Time Interval-Based Data

David R. Mikesell and John L. Pfaltz
University of Virginia
Department of Computer Science
Charlottesville, VA 22906
{drm9f,jlp}@cs.virginia.edu

Abstract

Our application is a global change simulation that models the carbon cycle. This model operates on dozens of parameters to calculate a measure of biological activity called net primary productivity. Data used as inputs to this model span dramatically different periods of time. To enable our application to reconcile differences in time scales, we have developed an interval-based representation of data. In this paper, we describe this representation, define operations on time intervals, and discuss how these operations affect parameters associated with time intervals.

1 Introduction

In scientific studies, as well as in many other fields, physical measurements are typically tied to temporal and spatial coordinates. A temperature measurement is useful only when we know when and where the measurement was taken. A precipitation measurement has greater meaning when we know over what time period the precipitation fell. We are developing an interval-based representation of temporal-spatial data that will use the ADAMS object-oriented database system [PF93] [PHF98] to implement a global primary productivity model [WSE95]. This representation includes a set of temporal operators

that are used to manipulate interval-based data.

The idea of an interval-based time representation is not new. Allen's contributions [All83] [All91] [AF94] have been the foundation of other developments in this area [LG93]. It is quite different from those who treat time as a sequence of time ticks and represent time using timestamps ("instants are points in time, intervals are sequences of temporally consecutive points" [CDI+97], p. 182). In other formulations [ATSS93], intervals are only a derivative concept consisting of that time between two timestamps that denote the beginning and end of the interval. But our approach, which associates temporal intervals with scientific and environmental processes, makes the interval central. Time ticks and timestamps are almost incidental.

The global primary productivity application models the global carbon cycle [PPE+90]. This model uses data gathered from different sources and recorded over different time scales to calculate net primary productivity (the total carbon generated or consumed in a specific ecosystem). Some of the data are recorded from actual measurements, while other data are estimations or derived values. Some data may be complete, while other data points may be missing. Our challenge was to develop a system of describing and manipulating temporal data such that we could maximize the utility of the data we have available, and to make reasonable estimations for missing data.

2 An Interval-Based Representation of Time

Processes can be considered in terms of intervals of time. If we consider a process to be a series of transitions, beginning from some start state and ending in some final state, then the associated time interval can be considered to be the span of time needed to complete the process. Time intervals are bounded by their endpoints. However, it is not necessary for the endpoints to be points of clock time. There may be cases where our interval endpoints are not known, or are purely artificial. For example, in a chemical process, a reaction may take only microseconds to complete. Assigning clock values to the start and end of this reaction would most likely be artificial, and probably would not be beneficial. A more reasonable approach would be to consider the interval as spanning the time to complete the reaction. Time interval endpoints can simply be considered as the start and end of some process, independent of any clock time measurement.

We consider time to have the following properties:

- Time is partially ordered.
- Time is not dense (if $t1 > t3$, then there may not exist $t2$ such that $t1 > t2 > t3$).
- Comparison and difference operators are well defined.
- If $t1 > t2$, then $t2 - t1 = 0$. Negative time values are not allowed.
- Display operators are necessary to map time values to recognizable forms.

Obviously, for some operations on time intervals, total ordering is necessary (*i.e.* time interval composition). But in many applications, partial ordering is the best we can achieve. In distributed computing applications, for example, partial ordering is

easily achieved, but establishing a total ordering is much more difficult due to clock synchronization and communication delays [Lam78] [Mat89].

The second point is also counterintuitive. However, this follows from our definition of time intervals in terms of processes. If time interval endpoints are taken to be the start and end of some process, and this process cannot be subdivided, then subdividing the underlying time interal makes no sense. There exists no identifiable $t2$ such that $start > t2 > end$. Moreover, we are modelling temporal *data*. Suppose we are representing hourly rainfall. To assert some intermediate interval for which we have no data is meaningless.

Having defined properties of time with respect to processes, we can now develop a representation of time intervals. We consider time intervals to have the following properties:

- Any interval can be recursively subdivided into subintervals.
- For any interval, the time-valued functions *begin*() and *end*() are well defined, and the result of *begin*() is always less than or equal to the value of *end*().
- Associated with every interval is a non-negative quantity called the *duration*.
- The *class* of an interval is defined by the process parameters associated with the interval.

