# Modeling Discrete Processes Over Multiple Levels Of Detail Using Partial Function Application

Paul WEISER [a] and Andrew U. FRANK [a]
[a] *Technical University of Vienna*
*Institute for Geoinformation and Cartography,*
*Gusshausstr. 27-29*
*1040 Wien*

**Abstract.** Despite considerable research efforts, process ontologies are still less advanced than their static counterparts. Often, models consider only one fixed level of detail. This work presents a formalism how to represent multiple levels of detail in discrete processes. This is one important step towards a sophisticated process ontology.

**Keywords.** Level of Detail, Process Ontology, Partial Function Application

## 1. Introduction

About 2500 years ago, Heraclitus of Ephesus noted the most intriguing feature of the world we live in: "Nothing endures but change". Processes shape the environment and our perceptions of it on a daily basis. Some are barely noticeable because they unfold over long periods of time and affect large regions of space (e.g. geological processes). Other processes happen quickly and are restricted to a rather small area of space making them more perceptible.

From a modeling perspective we can look at processes from two different perspectives. Either, we understand a process as a form of continuous change, or we define it as a discrete succession of states. The former can be understood as a function

$$f(x, y, z, t) = a \tag{1}$$

that takes three-dimensional coordinates and time as its arguments, returning a value $a$ at any arbitrary moment. The latter is usually modeled using a finite state automaton that accepts a sequence of certain states at certain points in time, e.g.,

$$s(t0), s(t1), ..., s(tn). \tag{2}$$

In its most general definition, process is the changing of a state. There are different conceptualizations of processes, depending on the context. The representation of continuous change is of particular interest to Physical Geography and it has been shown that

**Table 1.** Different types of GIS result in different types of queries

| **Type of GIS** | **Possible queries** |
|---|---|
| Static GIS | Where is feature X located and what are its properties? |
| Snapshot GIS | In area X, what is the increase of wasteland over the last 10 years? |
| "GIS with Object-Lifes" [7] [1] | When has feature X been destroyed? |
| Dynamic GIS | What has caused the change in feature X? |

physical processes can be modeled using Partial Differential Equations [6]. Human planning (e.g., a trip) is often represented as a discrete process involving physical objects. Socially constructed processes involve social objects and instantaneous state changes, e.g. marriage [9].

In this contribution, we focus on the representation of processes at a discrete level. For example, our daily trip to work with public transportation (e.g., train ride) is a discrete physical process. We carry out particular steps in order to get from point A to B. Also, at every stop the traveler has to decide to either get off or stay on the train. What happens between two stops is not of particular interest.

## 2. Problem Statement

The real world is infinitely complex and impossible to describe in all detail. Geographic Information Systems (GIS) abstract and model parts of the reality in order to make it possible to retrieve information from a particular domain. GIS have evolved to a point where it is possible to retrieve almost any information of representations of static phenomena. This helps to answer questions related to "Where" something is located, but falls short of answering "When" something happened or will happen [3].

Despite considerable research efforts spent on the integration of time, process ontologies are still far less advanced than their static counterparts. For an extensive overview on the historical development of temporal aspects in GIS see [12]. The full benefit of dynamic GIS can be seen if we consider typical queries a user could carry out to answer a particular question (see Table 1).

Also, information that can be retrieved is often fixed to one level of detail. Usually, we cannot adjust the amount of detail we want to retrieve. We simply get the complete information stored in a dataset, no matter if we need all of it or not. Ideally, we can get the level of detail we need depending on the task we carry out. This stands in contrast to the fact that a model needs to represent the information to be depicted in full detail. Using the example of public transportation, to build a useful model we would need to represent all stations, lines, trains, and timetables. For a passenger, however, to successfully get from point A to B only parts of the modeled information is enough.

The idea of representing multiple levels of detail in the context of navigation has been thoroughly investigated by [10]. The model, however, is static and describes multiple levels of abstraction for objects such as lanes and intersections, but does not consider processes. Our contribution is to represent process descriptions over multiple levels of detail, as they can be found by using an electronic trip planner.

## 3. Example

A visitor needs to go from the Technical University to Vienna International Airport. She uses an electronic trip planner, suggesting the following route:

> Take the subway U4 at Vienna Karlsplatz on track 1 leaving at 12.10 p.m., get off at station Wien-Mitte, get on S7 on track 3 leaving at 12.20 p.m., get off at Vienna International Airport.

This example description has both spatial and temporal information. Additionally, information on the different transport modes are displayed. Now consider the following:

> Take the subway U4 at Vienna Karlsplatz, get off at station Wien-Mitte, get on train S7, get off at Vienna International Airport.

This one has obviously less detail, because it does neither mention departure times nor what track the train leaves. Such a description would be perfectly fine if someone is more flexible in terms of when to take a train. In this particular case, it is sufficient to know what trains to take and where to get off. Now consider the next, even less detailed description:

> In order to get to the Airport, take the train at Vienna Karlsplatz.

In this example, the information is stripped down to the point where the person gets only told that she has to take a train that leaves at Vienna Karlsplatz. There is no indication on specific lines or possible changes.

## 4. Formalization

If we take a look at the example descriptions mentioned in section 3 we can observe the following:

1. The "same" process can be represented over several levels of detail.
2. Descriptions mention both processes (e.g., board) and objects (e.g., train).
3. Processes become more detailed if we "feed" them more objects. For example, "to board a train" is more general than "to board a train at Karlsplatz".

