio-temporal parts are the individual processes, such as Hortonian overland flow or percolation, which make up the overall process of watershed runoff. Alternatively, we might consider the traditional components that are extracted via methods of hydrograph separation, namely baseflow and stormflow that can be approximated as groundwater flow and a mix of Hortonian overland flow and saturation excess flow respectively. Each of these three types of queries can be augmented with attribute qualifiers.

5.3 Validation

Usually validation occurs by matching the output of the model with the real world, a good result being the ability to mirror that world *in silico*. For example, Endreny and Wood qualitatively validate their simulated flow networks with empirical data (2003). The standard approach to validation in watershed modeling is to compare the simulated output of volume of stream discharge over time, with discharge measurements over the same time period for the modeled watershed.

In validating and fully testing the model results described above, the central problem is that long term empirical observations are not available for describing the location and duration of processes. Some of the literature covering RCEW does provide limited discussion on the processes operating in certain parts of the watershed, however this is not enough for model validation. Without such real world data, any model developed with the process based methodology created as part of this dissertation cannot be effectively validated. This can be defined as a form of process modeling equifinality, where the same system state can result from many different process pathways, which is well recognized by watershed modelers as a problem of validating against hydrographs (Bevan, 2000). A possible solution would then be to validate the model against another model of similar nature, yet no such model exists.

As such, the author must leave validation of a fully specified domain model to a future research objective. This would involve intensive study of a particular watershed and the development of appropriate measurement methods that either standardize qualitative descriptions or propose new process based measurement approaches.

6. Concluding Part III

This chapter has attempted to clearly specify an application process model, which was then implemented within the flux modeling environment described in Section II. The implementation allowed for testing of the theory and methodology that form the core of this dissertation. The results of the test, while hydrologically inconclusive, proved the point of the methodology and validated the theoretical approach of taking process as the modeling primitive. This allowed for the application of process oriented queries, where the state of the process at an instant of time was queried for initial exploration of the model results. It also provides the basis for further development of the query tools such that the dynamics or change of process instants can be queried, and the future innovation into analytical techniques.

In terms of the watershed modeling application, future extensions would include incorporating better definitions of the processes based on expert knowledge. There are a slew of processes that have been ignored for the sake of modeling simplicity, such as the disregard of channel flow processes and erosion. Furthermore, choosing a watershed that the author and hydrological experts can actively explore and qualitatively compare the processes found in the real world with those found in the model would permit the validation of a domain model.

CHAPTER TEN

Conclusion

1. Rewind

This dissertation has presented an alternative theoretical and methodological approach to modeling geographic processes. The theory, in particular the notion of process as primitive, provided the basis for a conceptualization and modeling methodology. This methodology involved developing the modeling framework called flux, which extended current software to operate with a new process oriented data model. The flux simulator provides the first steps towards querying, analyzing, and exploring process definitions and the causal interactions of processes.

Using this flux framework, an application has been developed for watershed modeling that applies the theory and implements the methodology. This supplies proof that the theory and methodology work and produce novel and useful results. In particular, the methodology developed in this dissertation allows for the query and analysis of processes. Beyond the running model, basic tools have been developed that allow for querying the process data structure.

Although slightly off the topic of the original objectives expressed in Chapter 1, a prototype application that converts an ontology to a running model within the flux framework was also developed. This is the first step towards ontology based modeling that the author is aware of; it provides direction towards different applications utilizing the same model information in new ways and for interoperability.

2. Fast Forward

The immediate extension of the dissertation (beyond a few quick publications) is to scale the application from the Upper Sheep Creek watershed to the full Reynolds Creek Experimental Watershed (RCEW). Unfortunately recent attempts at doing so have hit computing limitations. Furthermore, this work will be expanded to incorporate processes and parameters at different spatio-temporal scales. In particular, how processes can be modeled at various scales and their cross scale interactions encoded will be explored; as expressed by Bauer *et al.* "our goal may be to produce methodologies that allow an interpretable, comprehensive representation across all spatial and temporal scales that is somehow simpler and more compelling than representation that includes all the separate components" (Bauer *et al.*, 1999: 686).

The next area of further development is the extension of the query tools to more novel queries that can only be considered with the process data model, and the development of analysis techniques. The analytical techniques to be created must recognize the spatio-temporal nature of the process primitive, and may also reuse current techniques in new ways, such as those used for vector fields (Li and Hodgson, 2004).

The third area of future work is the further development of the ontology driven process model. This involves the development of multiple process ontologies and experimentation with new ways of using the information expressed in the model individual. For example, process model ontologies could be used for automatic model interoperation and model component discovery via a future semantic grid.

Many questions burst out from this work, which might potentially form the basis for further areas of research. Do processes self organize? Can we develop genetic algorithms for process rules? Are there other data models that can be developed in order to better represent other aspects of the world? If we can model processes differently can we also measure them differently? Do we need new process observation tools rather than state observation tools?

3. Stop

But before all this gets out of hand, here ends the dissertation.

APPENDIX A

Flux Ontology

```
<!DOCTYPE rdf:RDF [
   <!ENTITY xsd "http://www.w3.org/2000/10/XMLSchema#">
   <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#">
   <!ENTITY oml "http://www.glue.umd.edu/~femke/processModel/oml#" >
]>

<rdf:RDF
    xmlns        ="http://www.glue.umd.edu/~femke/processModel/flux#"
    xmlns:base   ="http://www.glue.umd.edu/~femke/ProcessModel/flux#"
    xmlns:flux   ="http://www.glue.umd.edu/~femke/processModel/flux#"
    xmlns:method ="http://www.glue.umd.edu/~femke/processModel/method#"
    xmlns:owl    ="http://www.w3.org/2002/07/owl#"
    xmlns:rdf    ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs   ="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:xsd    ="http://www.w3.org/2000/10/XMLSchema#"
    xmlns:time   ="http://www.isi.edu/~pan/damltime/time.owl"
    xmlns:swrl
      ="http://www.iswc2003.semanticweb.org/rules/proposal/swrl.owl"
>

<owl:Ontology rdf:about="">
  <rdfs:comment>The process ontology.</rdfs:comment>
  <rdfs:label>flux ontology</rdfs:label>
</owl:Ontology>

<!-- CLASSES -->


<owl:Class rdf:ID="Process">
      <rdfs:comment>The single primitive</rdfs:comment>
      <rdfs:subClassOf>
            <owl:Restriction owl:cardinality="1">
                  <owl:onProperty rdf:resource="&rdfs;label"/>
            </owl:Restriction>
      </rdfs:subClassOf>
</owl:Class>



<!--  PROPERTIES -->

<owl:DatatypeProperty rdf:ID="spatialExtent">
      <rdfs:domain rdf:resource="#Process"/>
      <rdfs:domain rdf:resource="#Parameter"/>
      <rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="temporalExtent">
      <rdfs:domain rdf:resource="#Process"/>
      <rdfs:domain rdf:resource="#Parameter"/>
      <rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="spatialGrain">
      <rdfs:domain rdf:resource="#Process"/>
      <rdfs:domain rdf:resource="#Parameter"/>
      <rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="temporalGrain">
      <rdfs:domain rdf:resource="#Parameter"/>
      <rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="value">
      <rdfs:domain rdf:resource="#Process"/>
      <rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="x1">
      <rdfs:domain rdf:resource="#Process"/>
      <rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="x2">
      <rdfs:domain rdf:resource="#Process"/>
      <rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="y1">
      <rdfs:domain rdf:resource="#Process"/>
      <rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="y2">
      <rdfs:domain rdf:resource="#Process"/>
      <rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="z1">
      <rdfs:domain rdf:resource="#Process"/>
      <rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="z2">
      <rdfs:domain rdf:resource="#Process"/>
      <rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>



<!-- RULES -->


<owl:Impl rdf:ID="CreateProcessRule">
      <rdfs:subClassOf>
```

```
            <owl:Restriction owl:cardinality="1">
                  <owl:onProperty rdf:resource="&rdfs;label"/>
            </owl:Restriction>
      </rdfs:subClassOf>
</owl:Impl>

<owl:Impl rdf:ID="ChangeProcessRule">
      <rdfs:subClassOf>
            <owl:Restriction owl:cardinality="1">
                  <owl:onProperty rdf:resource="&rdfs;label"/>
            </owl:Restriction>
      </rdfs:subClassOf>
</owl:Impl>

<owl:Impl rdf:ID="DestroyProcessRule">
      <rdfs:subClassOf>
            <owl:Restriction owl:cardinality="1">
                  <owl:onProperty rdf:resource="&rdfs;label"/>
            </owl:Restriction>
      </rdfs:subClassOf>
</owl:Impl>


<!-- METHODS -->

<method:MethodObjectProperty rdf:ID="createProcess">
      <rdfs:label>newProcess</rdfs:label>
      <rdfs:range rdf:resource="#Process"/>
</method:MethodObjectProperty>

<method:MethodObjectProperty rdf:ID="changeProcess">
      <rdfs:label>changeProcess</rdfs:label>
      <rdfs:range rdf:resource="#Process"/>
</method:MethodObjectProperty>

<method:MethodObjectProperty rdf:ID="destroyProcess">
      <rdfs:label>destroyProcess</rdfs:label>
      <rdfs:range rdf:resource="#Process"/>
</method:MethodObjectProperty>



</rdf:RDF>
```