We assume any interval can be arbitrarily subdivided. There is no smallest interval, but, for practical purposes, this subdivision is limited by the precision of the representation, or by the nature of the process that we are subdividing.

The functions *begin*() and *end*() return a time value that represents the beginning and the end of the time interval. For any time interval, these must return a valid value.

However, the time values do not need to be clock time stamps; they can be any consistent measure of logical time.

Intervals cannot have a duration less than zero since a negative time interval does not make sense, either from a traditional time standpoint or a process standpoint. However, it is possible to have a time interval with duration equal to zero, where the start and end points are identical. This would represent an instant in time.

A time interval may have associated process parameters[1], and it is the combination of the interval and the set of its parameters that constitute the **class of the interval**. For example, if we have an interval I, and the parameters *average temperature* and *total rainfall* are associated parameters, then I would belong to the class of intervals that are associated with only those two process parameters.

The operations we will define on intervals are of two types: **strictly temporal**, and **class-dependent**. Strictly temporal operations may be performed on any interval regardless of class, and the result of the operation is independent of the class of the operands. By contrast, the class of the operands affects the result of a class-dependent operation.

Notational conventions used in this section will be:

- I: time interval
- t: time instant
- P: parameter

[1]Often, the process determining the interval is a data sensing/recording process

- C: class
- $Param(C)$: The set of parameters that are members of class C.
- δ: duration operator $(\delta(I) \rightarrow numeric)$
- $*$: superimposition operator $(I * I \rightarrow I)$
- $\circ$: composition operator $(I \circ I \rightarrow I)$
- ϕ: decomposition operator $(I\phi_n \rightarrow I_1, I_2, ..., I_n)$
- $-$: complement operator $(I - I \rightarrow I)$
- $=, \neq, >, <, \geq, \leq$: comparison operators $(t(operator)t \rightarrow Boolean)$
- $-$: difference operator $(t - t \rightarrow I)$
- $I.p_k$: A specific parameter of a particular interval.

Many of our operators are defined in terms of the duration function. The duration function is a strictly temporal operator.

Duration: $\delta(I) = end(I) - begin(I)$

Duration is the measure of the length of a time interval that uses the previously defined functions $end()$ and $begin()$. The duration can be a measure of either clock time or logical time, or a simply a measure of progress in a process.

The following are definitions of the class-dependent operations. As noted above, the result of a class-dependent operation depends on the class of the operands. These operations affect both the temporal aspects of intervals as well as their associated parameters. We will define and discuss the temporal effects of the operations, then we will discuss how each of these operations affect associated parameters.

Intervals can be recursively decomposed.

Decomposition (ϕ_n): For all I where $\delta(I) > 0$, I can be decomposed into n equal

subintervals ($n > 1$) such that:

- I_j *meets* $I_{j+1} = true$ (defined as $end(I_j) = begin(I_{j+1})$), $1 = j < n - 1$.
- Each subinterval $I_1 ... I_n$ belongs to the same class as I and therefore has the same parameter set as I.
- $\delta(I) = \delta(I_1) + \delta(I_2) + \ldots + \delta(I_n)$.
- $I_1 \ldots I_n$ are composible into I.
- $\delta(I_j)$ is computable, since we know the values of $begin(I_j)$ and $end(I_j)$.

A decomposition creates a set of subintervals, all of equal duration, which completely cover the span of the parent interval. Each subinterval has the same set of associated parameters as the parent interval.

Intervals can also be composed.

Composition ($\circ$): Given any two intervals I_j and I_k, where I_j belongs to the class C_j and I_k belongs to the class C_k, their composition $(I_j \circ I_k)$ yields I such that:

- $begin(I) = min(begin(I_j), begin(I_k))$.
- $end(I) = max(end(I_j), end(I_k))$.
- I belongs to the class C_I with parameters $(Param(C_j) \cap Param(C_k))$.
- $\delta(I)$ is computable, since we know the values of $begin(I)$ and $end(I)$.