Having made these observations, we can try to model processes and their objects on a very detailed level, i.e., using a finite state automaton. We define all processes that take place in our model and specify the objects that are needed. Then, we only partially evaluate the functions depending on the level of detail we want to depict.

To illustrate, we define the process *board* as a function that takes a mode of transport, a station, and time as arguments and returns a state change of the world. The agent (traveler) can also be understood as an argument to the function, though for simplicity reasons this was not considered here. Note that the syntax of the executable code is based on Haskell, a purely functional programming language [8].

```
data Mode = Train | Subway | Bus
data Station = Karlsplatz | Airport | Station X | ...

board :: Mode -> Station -> time -> stateChangeWorld
```

The returned argument "stateChangeWorld" is itself a function defined as:

```
stateChangeWorld :: StateOfWorld -> StateOfWorld
```

Thus, any process changes the world in terms of its state and can be defined as:

```
process :: arg1 -> arg2 -> ... -> stateChangeWorld
```

To reduce detail we could now define a new function *boardTrain* and only partially evaluate it. For example:

```
boardTrain :: Station -> time -> stateChangeWorld
boardTrain = board Train
```

Partial function evaluation is a well known method in functional languages. For any function with multiple arguments one can fix one argument and obtains a new function with one argument less. For example, the function *add* returns the sum of two integer values:

```
add :: Int -> Int -> Int
add x y = x + y
```

It can also be partially evaluated and if applied to an integer value this new function *add2* increments the value by two:

```
add2 :: Int -> Int
add2 = add 2
```

In the before mentioned example, we have fixed the mode of transport (train) for *board*. The resulting function has only arguments for the station and the time, i.e., where and when we board the train. We can create similar functions by feeding different modes of transport (e.g., subway). We could also add more detail and set the station:

```
boardTrainKarlsplatz :: time -> stateChangeWorld
boardTrainKarlsplatz = boardTrain Karlsplatz
```

If we want, we can set the time as well and get a new function defining the process at its most detailed level:

```
boardTrainKarlsplatz12PM :: stateChangeWorld
boardTrainKarlsplatz12PM = boardTrainKarlsplatz "12PM"
```

Partial evaluation is one form of producing descriptions of processes at different levels of detail. The order in which specializations from the most general (and most detailed) descriptions are produced can be freely selected. Other generalizations could be produced, e.g., the objects involved.

## 5. Conclusion and Future Work

The integration of process ontologies is an important step towards a fully dynamic GIS. This requires several key aspects, i.e., useful process models need to [11]:

- be capable of depicting dynamic phenomena and allow for an application-independent implementation [2].
- be both scalable and combinable to build more complex systems [12].
- capture more than just one level of detail.
- be able to account for concurrent and interleaved processes.

The mentioned formalization is by far not complete. As an underlying model we need to fully formalize a finite-state automaton depicting the entire domain. Using the example process *board* we have shown how different levels of abstractions for processes can be conceptualized. In order to test our model we will have to define the connections between the complete (automaton) and the incomplete information (Conceptual Level of Detail). The approach presented here, i.e., partial evaluation of functions as generalization, produces different levels of detail in travel instructions, as shown above. Modelling the instructions at different levels of detail gives a formal model of the amount of useful information received [4] [5].

## References


[1] K. Al-Taha and R. Barrera : Identities through time. In Proceedings of International Workshop on Requirements for Integrated Geographic Information System, 1994, New Orleans, Lousiana

[2] M. Craglia, M.F. Goodchild, A. Annoni et al. : Next/Generation Digital Earth. A Position Paper from the Vespucii Initiative for the Advancement of Geographic Information Science. In International Journal of Spatial Data Infrastructures Research, 2008, 3:146-167

[3] A.U. Frank: GIS for Politics, Paper presented at the GIS Planet '98, Lisbon, Portugal, 1998

[4] A.U. Frank: Spiele als Algebra, Paper presented at the 24th International Wittgenstein Symposium, 2001, Kirchberg a. Wechsel.

[5] A.U. Frank: Pragmatic Information Content - How to measure the Information in a Route Description, In Perspectives on Geographic Information Science, 2003, 47-68

[6] B. Hofer and A.U. Frank: Composing Models of Geographic Physical Processes, In Spatial Information Theory, Springer Berlin / Heidelberg, 2009, 5756, 421-435

[7] K. Hornsby and M. Egenhofer: Identiy-Based Change: A foundation for spatio-temporal knowledge representation. In International Journal of Geographic Information Science, 2000, 14:3,207-224

[8] S.P. Jones: Haskell 98 Language and Libraries. The Revised Report. 2002

[9] J.R. Searle: The Construction of Social Reality, 1995, Penguin Press, London, UK

[10] S. Timpf, G.S. Volta, D.W. Pollock and M.J. Egenhofer : A Conceptual Model of Wayfinding using Multiple Levels of Abstraction. In Lecture Notes in Computer Sciene, 1992, 639:348-367

[11] P.Weiser and A.U. Frank : Dynamic GIS - The final frontier?, Extended Abstract, 2010, GI-Forum, Salzburg

[12] M. Worboys: Event-oriented approaches to geographic phenomena. In International Journal of Geographic Information Science, 2005, 19:1-28