APPENDIX B

Method Ontology

```
<!DOCTYPE rdf:RDF [
   <!ENTITY xsd   "http://www.w3.org/2000/10/XMLSchema#">
   <!ENTITY rdfs  "http://www.w3.org/2000/01/rdf-schema#">
   <!ENTITY owl   "http://www.w3.org/2002/07/owl#">
   <!ENTITY rdf   "http://www.w3.org/1999/02/22-rdf-syntax-ns#">
   <!ENTITY swrl  "http://www.w3.org/2003/11/swrl#">
]>


<rdf:RDF
   xmlns =        "http://www.glue.umd.edu/~femke/processModel/method#"
   xml:base =     "http://www.glue.umd.edu/~femke/ProcessModel/method#"
   xmlns:method = "http://www.glue.umd.edu/~femke/processModel/method#"
   xmlns:rdf =    "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:rdfs =   "http://www.w3.org/2000/01/rdf-schema#"
   xmlns:owl =    "http://www.w3.org/2002/07/owl#"
>

<owl:Ontology rdf:about="">
      <rdfs:comment>
            Author: Femke Reitsma
            Last Modified: 20/09/04
      </rdfs:comment>
      <rdfs:label>Method</rdfs:label>
</owl:Ontology>


<owl:Class rdf:about="#MethodAtom">
      <rdfs:isDefinedBy
rdf:resource="http://www.glue.umd.edu/~femke/processModel/method#"/>
      <rdfs:label>MethodAtom</rdfs:label>
      <rdfs:comment>A method atom for a rule</rdfs:comment>
      <rdfs:subClassOf rdf:resource="&swrl;Atom"/>
</owl:Class>

<owl:Class rdf:about="#MethodProperty">
      <rdfs:isDefinedBy
rdf:resource="http://www.glue.umd.edu/~femke/processModel/method#"/>
      <rdfs:label>MethodProperty</rdfs:label>
      <rdfs:comment>The class of Method Properties.  These types of
      methods are defined on a class, in the sense of being a property
      of an object that can do something. </rdfs:comment>
      <rdfs:subClassOf rdf:resource="&rdf;Property"/>
</owl:Class>

<owl:Class rdf:about="#MethodObjectProperty">
      <rdfs:isDefinedBy
rdf:resource="http://www.glue.umd.edu/~femke/processModel/method#"/>
      <rdfs:label>methodObjectProperty</rdfs:label>
```

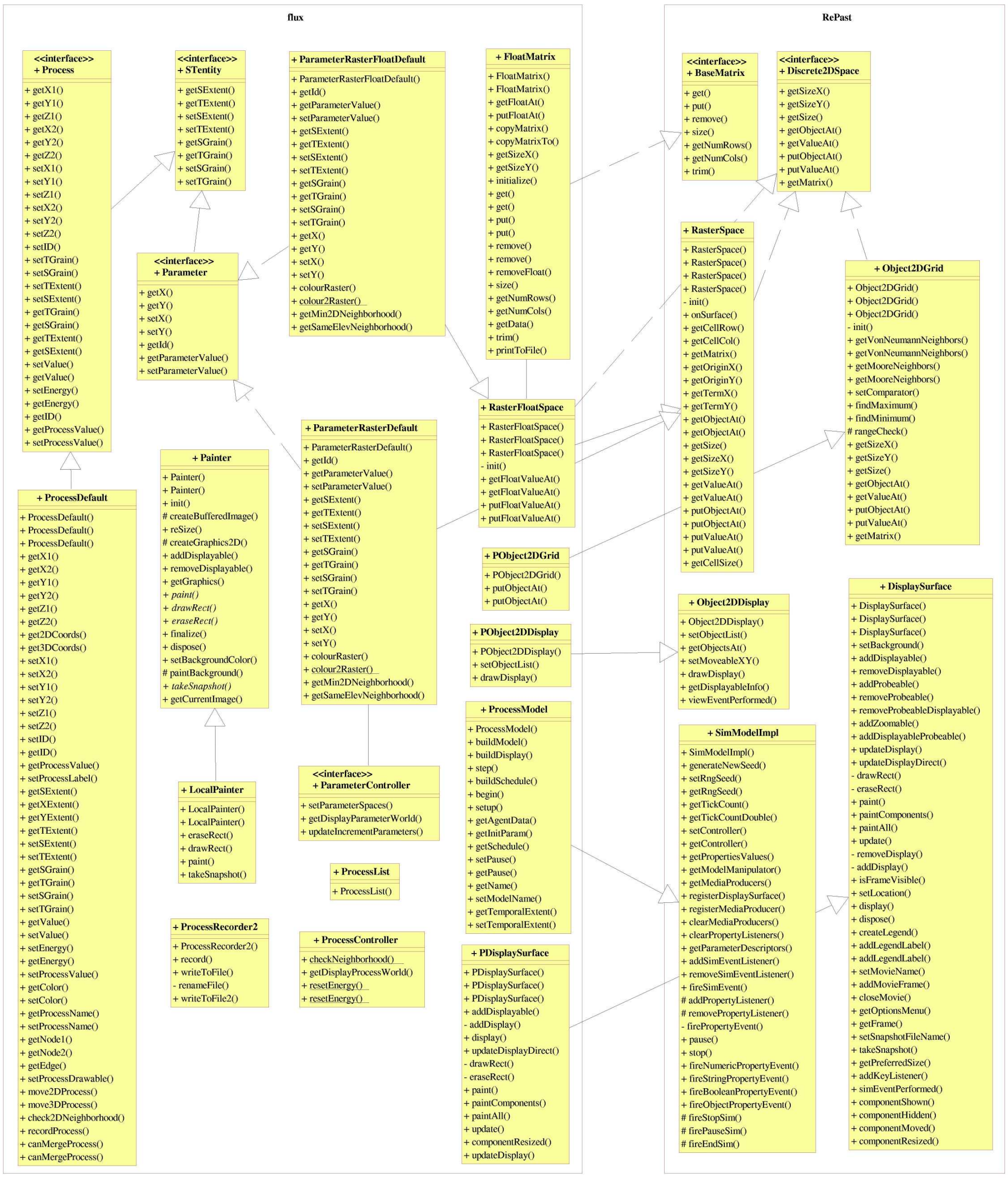
```
       <rdfs:comment>The property that indicates that it requires a
      method to determine the object range</rdfs:comment>
      <rdfs:subClassOf rdf:resource="#MethodProperty"/>
</owl:Class>

<owl:Class rdf:about="#MethodDatatypeProperty">
      <rdfs:isDefinedBy
rdf:resource="http://www.glue.umd.edu/~femke/processModel/method#"/>
      <rdfs:label>methodDatatypeProperty</rdfs:label>
       <rdfs:comment>The property that indicates that it requires a
      method to determine the datatype range</rdfs:comment>
      <rdfs:subClassOf rdf:resource="#MethodProperty"/>
</owl:Class>


</rdf:RDF>
```

APPENDIX C

Class Diagram of Flux Modeling Package



APPENDIX D

Flux Implementation Ontology

```
<!DOCTYPE rdf:RDF [
   <!ENTITY xsd   "http://www.w3.org/2000/10/XMLSchema#">
   <!ENTITY rdfs  "http://www.w3.org/2000/01/rdf-schema#">
   <!ENTITY method
"file:///C:/workspace/eclipseWorkspace/flux/input/ontologies/method#" >
   <!ENTITY swrl  "http://www.w3.org/2003/11/swrl#">
]>


<rdf:RDF
xmlns
="file:///C:/workspace/eclipseWorkspace/flux/input/ontologies/flux#"
xml:base
="file:///C:/workspace/eclipseWorkspace/flux/input/ontologies/flux#"
xmlns:flux
="file:///C:/workspace/eclipseWorkspace/flux/input/ontologies/flux#"
xmlns:method
="file:///C:/workspace/eclipseWorkspace/flux/input/ontologies/method#"
    xmlns:owl     ="http://www.w3.org/2002/07/owl#"
    xmlns:rdf     ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs    ="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:xsd     ="http://www.w3.org/2000/10/XMLSchema#"
    xmlns:time    ="http://www.isi.edu/~pan/damltime/time.owl"
    xmlns:swrl    ="http://www.w3.org/2003/11/swrl#"
    xmlns:swrlb   ="http://www.w3.org/2003/11/swrlb#"
>


<owl:Ontology rdf:about="">
  <rdfs:comment>The process modeling ontology.  This provides the base
set of classes that a user must implement</rdfs:comment>
  <rdfs:label>A Process Modeling Ontology</rdfs:label>
</owl:Ontology>

<!--  PROPERTIES -->

<!-- each property can be converted to a get and set method in java -->

<owl:DatatypeProperty rdf:ID="datavalue">
      <rdfs:label>dataValue</rdfs:label>
      <rdfs:domain rdf:resource="&swrl;Variable"/>
      <rdfs:range rdf:resource="&xsd;integer"/>
      <rdfs:comment>The datavalue property is the datavalue assigned to
a SWRL variable
      in a SWRL rule</rdfs:comment>
</owl:DatatypeProperty>

<owl:ObjectProperty rdf:ID="individual">
      <rdfs:label>individual</rdfs:label>
      <rdfs:domain rdf:resource="&swrl;Variable"/>
      <rdfs:range rdf:resource="#Process"/>
      <rdfs:comment>The individual property is the individual assigned
to a SWRL variable
      in a SWRL rule</rdfs:comment>
</owl:ObjectProperty>


<owl:DatatypeProperty rdf:ID="spatialExtent">
      <rdfs:domain rdf:resource="#Process"/>
      <rdfs:domain rdf:resource="#Parameter"/>
      <rdfs:range rdf:resource="#Parameter"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="temporalExtent">
      <rdfs:domain rdf:resource="#Process"/>
      <rdfs:domain rdf:resource="#Parameter"/>
      <rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="spatialGrain">
      <rdfs:domain rdf:resource="#Process"/>
      <rdfs:domain rdf:resource="#Parameter"/>
      <rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="temporalGrain">
      <rdfs:domain rdf:resource="#Parameter"/>
      <rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="modelTemporalExtent">
      <rdfs:subPropertyOf rdf:resource="#temporalExtent"/>
      <rdfs:domain rdf:resource="#ProcessModel"/>
      <rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="color">
      <rdfs:domain rdf:resource="#Process"/>
      <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="energy">
      <rdfs:domain rdf:resource="#Process"/>
      <rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="startTime">
      <rdfs:domain rdf:resource="#Parameter"/>
      <rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="inputFilePath">
      <rdfs:label>inputFilePath</rdfs:label>
      <rdfs:comment>The full path of the input file for static
parameters</rdfs:comment>
```

```
      <rdfs:domain rdf:resource="#StaticParameter"/>
      <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="incrementFileStart">
      <rdfs:label>incrementFileStart</rdfs:label>
      <rdfs:comment>The string that is the first part of the file name,
the second part is a number that increments</rdfs:comment>
      <rdfs:domain rdf:resource="#IncrementParameter"/>
      <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="inputFileDirectory">
      <rdfs:label>inputFileDirectory</rdfs:label>
      <rdfs:comment>The directory of the input files for increment
parameters</rdfs:comment>
      <rdfs:domain rdf:resource="#IncrementParameter"/>
      <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="constant">
      <rdfs:label>constant</rdfs:label>
      <rdfs:domain rdf:resource="#Parameter"/>
      <rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<owl:ObjectProperty rdf:ID="displayParameter">
      <rdfs:label>displayParameter</rdfs:label>
      <rdfs:comment>This property defines the parameter that will be
displayed in the model as the base layer</rdfs:comment>
      <rdfs:domain rdf:resource="#ProcessModel"/>
      <rdfs:range rdf:resource="#Parameter"/>
</owl:ObjectProperty>

<owl:DatatypeProperty rdf:ID="modelName">
      <rdfs:label>modelName</rdfs:label>
      <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

<owl:ObjectProperty rdf:ID="hasProcess">
      <rdfs:label>hasProcess</rdfs:label>
      <rdfs:domain rdf:resource="#ProcessModel"/>
      <rdfs:range rdf:resource="#Process"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="hasParameter">
      <rdfs:label>hasParameter</rdfs:label>
      <rdfs:domain rdf:resource="#ProcessModel"/>
      <rdfs:range rdf:resource="#Parameter"/>
</owl:ObjectProperty>

<owl:DatatypeProperty rdf:ID="hasExtent">
      <rdfs:label>hasExtent</rdfs:label>
      <rdfs:comment>This defines the spatial extent of the model.  All
parameters are assumed to have the same extent or less than that
defined here</rdfs:comment>
```

```
      <rdfs:domain rdf:resource="#ProcessModel"/>
      <rdfs:range rdf:resource="#Parameter"/>
</owl:DatatypeProperty>

<method:DatatypeProperty rdf:ID="minMooreNeighborX">
      <rdfs:label>minMooreNeighborX</rdfs:label>
      <rdfs:domain rdf:resource="#MinMooreNeighbor"/>
      <rdfs:range rdf:resource="&xsd;integer"/>
</method:DatatypeProperty>

<method:DatatypeProperty rdf:ID="minMooreNeighborY">
      <rdfs:label>minMooreNeighborY</rdfs:label>
      <rdfs:domain rdf:resource="#MinMooreNeighbor"/>
      <rdfs:range rdf:resource="&xsd;integer"/>
</method:DatatypeProperty>

<method:DatatypeProperty rdf:ID="minMooreNeighborXYValue">
      <rdfs:label>minMooreNeighborXYValue</rdfs:label>
      <rdfs:comment>specifies the value found at a specific x y
location</rdfs:comment>
      <rdfs:domain rdf:resource="#MinMooreNeighbor"/>
      <rdfs:range rdf:resource="&swrl;Variable"/>
</method:DatatypeProperty>

<!-- METHOD PROPERTIES -->

<method:MethodOjectProperty rdf:ID="hasMinMooreNeighbor">
      <rdfs:label>hasMinMooreNeighbor</rdfs:label>
      <rdfs:domain rdf:resource="#Process"/>
      <rdfs:range rdf:resource="#MinMooreNeighbor"/>
</method:MethodOjectProperty>

<method:MethodDatatypeProperty rdf:ID="parameterXYValue">
      <rdfs:label>parameterXYValue</rdfs:label>
      <rdfs:comment>specifies the value found at a specific x y
location</rdfs:comment>
      <rdfs:domain rdf:resource="#Parameter"/>
      <rdfs:range rdf:resource="&swrl;Variable"/>
</method:MethodDatatypeProperty>

<method:MethodDatatypeProperty rdf:ID="processX2Y2Value">
      <rdfs:label>processX2Y2Value</rdfs:label>
      <rdfs:comment>specifies the value found at a specific x y
location</rdfs:comment>
      <rdfs:domain rdf:resource="#Process"/>
      <rdfs:range rdf:resource="&swrl;Variable"/>
</method:MethodDatatypeProperty>

<method:MethodOjectProperty rdf:ID="move2DProcessTo">
      <rdfs:label>move2DProcessTo</rdfs:label>
      <rdfs:comment>Moves a process instance a lower neighbor in its
MooreNeighborhood</rdfs:comment>
      <rdfs:domain rdf:resource="#Process"/>
      <rdfs:range rdf:resource="#MinMooreNeighbor"/>
</method:MethodOjectProperty>
```

```
<!-- CLASSES -->

<method:Method rdf:ID="MinMooreNeighbor">
      <rdfs:label>minMooreNeighbor</rdfs:label>
      <rdfs:comment>A predefined method in flux that finds the x, y,
and value of the lowest neighbor in the Moore neighborhood of a
location</rdfs:comment>
</method:Method>

<owl:Class rdf:ID="Paramater">
      <rdfs:label>Parameter</rdfs:label>
      <rdfs:comment>A Parameter is a process that we do not model
because either it is outside of the scope of our model and thus simply
forms the input to our model or because the temporal granularity of the
process is greater than the temporal extent of the model</rdfs:comment>
      <rdfs:subClassOf>
            <owl:Restriction owl:cardinality="1">
                  <owl:onProperty rdf:resource="&rdfs;label"/>
            </owl:Restriction>
      </rdfs:subClassOf>
      <rdfs:subClassOf>
            <owl:Restriction owl:cardinality="1">
                  <owl:onProperty rdf:resource="#temporalGrain"/>
            </owl:Restriction>
      </rdfs:subClassOf>
      <rdfs:subClassOf>
            <owl:Restriction owl:cardinality="1">
                  <owl:onProperty rdf:resource="#temporalExtent"/>
            </owl:Restriction>
      </rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:ID="IncrementParameter">
      <rdfs:label>IncrementParameter</rdfs:label>
      <rdfs:comment>The increment parameter is a parameter that is
updated during the model according to a schedule defined by the
temporal granularity</rdfs:comment>
      <rdfs:subClassOf rdf:resource="Parameter"/>
      <rdfs:subClassOf>
            <owl:Restriction owl:cardinality="1">
                  <owl:onProperty rdf:resource="#temporalGrain"/>
            </owl:Restriction>
      </rdfs:subClassOf>
      <rdfs:subClassOf>
            <owl:Restriction owl:cardinality="1">
                  <owl:onProperty rdf:resource="#inputFileDirectory"/>
            </owl:Restriction>
      </rdfs:subClassOf>
      <rdfs:subClassOf>
            <owl:Restriction owl:cardinality="1">
                  <owl:onProperty rdf:resource="#incrementFileStart"/>
            </owl:Restriction>
      </rdfs:subClassOf>
```

```
</owl:Class>

<owl:Class rdf:ID="StaticParameter">
      <rdfs:comment>The static parameter is a parameter that is not
updated during the model, i.e. the temporal granularity of the
parameter is greater than the temporal extent of the model
</rdfs:comment>
      <rdfs:subClassOf rdf:resource="Parameter"/>
      <rdfs:subClassOf>
            <owl:Restriction owl:cardinality="1">
                  <owl:onProperty rdf:resource="#inputFileDirectory"/>
            </owl:Restriction>
      </rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:ID="Process">
      <rdfs:label>Process</rdfs:label>
      <rdfs:comment></rdfs:comment>
      <rdfs:subClassOf>
            <owl:Restriction owl:cardinality="1">
                  <owl:onProperty rdf:resource="&rdfs;label"/>
            </owl:Restriction>
      </rdfs:subClassOf>
      <rdfs:subClassOf>
            <owl:Restriction owl:cardinality="1">
                  <owl:onProperty rdf:resource="#color"/>
            </owl:Restriction>
      </rdfs:subClassOf>
      <rdfs:subClassOf>
            <owl:Restriction owl:cardinality="1">
                  <owl:onProperty rdf:resource="#spatialExtent"/>
            </owl:Restriction>
      </rdfs:subClassOf>
      <rdfs:subClassOf>
            <owl:Restriction owl:cardinality="1">
                  <owl:onProperty rdf:resource="#changeProcessRule"/>
            </owl:Restriction>
      </rdfs:subClassOf>
      <rdfs:subClassOf>
            <owl:Restriction owl:cardinality="1">
                  <owl:onProperty rdf:resource="#destroyProcessRule"/>
            </owl:Restriction>
      </rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:ID="ProcessModel">
      <rdfs:label>ProcessModel</rdfs:label>
      <rdfs:comment>This defines the properties, processes, and
parameters of the model.  A ProcessModel must have at least one
Process, one Parameter, and one ModelControl.  A process must also have
a modelName.</rdfs:comment>
      <rdfs:subClassOf>
            <owl:Restriction owl:minCardinality="1">
                  <owl:onProperty rdf:resource="#hasProcess"/>
            </owl:Restriction>
      </rdfs:subClassOf>
      <rdfs:subClassOf>
            <owl:Restriction owl:minCardinality="1">
                  <owl:onProperty rdf:resource="#hasParameter"/>
            </owl:Restriction>
      </rdfs:subClassOf>
      <rdfs:subClassOf>
            <owl:Restriction owl:cardinality="1">
                  <owl:onProperty rdf:resource="#hasModelControl"/>
            </owl:Restriction>
      </rdfs:subClassOf>
      <rdfs:subClassOf>
            <owl:Restriction owl:cardinality="1">
                  <owl:onProperty rdf:resource="#modelName"/>
            </owl:Restriction>
      </rdfs:subClassOf>
      <rdfs:subClassOf>
            <owl:Restriction owl:cardinality="1">
                  <owl:onProperty rdf:resource="#hasExtent"/>
            </owl:Restriction>
      </rdfs:subClassOf>
      <rdfs:subClassOf>
            <owl:Restriction owl:cardinality="1">
                  <owl:onProperty rdf:resource="#modelTemporalExtent"/>
            </owl:Restriction>
      </rdfs:subClassOf>
</owl:Class>

<!-- RULES -->


<swrl:Imp rdf:ID="CreateProcessRule">
      <rdfs:label>CreateProcessRule</rdfs:label>
      <rdfs:subClassOf>
            <owl:Restriction owl:cardinality="1">
                  <owl:onProperty rdf:resource="&rdfs;label"/>
            </owl:Restriction>
      </rdfs:subClassOf>
</swrl:Imp>

<swrl:Imp rdf:ID="ChangeProcessRule">
      <rdfs:label>ChangeProcessRule</rdfs:label>
      <rdfs:subClassOf>
            <owl:Restriction owl:cardinality="1">
                  <owl:onProperty rdf:resource="&rdfs;label"/>
            </owl:Restriction>
      </rdfs:subClassOf>
</swrl:Imp>

<swrl:Imp rdf:ID="DestroyProcessRule">
      <rdfs:label>DestroyProcessRule</rdfs:label>
      <rdfs:subClassOf>
            <owl:Restriction owl:cardinality="1">
                  <owl:onProperty rdf:resource="&rdfs;label"/>
            </owl:Restriction>
      </rdfs:subClassOf>
</swrl:Imp>
```

```
<!-- METHODS & THEIR PROPERTIES -->

<method:MethodObjectProperty rdf:ID="newProcess">
      <rdfs:label>newProcess</rdfs:label>
      <rdfs:domain rdf:resource="#ProcessModel"/>
      <rdfs:range rdf:resource="#Process"/>
</method:MethodObjectProperty>

<method:MethodObjectProperty rdf:ID="changeProcess">
      <rdfs:label>changeProcess</rdfs:label>
      <rdfs:domain rdf:resource="#ProcessModel"/>
      <rdfs:range rdf:resource="#Process"/>
</method:MethodObjectProperty>

</rdf:RDF>
```

APPENDIX E

Runoff Implementation Ontology

```
<!DOCTYPE rdf:RDF [
   <!ENTITY xsd   "http://www.w3.org/2000/10/XMLSchema#">
   <!ENTITY owl   "http://www.w3.org/2002/07/owl#">
   <!ENTITY flux
"file:///C:/workspace/eclipseWorkspace/flux/input/ontologies/flux#">
   <!ENTITY swrlb "http://www.w3.org/2003/11/swrlb#">
   <!ENTITY rdf   "http://www.w3.org/1999/02/22-rdf-syntax-ns#">
   <!ENTITY method
"file:///C:/workspace/eclipseWorkspace/flux/input/ontologies/method#" >
   <!ENTITY swrl  "http://www.w3.org/2003/11/swrl#">
]>


<rdf:RDF
    xmlns
="file:///C:/workspace/eclipseWorkspace/flux/input/ontologies/runoff#"
    xmlns:runoff
="file:///C:/workspace/eclipseWorkspace/flux/input/ontologies/runoff#"
    xml:base
="file:///C:/workspace/eclipseWorkspace/flux/input/ontologies/runoff#"
    xmlns:flux
="file:///C:/workspace/eclipseWorkspace/flux/input/ontologies/flux#"
    xmlns:owl          ="http://www.w3.org/2002/07/owl#"
    xmlns:rdf          ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs         ="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:xsd          ="http://www.w3.org/2000/10/XMLSchema#"
    xmlns:time         ="http://www.isi.edu/~pan/damltime/time.owl"
    xmlns:swrl         ="http://www.w3.org/2003/11/swrl#"
    xmlns:ruleml  ="http://www.w3.org/2003/11/swrl#"
    xmlns:swrlb        ="http://www.w3.org/2003/11/swrlb#"
    xmlns:method
="file:///C:/workspace/eclipseWorkspace/flux/input/ontologies/method#"
>

<owl:Ontology
rdf:about="http://www.glue.umd.edu/~femke/ProcessModel/runoff.owl#">
  <rdfs:comment>An example process modeling ontology</rdfs:comment>
  <rdfs:label>Simple Sample Runoff Ontology</rdfs:label>
</owl:Ontology>


<!-- CLASSES -->

<owl:Class rdf:ID = "Precipitation">
      <rdfs:label>Precipitation</rdfs:label>
      <rdfs:subClassOf rdf:resource="&flux;IncrementParameter"/>
      <flux:temporalGrain>1</flux:temporalGrain>
      <flux:temporalExtent>3</flux:temporalExtent>
      <flux:inputFileDirectory>C:/workspace/eclipseWorkspace/flux/input
/data/test/</flux:inputFileDirectory>
      <flux:incrementFileStart>test</flux:incrementFileStart>
</owl:Class>

<owl:Class rdf:ID ="Runoff">
      <rdfs:label>Runoff</rdfs:label>
      <rdfs:subClassOf rdf:resource="&flux;Process"/>
      <flux:temporalExtent>20</flux:temporalExtent>
      <flux:spatialExtent rdf:resource="#Elevation"/>
      <flux:spatialGrain>1</flux:spatialGrain>
      <flux:energy>4</flux:energy>
      <flux:color>blue</flux:color>
</owl:Class>

<owl:Class rdf:ID = "Elevation">
      <rdfs:label>Elevation</rdfs:label>
      <rdfs:subClassOf rdf:resource="&flux;StaticParameter"/>
      <flux:temporalGrain>20</flux:temporalGrain>  <!-- i.e. temporal
grain = temporalExtent -->
      <flux:temporalExtent>20</flux:temporalExtent>
      <flux:inputFilePath>C:/workspace/eclipseWorkspace/flux/input/data
/dem30m_subset4</flux:inputFilePath>
</owl:Class>


<!-- INDIVIDUALS -->

<flux:ProcessModel rdf:ID = "RunoffModel">
      <rdfs:label>RunoffModel</rdfs:label>
      <rdfs:subClassOf rdf:resource="&flux;ProcessModel"/>
      <flux:hasProcess rdf:resource="#Runoff"/>
      <flux:hasParameter rdf:resource="#Elevation"/>
      <flux:hasParameter rdf:resource="#Precipitation"/>
      <flux:modelName>RunoffModel</flux:modelName>
      <flux:hasExtent rdf:resource="#Elevation"/>
      <flux:modelTemporalExtent>30</flux:modelTemporalExtent>
      <flux:displayParameter rdf:resource="#Elevation"/>
</flux:ProcessModel>

<!-- PROPERTIES -->

<method:MethodObjectProperty rdf:ID="newRunoffProcess">
      <rdfs:label>newRunoffProcess</rdfs:label>
      <rdfs:subPropertyOf rdf:resource="&flux;newProcess"/>
      <rdfs:domain rdf:resource="#RunoffModel"/>
      <rdfs:range rdf:resource="#Runoff"/>
</method:MethodObjectProperty>

<method:MethodObjectProperty rdf:ID="changeRunoffProcess">
      <rdfs:label>changeRunoffProcess</rdfs:label>
      <rdfs:subPropertyOf rdf:resource="&flux;changeProcess"/>
      <rdfs:domain rdf:resource="#RunoffModel"/>
      <rdfs:range rdf:resource="#Runoff"/>
</method:MethodObjectProperty>
```

```
<!-- RULES - THRESHOLDS -->


<swrl:Variable rdf:ID="a"/>
<swrl:Variable rdf:ID="x"/>
<swrl:Variable rdf:ID="y"/>
<swrl:Variable rdf:ID="p"/>
<swrl:Imp rdf:ID ="createRunoffRule">
      <rdfs:label>createRunoffRule</rdfs:label>
      <rdf:type rdf:resource="&flux;CreateProcessRule"/>
      <swrl:body rdf:parseType="Collection">
            <method:MethodAtom>
                  <swrl:propertyPredicate
                  rdf:resource="&flux;parameterXYValue"/>
                  <swrl:argument1 rdf:resource="#Precipitation"/>
                  <swrl:argument2 rdf:resource="#a"/>
            </method:MethodAtom>
            <swrl:BuiltinAtom>
                  <swrl:propertyPredicate
                  rdf:resource="&swrlb;greaterThan"/>
                  <swrl:argument1 rdf:resource="#a"/>
                  <swrl:argument2>1.65</swrl:argument2>
            </swrl:BuiltinAtom>
      </swrl:body>
      <swrl:head rdf:parseType="Collection">
            <swrl:DatavaluedPropertyAtom>
                  <swrl:propertyPredicate
                  rdf:resource="&flux;parameterX"/>
                  <swrl:argument1 rdf:resource="#Precipitation"/>
                  <swrl:argument2 rdf:resource="#x"/>
            </swrl:DatavaluedPropertyAtom>
            <swrl:DatavaluedPropertyAtom>
                  <swrl:propertyPredicate
                  rdf:resource="&flux;parameterY"/>
                  <swrl:argument1 rdf:resource="#Precipitation"/>
                  <swrl:argument2 rdf:resource="#y"/>
            </swrl:DatavaluedPropertyAtom>
            <method:MethodAtom>
                  <swrl:propertyPredicate
                  rdf:resource="#newRunoffProcess"/>
                  <swrl:argument1 rdf:resource="#RunoffModel"/>
                  <swrl:argument2 rdf:resource="#p"/>
            </method:MethodAtom>
      </swrl:head>
</swrl:Imp>

<!-- change runoff process rule -->
<swrl:Variable rdf:ID="b"/>
<swrl:Variable rdf:ID="c"/>
<swrl:Variable rdf:ID="d"/>
<swrl:Variable rdf:ID="e"/>
<swrl:Variable rdf:ID="minNeighbor"/>

<swrl:Imp rdf:ID = "changeRunoffRule">
      <rdf:type rdf:resource="&flux;ChangeProcessRule"/>
      <rdfs:label>changeRunoffRule</rdfs:label>
      <swrl:body rdf:parseType="Collection">
            <swrl:ClassAtom>
                  <swrl:classPredicate rdf:resource="#Runoff"/>

                  <swrl:argument1 rdf:resource="#e"/>
            </swrl:ClassAtom>
            <swrl:ClassAtom>
                  <swrl:classPredicate rdf:resource="#Elevation"/>

                  <swrl:argument1 rdf:resource="#minNeighbor"/>
            </swrl:ClassAtom>
            <method:MethodAtom>
                  <swrl:propertyPredicate
                  rdf:resource="&flux;hasMinMooreNeighbor"/>
                  <swrl:argument1 rdf:resource="#e"/>
                  <swrl:argument2 rdf:resource="#minNeighbor"/>
            </method:MethodAtom>
            <method:MethodAtom>
                  <swrl:propertyPredicate
                  rdf:resource="&flux;minMooreNeighborXYValue"/>

                  <swrl:argument1 rdf:resource="#minNeighbor"/>
                  <swrl:argument2 rdf:resource="#b"/>
            </method:MethodAtom>
            <method:MethodAtom>
                  <swrl:propertyPredicate
                  rdf:resource="&flux;processX2Y2Value"/>

                  <swrl:argument1 rdf:resource="#e"/>
                  <swrl:argument2 rdf:resource="#c"/>
            </method:MethodAtom>
            <swrl:BuiltinAtom>
                  <swrl:propertyPredicate
                  rdf:resource="&swrlb;lessThan"/>
                  <swrl:argument1 rdf:resource="#b"/>
                  <swrl:argument2 rdf:resource="#c"/>
            </swrl:BuiltinAtom>
      </swrl:body>
      <swrl:head rdf:parseType="Collection">
            <method:MethodAtom>
                  <swrl:propertyPredicate
                  rdf:resource="&flux;move2DProcessTo"/>
                  <swrl:argument1 rdf:resource="#e"/>
                  <swrl:argument2 rdf:resource="#minNeighbor"/>
            </method:MethodAtom>

      </swrl:head>
</swrl:Imp>


</rdf:RDF>
```

APPENDIX F

Summary of RCEW Data

Data Report	Parameter Measured	Max No. Stations	1996 No. Stations	Years of Record[a]	1996 Sampling Interval[b]
Precipitation (Hanson, 2001)	shielded precipitation, unshielded precipitation, calculated precipitation	53	17	1962-1996	breakpoint[c], 15 minute
Snow (Marks *et al.*, 2001)	Snow course SWE	8	8	1961-1996	biweekly
	Snow pillow SWE	1	1		15 minute
Daily climate (Hanson *et al.*, 2001)	$T_{\max}$ and $T_{\min}$	3	3	1964-1996	daily
	pan evaporation			1974-1996	
Continuous climate (Hanson *et al.*, 2001)	air temp, humidity, solar radiation, wind speed and direction, barometric pressure	3	3	1981-1996	15 minute
Soil Lysimeter (Seyfried *et al.*, 2001c)	lysimeter water content	4	0	1976-1991	hourly
Neutron probe (Seyfried *et al.*, 2001d)	soil water content (various depths)	18	14	1970-1996	biweekly
Soil temperature (Seyfried *et al.*, 2001b)	soil temperature (various depths)	5	5	1981-1996	15 minute
Discharge and sediment (Pierson *et al.*, 2001)	stream discharge	13	8	1963-1996	breakpoint[d], 15 minute
	Suspended sediment	3	3	1965-1996	event based

Source: adapted from (Slaughter *et al.*, 2001)

[a] The period of record indicates the initial and final year of data considering all sites. Some sites may have started later or ended earlier, and gaps in the record may occur.

[b] The 1996 sampling interval may not be the same as the data recording interval in the database.

[c] Nominal value is 0.25 mm of precipitation or 15 min sample.

[d] Nominal value is 0.5 mm of stage in 5 min or 15 min stage sample for small weirs and fixed 15 min stage sample for large weirs.

REFERENCES

Abbott, M. B., J. C. Bathurst, J. A. Cunge, P. E. O'Connell, and J. Rasmussen, 1986a. An Introduction to the European Hydrological System - Systeme Hydrologique Europeen, "SHE", 1: history and philosophy of a physically-based, distributed modelling system. *Journal of Hydrology*, 87**:** 45-59.

Abbott, M. B., J. C. Bathurst, J. A. Cunge, P. E. O'Connell, and J. Rasmussen, 1986b. An Introduction to the European Hydrological System - Systeme Hydrologique Europeen, "SHE", 2: structure of a physically-based, distributed modelling system. *Journal of Hydrology*, 87**:** 61-77.

Abler, R., J. S. Adams, and P. Gould, 1971. *Spatial Organization: The Geographer's View of the World.* Prentice-Hall, New Jersey.

Abraham, T. and J. F. Roddick, 1999. Survey of Spatio-Temporal Databases. *Geoinformatica*, 3**:** 61-99.

Ahl, V. and T. F. H. Allen, 1996. *Hierarchy Theory: a vision, vocabulary, and epistemology*. Columbia University Press, New York.

Ahrens, C. D., 1991. *Meteorology Today: an introduction to weather, climate, and the environment.* 4th ed. West Publishing Company, New York.

Albrecht, J., 1999. Towards Interoperable Geo-information Standards: a comparison of reference models for geo-spatial information. *The Annals of Regional Science*, 33**:** 151-169.

Albrecht, J. and A. Car, 1999. GIS analysis for scale-sensitive environmental modelling based on hierarchy theory. *GIS for Earth Surface Systems*, R. Dikau and H. Saurer, Eds., Gebruder Borntraeger, 1-23.

Allen, T. F. H. and T. B. Starr, 1982. *Hierarchy Theory: perspectives for ecological complexity*. The University of Chicago Press, Chicago.

Audi, R., Ed., 1999: *The Cambridge Dictionary of Philosophy*. Second Edition ed. Cambridge University Press.

Baird, A., 2004. Soil and Hillslope Hydrology. *Environmental Modelling: Finding Simplicity in Complexity*, J. Wainright and M. Mulligan, Eds., John Wiley & Sons Ltd.

Balashov, Y., 2000. Persistence and Space-time: Philosophical Lessons of the Pole and Barn. *Monist*, 83**:** 321-341.

Bauer, B. O., T. T. Veblen, and J. A. Winkler, 1999. Old Methodological Sneakers: Fashion and Function in a Cross-Training Era. *Annal of the Association of American Geographers*, 89**:** 679-687.

Bernard, L. and T. Kruger, 2000. Integration of GIS and Spatio-temporal Simulation Models: Interoperable Components for Different Simulation Strategies. *Transactions in GIS*, 4**:** 197-215.

Berners-Lee, T., J. Hendler, and O. Lassila, 2001. The Semantic Web: a new form of web content that is meaningful to computers will unleash a revolution of new possibilities. *The Scientific American*, 284**:** 34-43.

Bevan, K. J., 2000. Uniqueness of Place and Process Representations in Hydrological Modelling. *Hydrology and Earth System Sciences*, 4**:** 203-213.

Beven, K. J., 2001. *Rainfall-runoff Modelling: The Primer*. Wiley, Chichester.

Beven, K. J., 2002. Towards an Alternative Blueprint for a Physically Based Digitally Simulated Hydrologic Response Modelling System. *Hydrological Processes*, 16**:** 189-206.

Bian, L., 2000. Object-oriented Representation for Modelling Objects in an Aquatic Environment. *International Journal of Geographic Information Science*, 14**:** 603-623.

Bittner, T. and G. Edwards, 2001. Towards an Ontology for Geomatics. *Geomatica*, 55**:** 475-490.

Bittner, T. and B. Smith, 2001: A Taxonomy of Granular Partitions. *COSIT 2001*, Morro Bay, Springer, 28-43.

Bittner, T. and B. Smith, 2003. Formal Ontologies for Space and Time.

Blaut, J. M., 1961. Space and Process. *The Professional Geographer*, 13**:** 1-7.

Bohm, D., 1980. *Wholeness and the Implicate Order*. Routeledge & Kegan Paul, London.

Borah, D. K. and M. Bera, 2003. Watershed-Scale Hydrologica and Nonpoint-Source Pollution Models: Review of Mathematical Bases. *Transactions of the American Society of Agricultural Engineers*, 46**:** 1553-1566.

Borah, D. K. and M. Bera, 2004. Watershed-Scale Hydrologica and Nonpoint-Source Pollution Models: Review of Applications. *Transactions of the American Society of Agricultural Engineers*, 47**:** 789-803.

Borges, K. A. V., C. A. Davis, and A. H. F. Laender, 2001. OMT-G: An Object-Oriented Data Model for Geographic Applications. *Geoinformatica*, 5**:** 221-260.

Box, P., 2002. Spatial Units as Agents: Making the Landscape an Equal Player in Agent-Based Simulations. *Integrating Geographic Information Systems and Agent-based Modeling Techniques for Simulating Social and Ecological Processes*, H. R. Gimblett, Ed., Oxford University Press, 59-82.

Brodeur, J., Y. Bedard, G. Edwards, and B. Moulin, 2003. Revisiting the Concept of Geospatial Data Interoperability within the Scope of Human Communication Processes. *Transactions in GIS*, 7**:** 243-265.

Brogaard, B., 2000. Presentist Four-Dimensionalism. *Monist*, 83**:** 341-357.

Brown, D. G., S. E. Page, R. L. Riolo, and W. Rand, forthcoming. Agent-based and Analytical Modeling to Evaluate the Effectiveness of Greenbelts. *Environmental Modelling & Software*.

Casati, R., B. Smith, and A. C. Varzi, 1998. Ontological Tools for Geographic Representation. *Formal Ontology in Information Systems*, N. Gaurino, Ed., IOS Press, 77-85.

Chapman, G. P., 1977. *Human and Environmental Systems: A Geographer's Appraisal*. Academic Press, London.

Chen, J. and J. Jiang, 2000. An Event-Based Approach to Spatio-Temporal Data Modeling in Land Subdivision Systems. *Geoinformatica*, 4**:** 387-402.

Christopherson, R. W., 2001. *Elemental Geosystems*. Prentice Hall, Upper Sadle River.

Claramunt, C. and M. Thériault, 1996: Toward Semantics for Modelling Spatio-Temporal Processes within GIS. *Proceedings of the International Symposioum on Spatial Data Handling (SDH'96), Netherlands*, 2.27-2.43.

Claramunt, C. and C. Parent, 2003. Modelling Concepts for the Representation of Evolution Constraints. *Computers, Environment and Urban Systems*, 27**:** 225-241.

Claramunt, C., C. Parent, and M. Thériault, 1997. Design Patterns for Spatio-temporal Processes. *Searching for Semantics: Data Mining, Reverse Engineering*, S. Spaccapieta and F. Marganski, Eds., Chapman & Hall, 415-428.

Clarke, K. C., B. O. Parks, and M. P. Crane, 2001. *Geographic Information Systems and Environmental Modeling*. Prentice Hall Series in Geographic Information Science, Prentice-Hall, Upper Saddle River.

Coffey, W. J., 1981. *Geography: Towards a General Spatial Systems Approach.* Methuen, London.

Cohn, A. G. and S. M. Hazarika, 2001. Qualitative Spatial Representation and Reasoning: An Overview. *Fundamenta Informaticae*, 43**:** 2-32.

Couclelis, H., 1992. Location, Place, Region, and Space. *Geography's Inner Worlds*, R. F. Abler, M. G. Marcus, and J. M. Olson, Eds., Rutgers University Press, 215-233.

Couclelis, H., 1992b. People Manipulate Objects (But Cultivate Fields): Beyond the Raster-Vector Debate in GIS. *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*, A. Frank, I. Campari, and U. Formentini, Eds., Springer.

Couclelis, H. and X. Liu, 2000: The Geography of Time and Ignorance: Dynamics and Uncertainty in Integrated Urban-Environmental Process Models. *4th International Conference on Integrating GIS and Environmental Modeling (GIS/EM4)*, Banff, Alberta, Canada.

Coulthard, T. J., M. J. Kirkby, and M. G. Mackline, 2000. Modelling Geomorphic Response to Environmental Change in an Upland Catchment. *Hydrological Processes*, 14**:** 2031-2045.

Cova, T. J. and M. F. Goodchild, 2002. Extending Geographical Representation to Include Fields of Spatial Objects. *International Journal of Geographic Information Science*, 16**:** 509-532.

De Roo, A. P. J., C. G. Wesseling, and W. P. A. van Deursen, 2000. Physically-based River Basin Modelling Within a GIS: The LISFLOOD Model. *Hydrological Processes*, 14**:** 1981-1992.

De Vasconcelos, M. J. P., A. Gonçalves, F. X. Catry, J. U. Paúl, and F. Barros, 2002. A Working Prototype of a Dynamic Geographical Information System. *International Journal of Geographic Information Science*, 16**:** 69-91.

Duckham, M., M. F. Goodchild, and M. F. Worboys, Eds., 2003: *Foundations of Geographic Information Science*. Taylor & Francis.

Eberhart, A., 2002: Automatic Generation of Java/SQL Based Inference Engines from RDF Schema and RuleML. *International Semantic Web Converence (ISWC) 2002, Lecture Notes in Computer Science, Springer-Verlag, Heidelberg, Volume 2342, pp. 102-116.*

Egenhofer, M. and J. Herring, 1990: A Mathematical Framework for the Definition of Topological Relationships. *Fourth International Symposium on Spatial Data Handling*, K. Brassel and H. Kishimoto (eds) Zurich, Switzerland, pp. 803-813.

Egenhofer, M., J. Glasgow, O. Günther, J. Herring, and D. J. Peuquet, 1999. Progress in Computational Methods for Representing Geographical Concepts. *International Journal of Geographic Information Science*, 13**:** 775-796.

Endreny, T. A. and E. F. Wood, 2003. Maximizing Spatial Congruence of Observed and DEM-delineated Overland Flow Networks. *International Journal of Geographic Information Science*, 17**:** 699-713.

Epstein, J. M. and R. Axtell, 1996. *Growing Artificial Societies: Social Science from the Bottom Up*. The MIT Press, Cambridge.

Erwig, M. and M. Schneider, 1999: Developments in Spatio-Temporal Query Languages. *IEEE Dexa 1999, Proceedings of the 10th International Conference, Florence.*

Erwig, M., R. H. Güting, M. Schneider, and M. Vazirgiannis, 1999. Spatio-temporal Data Types: An Approach to Modeling and Querying Moving Objects in Databases. *Geoinformatica*, 3**:** 269-296.

Farquhar, A., R. Fikes, and J. Rice: The Ontolingua Server: a Tool for Collaborative Ontology Construction. [Available online from www.ksl.stanford.edu/software/ontolingua/project-papers.html.]

Favis-Mortlock, D. T., J. Boardman, A. J. Parsons, and B. Lascelles, 2000. Emergence and Erosion: a model for rill initiation and development. *Hydrological Processes*, 14**:** 2173-2205.

Feng, C. and A. Sorokine, 2001: Incorporating Hydrologic Semantic Information for Interoperable GIS with Hydrologic Model. *Proceedings of the 9th ACM International Symposium on Advances in Geographic Information Systems*, Atlanta, Georgia, ACM Press, New York, p59-63.

Fonseca, F. and M. Egenhofer, 1999: Ontology-Driven Geographic Information Systems. *7th ACM Symposium on Advances in Geographic Information Systems*, Kansas City, MO.

Fonseca, F., M. Egenhofer, P. Agouris, and G. Câmara, 2002. Using Ontologies for Integrated Geographic Information Systems. *Transactions in GIS*, 6**:** 231-257.

Forbus, K. D., 1984. Qualitative Process Theory. *Artificial Intelligence*, 24**:** 85-168.

Fotheringham, A. S., C. Brunsdon, and M. Charlton, 2000. *Quantitative Geography: perspectives on spatial data analysis*. SAGE Publications, London.

Frank, A., forthcoming. Ontology for Spatio-temporal databases. *Spatio-temporal Databases: The Chorochronos Approach*, Springer-Verlag.

Frank, A., J. Raper, and J. Cheylan, Eds., 2001: *Life and Motion of Socio-economic Units*. Vol. 8. *GISDATA*, Taylor & Francis.

Frank, A. U., 2001. Socio-Economic Units: their life and motion. *Life and Motion of Socio-Economic Units*, 8 ed. A. U. Frank, J. Raper, and J. Cheylan, Eds., Taylor & Francis.

Freeze, R. A. and R. L. Harlan, 1969. Blueprint for a Physically-based, Digitally-simulated Hydrologic Response Model. *Journal of Hydrology*, 9**:** 237-258.

Frihida, A., D. J. Marceau, and M. Thériault, 2002. Spatio-Temporal Object-Oriented Data Model for Disaggregate Travel Behavior. *Transactions in GIS*, 6**:** 277-294.

Galton, A., 2000. *Qualitative Spatial Change*. Spatial Information Systems, Oxford University Press, Oxford.

Galton, A., 2001. Space, Time, and the Representation of Geographical Reality. *Topoi*, 20**:** 173-187.

Gangemi, A., N. Guarino, C. Masolo, A. Oltramari, and L. Schneider, 2002: Sweetening Ontologies with DOLCE. *EKAW 2002*, Siguenza, Spain, Lecture Notes in Computer Science, Springer, 166-181.

Gao, X., S. Sorooshian, and D. C. Goodrich, 1993. Linkage of a GIS to a Distributed Rainfall-Runoff Model. *Environmental Modeling with GIS*, M. F. Goodchild, B. O. Parks, and L. T. Steyaert, Eds., Oxford University Press, 182-187.

Gatrell, A., 1983. *Distance and Space: A Geographical Perspective*. Clarendon Press, Oxford.

Getis, A. and B. Boots, 1978. *Models of spatial processes: an approach to the study of point, line and area patterns*. Vol. 8, Cambridge Geographical Studies, Cambridge University Press, Cambridge.

Gilbert, N. and P. Terna, 2000. How to Build and Use Agent-based Models in Social Science. *Mind & Society*, 1**:** 57-72.

Gimblett, H. R., Ed., 2002: *Integrating Geographic Information Systems and Agent-Based Modeling Techniques for Simulating Social and Ecological Processes*. *Sante Fe Institute Studies in the Sciences of Complexity*, Oxford University Press.

Goodchild, M. F., 2003. The Nature and Value of Geographic Information. *Foundations of Geographic Information Science*, M. Duckham, M. F. Goodchild, and M. F. Worboys, Eds., Taylor & Francis, 19-32.

Grayson, R. and G. Blöschl, 2000. Spatial Modeling of Catchment Dynamics. *Spatial Patterns in Catchment Hydrology*, R. Grayson and G. Blöschl, Eds., Cambridge University Press, 51-81.

Grenon, P. and B. Smith, 2003. SNAP and SPAN: Prolegomenon to Geodynamic Ontology.

Griffiths, T., A. A. A. Fernandes, N. D, and N. W. Paton, 2001: A Query Calculus for Spatio-Temporal Object Databases. *Proceedings of the 8th International Symposium on Temporal Representation and Reasoning (TIME-01), IEEE Computer Society Press, pp. 101-110, Cividale del Friuli, Italy. June 14-16, 2001.*

Griffiths, T., A. A. A. Fernandes, N. W. Paton, K. T. Mason, B. Huang, and M. F. Worboys, 2001: Tripod: A Comprehensive Model for Spatial and Aspatial Historical Objects. *Proceedings ER 2001 - 20th International Conference on Conceptual Modelling, Yokohama, Japan, November 27-30. Lecture Notes in Computer Science, Springer-Verlag, Heidelberg.*

Griffiths, T., A. Fernandes, N. W. Paton, K. T. Mason, B. Huang, M. F. Worboys, C. Johnson, and J. Stell, 2001: Tripod: A Comprehensive System for the Management of Spatial and Aspatial Historical Objects. *ACM-GIS*, 118-123.

Gruber, T. R., 1993. A Translation Approach to Portable Ontologies. *Knowledge Acquisition*, 5**:** 199-220.

Grüninger, M., 2003. Ontology of the Process Specification Language. *Handbook on Ontologies*, S. Staab and R. Studer, Eds., Springer, 599-618.

Grüninger, M., 2003: Applications of PSL to Semantic Web Services. *Proceedings of SWDB'03, the First International Workshop on Semantic Web and Databases, September 7-8, 2003, Berlin, Germany, pp. 217-230.*

Guarino, N., 1998: Formal Ontology and Information Systems. *Proceedings of FOIS' 98*, Trento, Italy, IOS Press, 3-15.

Gulyás, L., 2002. On the Transition to Agent-Based Modeling. *Social Science Computer Review*, 20**:** 389-399.

Gurtz, J., M. Zappa, J. Karsten, L. Herbert, M. Verbunt, A. Badoux, and T. Vitvar, 2003. A Comparative Study in Modelling Runoff and its Components in Two Mountainous Catchments. *Hydrological Processes*, 17**:** 297-311.

Haff, P. K., 2001. Waterbots. *Landscape Erosion and Evolution Modeling*, R. S. Harmon and W. W. Doe, Eds., Kluwer, 239-275.

Hägerstrand, T., 1967. *Innovation Diffusion as a Spatial Process*. The University of Chicago Press, Chicago.

Hamre, T., K. Azim Mughal, and A. Jacob, 1997. A 4D Marine Data Model: Design and Application in Ice Monitoring. *Marine Geodesy*, 20.

Hanson, C. L., 2001. Long-term Precipitation Database, Reynolds Creek Experimental Watershed, Idaho, United States. *Water Resources Research*, 37**:** 2831-2834.

Hanson, C. L. and J. R. Wight, 1995: Evapotranspiration Fluxes Within a Rangeland Watershed. *Fifth International Rangeland Congress*, Salt Lake City, Utah, 205-206.

Hanson, C. L., D. Marks, and S. S. Van Vactor, 2001. Long-term Climate Database, Reynolds Creek Experimental Watershed, Idaho, United States. *Water Resources Research*, 37**:** 2839-2841.

Hartshorne, C., 1998. The Development of Process Philosophy. *Philosophers of Process*, D. Browning and W. T. Myers, Eds., Fordham University Press, 391-407.

Hartshorne, R., 1996. The Character of Regional Geography. *Human Geography: an essential anthology*, J. Agnew, D. Livingstone, and A. Rogers, Eds., Blackwell Publishers.

Harvey, D., 1969. *Explanation in Geography*. Edward Arnold, London.

Henderson-Sellers, A. and K. McGuffie, 1987. *A Climate Modelling Primer*. John Wiley & Sons, New York.

Honderich, T., Ed., 1995: *The Oxford Companion to Philosophy*. Oxford University Press.

Hornsby, K. and M. Egenhofer, 2000. Identitiy-Based Change: A Foundation For Spatio-Temporal Knowledge Representation. *International Journal of Geographic Information Systems*, 14**:** 207-224.

Huang, B. and C. Claramunt, 2002: STOQL: An ODMG-Based Spatio-Temporal Object Model Query Language. *Advances in Spatial Data Handling, 10th International Symposium on Spatial Data Handling*, Ottawa, Canada, July 9-12, Springer-Verlag, Heidelberg, 225-238.

Johansson, I., 1989. *Ontological Investigations: an inquiry into the categories of nature, man and society*. Routledge, London.

Kalyanpur, A., D. Partor, S. Battle, and J. Padget, 2004: Automatic Mapping of OWL Ontologies into Java. *Proceedings of Sixteenth International Conference on Software Engineering and Knowledge Engineering (SEKE 2004), 20-24 June, Banff.*

Kavouras, M., 2001. Understanding and Modelling Spatial Change. *Life and Motion of Socio-economic Units*, A. Frank, J. Raper, and J. Cheylan, Eds., Taylor & Francis, 49-62.

Kavouras, M. and M. Kokla, 2002. A Method for the Formalization and Integration of Geographical Categorizations. *International Journal of Geographic Information Systems*, 16**:** 439-453.

Kemp, K., 1997. Fields as a framework for integrating GIS and environmental process models. Part 1: Representing spatial continuity. *Transactions in GIS*, 1**:** 219-234.

Kuhn, W., 2001. Ontologies in Support of Activities in Geographic Space. *International Journal of Geographic Information Science*, 15**:** 613-631.

Lam, N. S. and D. A. Quattrochi, 1992. On the Issue of Scale, Resolution, and Fractal Analysis in the Mapping Sciences. *Professional Geographer*, 44**:** 88-98.

Langran, 1993. *Time in Geographic Information Systems*. Taylor and Francis, London.

Langton, C., 1986. Studying Artificial Life with Cellular Automata. *Physica D*, 22.

Li, X. and M. Hodgson, 2004. Vector Field Data Model and Operations. *GIScience and Remote Sensing*, 41**:** 1-24.

Libourel, T., 2001. How Do Databases Perform Change? *Life and Motion of Socio-economic Units*, A. Frank, J. Raper, and J. Cheylan, Eds., Taylor & Francis, 155-166.

Lippincott, K., 1999. *The Story of Time*. Merrell Holberton, London.

Maidment, D. R., 1993. GIS and Hydrologic Modeling. *Environmental Modeling and GIS*, M. F. Goodchild, B. O. Parks, and L. T. Steyaert, Eds., Oxford University Press, 147-167.

Mark, D. M., M. Egenhofer, and K. Hornsby, 1996. Formal Models of Commonsense Geographic Worlds: report on the specialist meeting of research initiative 21NCGIA Technical Report 97-2, National Center for Geographic Information and Analysis, 1-39 pp.

Mark, D. M., M. Egenhofer, S. C. Hirtle, and B. Smith, 2002. Ontological Foundations for Geographic Information Science. *UCGIS Research Priorities: emerging research theme*.

Marks, D., 2001. Introduction to Special Section: Reynolds Creek Experimental Watershed. *Water Resources Research*, 37**:** 2817.

Marks, D., K. R. Cooley, D. C. Robertson, and A. Winstral, 2001. Long-term Snow Database, Reynolds Creek Experimental Watershed, Idaho, United States. *Water Resources Research*, 37**:** 2835-2838.

Mead, G. H., 1998. The Present as Locus of Reality. *Philosphers of Process*, D. Browning and W. T. Myers, Eds., Fordham University Press, 349-370.

Menzel, C. and M. Grüninger, 2001: A Formal Foundation for Process Modeling. *Formal Ontology in Information Systems FOIS'01, October 17-19 2001*, Ogunquit, Maine, USA.

Merz, R. and G. Blöschl, 2003: Regional Flook Risk - What are the Driving Processes? *Water Resources Systems - Hydrological Risk, Management and Development, Proceedings of Symposium HSO2b held during IUGG2003, Sapporo, July 2003*, 49-58.

Miller, H. J., 2003a. Activities in Space and Time. *Handbook in Transport 5: Transport Geography and Spatial Systems*, P. Stopher, K. Haynes, and D. Hensher, Eds., Pergamon/Elsevier Science: in press.

Miller, H. J., 2003b. What About People in Geographic Information Science. *Re-Presenting Geographic Information Systems*, D. Unwin, Ed., John Wiley: in press.

Miller, S. N., D. J. Semmens, R. C. Miller, M. Hernandez, D. C. Goodrich, W. P. Miller, W. G. Kepner, and D. Ebert, 2002: GIS-based Hydrological Modeling: the Automated Geospatial Watershed Assessment Tool. *Proceedings of the Second Interagency Hydrologic Modeling Conference, July 28-August 1 2002, Las Vegas, Nevada.*

Moravcsik, J. M., 1991. Appearance and Reality in Heraclitus' Philosophy. *Monist*, 74**:** 551-568.

Mota, L., J. Bento, and L. Botelho, 2002: Ontology Definition Languages for Multi-Agent Systems: the Geographical Information Ontology Case Study. *Proceedings of the Workshop on Ontologies in Agent Systems (OAS '02), 1st International Joint Conference on Autonomous Agents and Multi-agent Systems*, Bologna, Italy, CEUR Workshop Proceedings.

Mountrakis, G., P. Agouris, and A. Stefanidis, 2002: A Differential Spatio-temporal Model: Primitives and Operators. *Advances in Spatial Data Handling, 10th International Symposium on Spatial Data Handling*, Ottawa, Canada, July 9-12, Springer-Verlag, Heidelberg, 255-268.

Mulligan, M., 2004. Modelling Catchment Hydrology. *Environmental Modelling: Finding Simplicity in Complexity*, J. Wainright and M. Mulligan, Eds., John Wiley & Sons Ltd.

Munsat, S., 1969. What is a Process. *American Philosophical Quaterly*, 6**:** 79-83.

O'Sullivan, D., 2001. Exploring Spatial Process Dynamics Using Irregular Cellular Automaton Models. *Geographical Analysis*, 33**:** 1-18.

Olsson, G., 1975. *Birds in Egg*. Mighigan Geographical Publication No. 15, Department of Geography, University of Michigan, Ann Arbor, 523 pp.

Pan, F. and J. Hobbs, 2004: Time in OWL-S. *Proceedings of AAAI Spring Symposium on Semantic Web Services*, Stanford University, CA.

Pardo-Iguzquiza, E., 1998. Comparison of Geostatistical Methods for Estimating the Areal Average Climatological Rainfall Mean Using Data on Precipitation and Topography. *International Journal of Climatology*, 18**:** 1031-1047.

Parker, D. C. and V. Meretsky, 2004. Measuring Pattern Outcomes in an Agent-Based Model of Edge-Effect Externalities using Spatial Metrics. *Agriculture, Ecosystems, and Environment*, 101**:** 233-250.

Parsons, J., 2000. Must a Four-Dimensionalist Believe in Temporal Parts. *Monist*, 83**:** 399-419.

Pebesma, E. and C. G. Wesseling, 1998. Gstat: A Program for Geostatistical Modelling, Prediction and Simulation. *Computers & Geosciences*, 24**:** 17-31.

Pereira, G. M., 2002. A Typology of Spatial and Temoral Scale Relations. *Geographical Analysis*, 34**:** 21-33.

Peuquet, D. J., 1994. It's About Time: A Conceptual Framework for the Representation of Temporal Dynamics in Geographic Information Systems. *Annals of the Association of American Geographers*, 84**:** 441-461.

Peuquet, D. J., 2001. Making Space for Time: Issues in Space-Time Data Representation. *Geoinformatica*, 5**:** 11-32.

Peuquet, D. J., 2002. *Representations of Space and Time*. The Guilford Press, New York.

Pierson, F. B., C. W. Slaughter, and Z. K. Cram, 2001. Long-term Stream Discharge and Suspended-sediment Database, Reynolds Creek Experimental Watershed, Idaho, United States. *Water Resources Research*, 37**:** 2857-2861.

Price, R., K. Ramamohanarao, and B. Srinivasan, 1999: Spatiotemporal Extensions to the Unified Modeling Language. *IEEE Dexa 1999, Proceedings of the 10th International Conference, Florence.*, 460-461.

Pullar, D., 2003. Simulation Modelling Applied to Runoff Modelling Using MapScript. *Transactions in GIS*, 7**:** 267-283.

Quinn, P. F., K. J. Bevan, and R. Lamb, 1995. The ln(a/tanB) Index: How to Calculate it And How to Use it Within the TOPMODEL Framework. *Hydrological Processes*, 9**:** 161-182.

Rand, W., D. G. Brown, S. E. Page, R. Riolo, L. E. Fernandex, and M. Zellner, 2003. Statistical Validation of Spatial Patterns in Agent-Based Models. *Submitted to Agent-Based Simulation 4*.

Randell, D., Z. Cui, and A. G. Cohn, 1992: A Spatial Logic Based on Region and Connection. *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning*, pp. 165-176.

Raper, J., 2000. *Multidimensional Geographic Information Science*. Taylor & Francis, New York.

Raper, J. and D. Livingstone, 1995. Development of a Geomorphological Spatial Model Using Object-oriented Design. *International Journal of Geographic Information Science*, 9**:** 359-383.

Raper, J. and D. Livingstone, 1996: Spatio-Temporal Interpolation in Four Dimensional Coastal Process Models. *In Proceedings of the 3rd International Conference on Integrating GIS and Environmental Modelling, Sante Fe, NM, USA*.

Rawls, W. J., D. L. Brakensiek, and S. K. E, 1982. Estimation of Soil Water Properties. *Transactions of the American Society of Agricultural Engineers*, 25**:** 1316-1320.

Reitsma, F. and T. Bittner, 2003: Scale in Object and Process Ontologies. *Spatial Information Theory: Foundations of Geographic Information Science, COSIT'03*, Ittingen, Switzerland, Springer Lecture Notes in Computer Science, Berlin, 13-30.

Renolen, A., 2000. Modelling the Real World: Conceptual Modelling in Spatiotemporal Infomation System Design. *Transactions in GIS*, 4**:** 23-42.

Rescher, N., 1999. On Situating Process Philosophy. *Process Studies*, 28**:** 37-42.

Rescher, N., 2000. *Process Philosophy: a survey of basic issues*. University of Pittsburgh Press, Pittsburgh.

Romanowicz, R. and K. J. Beven, 1993. GIS and Distributed Hydrological Models. *Geographical Information Handling - Research and Applicaitons*, P. M. Mather, Ed., John Wiley & Sons, 197-205.

Rosenthal, S. B., 1999. Contemporary Metaphysics and the Issue of Time: Re-thinking the "Great Divide". *International Philosophical Quarterly*, 39**:** 157-171.

Ruelle, D., 2000. Conversations on Mathematics With a Visitor From Outer Space. *Mathematics: Frontiers and Perspectives*, V. Arnold, M. Atiyah, P. Lax, and B. Mazur, Eds., American Mathematical Society, 251-259.

Rumbaugh, J., I. Jacobson, and G. Booch, 1999. *The Unified Modeling Language Reference Manual.* Addison Wesley Longman Inc., Massachusetts.

Rumpe, B., 2002. Executable Modeling with UML. *Issues and Trends of Information Technology Management in Contemporary Associations.*, Idea Group Publishing.

Sauer, C. O., 1925. The Morphology of Landscape. *University of California Publications in Geography*, 2**:** 19-54.

Seibert, J. and J. McDonnell, 2000: Towards a Better Process Representation of Catchment Hydrology in Conceptual Runoff Modelling. *IAHS Conference on Meso-scale Modelling. Proceedings of the International Workshop on Runoff Generation and Implications for River Basin Modeling. Freiburg, 9-12 October 2000.*

Semboloni, F., 2000. The Growth of an Urban Cluster into a Dynamic Self-Modifying Spatial Pattern. *Environment and Planning B*, 27**:** 549-564.

Seyfried, M., R. Harris, D. Marks, and B. Jacob, 2001a. Geographic Database, Reynolds Creek Experimental Watershed, Idaho, United States. *Water Resources Research*, 37**:** 2825-2829.

Seyfried, M. S., C. L. Hanson, M. D. Murdock, and S. S. Van Vactor, 2001c. Long-term Lysimeter Database, Reynolds Creek Experimental Watershed, Idaho, United States. *Water Resources Research*, 37**:** 2853-2856.

Seyfried, M. S., G. N. Flerchinger, M. D. Murdock, C. L. Hanson, and S. S. Van Vactor, 2001b. Long-term Soil Temperature Database, Reynolds Creek Experimental Watershed, Idaho, United States. *Water Resources Research*, 37**:** 2843-2846.

Seyfried, M. S., M. D. Murdock, C. L. Hanson, G. N. Flerchinger, and S. S. Van Vactor, 2001d. Long-term Soil Water Content Database, Reynolds Creek Experimental Watershed, Idaho, United States. *Water Resources Research*, 37**:** 2847-2851.

Shaw, S. and D. Wang, 2000. Handling Disaggregate Spatiotemporal Travel Data in GIS. *Geoinformatica*, 4**:** 161-178.

Sider, T., 2001. *Four-Dimensionalism*. Clarendon Press, Oxford.

Silva, E. A. and K. C. Clarke, 2002. Calibration of the SLEUTH Urban Growth Model for Lisbon and Porto, Portugal. *Computers, Environment and Urban Systems*, 26**:** 525-552.

Simmons, R. G., 1983. Representing and Reasoning About Change in Geologic Interpretation, MIT Artificial Intelligence Laboratory, Technical Report 749.

Simons, P., 1987. *Parts: a study in ontology*. Clarendon Press, Oxford.

Simons, P., 2000. How to Exist at a Time When you Have no Temporal Parts. *Monist*, 83**:** 419-437.

Singh, V. P., Ed., 1995: *Computer Models of Watershed Hydrology*. Water Resources Publications.

Slaughter, C. W., D. Marks, G. N. Flerchinger, S. S. Van Vactor, and M. Burgess, 2001. Thirty-five Years of Research Data Collection at the Reynolds Creek Experimental Watershed, Idaho, United States. *Water Resources Research*, 37**:** 2819-2823.

Smith, B., 1995: On Drawing Lines on a Map. *Spatial Information Theory, Proceedings of COSIT '95*, Berlin, Springer Verlag, 475-484.

Smith, B. and D. M. Mark, 1998: Ontology and Geographic Kinds. *8th International Symposium on Spatial Data Handling (SDH'98)*, Vancouver, Canada, International Geographical Union, 308-320.

Smith, B. and C. Welty, 2001: Ontology: Towards a New Synthesis. *Formal Ontology in Information Systems*, Ongunquit, Maine, ACM Press, iii-x.

Smith, B. and B. Brogaard, to appear. Quantum Mereotopology. *Annals of Mathematics and Artificial Intelligence*.

Smyth, C. S., 1998. A Represenational Framework for Geographic Modeling. *Spatial and Temporal Reasoning in Geographic Information Systems*, M. Egenhofer and R. G. Golledge, Eds., Oxford University Press, 191-213.

Sowa, J. F., 2000. *Knowledge Representation: Logical, Philosphical, and Computational Foundations*. Brooks/Cole, Pacific Grove, California.

Stefanakis, E., 2003. Modelling the History of Semi-structured Geographical Entities. *International Journal of Geographic Information Science*, 17**:** 517-546.

Streit, U. and K. Wiesmann, 1996. Problems of Integrating GIS and Hydrological Models. *Spatial Analytical Perspectives on GIS in the Environmental and Socio-Economic Sciences.*, M. Fischer, H. Scholten, and D. Unwin, Eds., Taylor and Francis, 161-173.

Sunyé, G., F. Pennaneac'h, W. Ho, A. Le Guenned, and J. Jézéquel, 2001: Using UML Action Semantics for Executable Modeling and Beyond. *Proceedings of Advanced Information Systems Engineering: 13th International Conference, CAiSE 2001, Interlaken, Switzerland, June 4-8, 2001. K. R. Dittrick, A. Geppert, M. C. Morrie (eds). Lecture Notes in Computer Science, Springer-Verlag, Heidelberg.*

Thériault, M., C. Claramunt, and P. Y. Villeneuve, 1999: A Spatio-temporal Taxonomy for the Representation of Spatial Set Behaviours. *Spatio-Temporal Database Management, International Workshop, Edinburgh, Scotland, September 10-11. Proceedings: Lecture Notes in Computer Science 1687 Springer.*

Thomasson, A. L., 2001. Geographic Objects and the Science of Geography. *Topoi*, 20**:** 149-159.

Thornes, J. B. and D. Brunsden, 1977. *Geomorphology and Time*. John Wiley & Sons, New York.

Tryfona, N. and C. S. Jensen, 1998. Conceptual Data Modeling for Spatiotemporal Applications, Chorochronos Technical Report CH-98-08.

Tryfona, N. and D. Pfoser, 2001: Designing Ontologies for Moving Objects Applications. *Workshop on Complex Reasoning on Geographical Data*, Cyprus, December 1.

Tversky, B., 1990. Where Partonomies and Taxonomies Meet. *Meanings and Prototypes: studies in linguistic catogorization*, S. L. Tsohatzidis, Ed., Routledge.

Uschold, M. and M. Gruninger, 1996. Ontologies: principles, methods and applications. *Knowledge Engineering Review*, 11.

van Deursen, W. P. A., 1995. Geographical Information Systems and Dynamic Models, Faculty of Spatial Sciences, University of Utrecht.

Van Dyke Parunak, H., R. Savit, and R. Riolo, 1998: Agent-Based Modeling vs. Equation-Based Modeling: A Case Study and Users' Guide. *Proceedings of Multi-agent Systems and Agent-based Simulation (MABS'98)*, Springer, LNAI 1534.

van Ingwagen, P., 2000. Temporal Parts and Identity Across Time. *Monist*, 83**:** 437-460.

van Oosterom, P., B. Maessen, and W. Quak, 2002. Generic Query Tool for Spatio-temporal Data. *International Journal of Geographic Information Science*, 16**:** 713-748.

Varzi, A. C., 2001. Philosophical Issues in Geography - An Introduction. *Topoi*, 20**:** 119-130.

Veith, T. L., M. S. Srinivasan, and W. J. Gburek, 2003: Process Representation in Watershed-scale Hydrologic Models: an Evaluation in an Experimental Watershed. *First Interagency Conference on Research in the Watersheds, October 27- 30 2003, Benson, Arizona.*

Visser, U., H. Stuckenschmidt, G. Schuster, and T. Vögele, 2002. Ontologies for Geographic Information Processing. *Computers & Geosciences*, 28**:** 103-117.

Wachowicz, M., 1999. *Object-Oriented Design for a Temporal GIS*. Taylor & Francis, London.

Ward, R. C. and M. Robinson, 2000. *Principles of Hydrology*. McGraw-Hill Publishing Company, Maidenhead.

Waters, N., 2002. Modeling the Environment with GIS: A Historical Perspective. *Geographic Information Systems and Environmental Modeling*, K. C. Clarke, B. O. Parks, and M. P. Crane, Eds., Prentice-Hall.

Weber, C., 1998. How to Deal with Water Supply Pollution with GIS and Multi-agent Systems. *Innovations in GIS 5*, S. Carver, Ed., Taylor & Francis, 89-97.

Whitehead, A. N., 1969. *Process and Reality: an essay in cosmology*. The Free Press, New York.

Whitehead, A. N., 1998. Critique of Scientific Materialism. *Philosophers of Process*, D. Browning and W. T. Myers, Eds., Fordham University Press, 274-288.

Winter, S. and S. Nittel, 2003. Formal Information Modelling for Standardisation in the Spatial Domain. *International Journal of Geographic Information Science*, 17**:** 721-741.

Worboys, M. F., 2001: Modelling Changes and Events in Dynamic Spatial Systems with Reference to Socio-Economic Units. *Life and Motion of Socio-Economic Units*.

Yuan, M., 1996: Temporal GIS and Spatio-Temporal Modeling. *Proceedings of the 3rd International Conference on Integrating GIS and Environmental Modeling, Sante Fe, New Mexico, USA*.

Yuan, M., 2001. Representing Complex Geographic Phenomena in GIS. *Cartography and Geographic Information Science*, 28**:** 83-96.

Zeigler, B. P., H. Praehofer, and T. G. Kim, 2000. *Theory of Modeling and Simulation: integrating discrete event and continuous complex dynamic systems*. Academic Press, San Diego.

Zhang, W. and G. J. Hunter, 2000. Temporal Interpolation of Spatially Dynamic Objects. *Geoinformatica*, 4**:** 403-418.