Note that there is no temporal restriction on the relationship between intervals that are to be composed. Two intervals involved in a composition may or may not meet, they may or may not overlap, they may or may not be equal, and in fact they may even gap. There are also no class restrictions on composing intervals. Intervals from different classes may be composed, and the resulting interval may belong to an entirely different class than the classes of either of the composing intervals.

The superimposition of two intervals yields an interval that corresponds to the overlap of the two original intervals.

Superimposition $(*)$: Given any two intervals I_j and I_k, where I_j belongs to the class C_j and I_k belongs to the class C_k, and $begin(I_j) < begin(I_k)$ and I_j *meets* $I_k = true$ or I_j *overlaps* $I_k = true$, $I_j * I_k$ yields I such that:

- $begin(I) = max((begin(I_j), begin(I_k))$.
- $end(I) = min(end(I_j), end(I_k))$.
- I belongs to the class C_I with parameters $(Param(C_j) \cap Param(C_k))$.
- $\delta(I)$ is computable, since we know the values of $begin(I)$ and $end(I)$.

The superimposition of two intervals that meet will be an interval with a duration = 0. Intervals that gap cannot be superimposed.

Complement $(-)$: Given any two intervals I_j and I_k, where I_j belongs to the class C_j and I_k belongs to the class C_k, and I_k is either a prefix or a suffix of I_j, $I_j - I_k$ yields I such that:

- If I_k is a prefix of I_j, then: $begin(I) = end(I_k)$, and $end(I) = end(I_j)$.
- otherwise, I_k is a suffix of I_j, so: $begin(I) = begin(I_j)$, and $end(I) = begin(I_k)$.
- I belongs to the class C_I with parameters $(Param(C_j) \cap Param(C_k))$
- $\delta(I) = \delta(I_j) - \delta(I_k)$.
- $I \circ I_k = I_j$.

In order to simplify the use and implementation of the complement operator, I_k must be either a prefix or suffix of I_j.

Next, we define the other strictly temporal operators. As stated previously, strictly temporal operations may be performed on any interval regardless of class, and the result

of the operation is independent of the class of the operands. The following are Boolean relationship operators that test the temporal relationship between two time intervals.

equals: I_j *equals* $I_k \rightarrow$ *Boolean*
true if $\delta(I_j * I_k) = \delta(I_j) = \delta(I_k)$.

meets: I_j *meets* $I_k \rightarrow$ *Boolean*
true if $\delta(I_j * I_k) = 0$, and $\delta(I_j \circ I_k) = \delta(I_j) + \delta(I_k)$.

In order for two intervals to *meet*, they cannot *overlap*, and there must be no gap between them.

overlaps: I_j *overlaps* $I_k \rightarrow$ *Boolean*
true if $\delta(I_j * I_k) > 0$, and $\delta(I_j \circ I_k) < \delta(I_j) + \delta(I_k)$.

Two intervals *overlap* if the duration of their superimposition is not zero, and if the duration of their composition is greater than the sum of their durations. Intervals that span exactly the same time are not said to *overlap*, but are instead *equal*.

gaps: I_j *gaps* $I_k \rightarrow$ *Boolean*
true if I_j *overlaps* $I_k = false$, and I_j *meets* $I_k = false$.

If I_j and I_k gap, then there exists some interval I' with $\delta(I') > 0$, such that $end(I_j) = begin(I')$, and $begin(I_k) = end(I')$.

includes: I_j *includes* $I_k \rightarrow$ *Boolean*
true if $I_j * I_k = I_k$, and $I_j \circ I_k = I_j$.

For the definitions of the $prefixes$ and $suffixes$ operations, I_j is the larger interval, while I_k is the subinterval.

prefixes: I_k *prefixes* $I_j \rightarrow$ *Boolean*
true if I_j *includes* I_k, and $begin(I_j) = begin(I_k)$.

suffixes: I_k *suffixes* $I_j \rightarrow$ *Boolean*
true if I_j *includes* I_k, and $end(I_j) = end(I_k)$.

Figure 1 displays a graphic representation of these relationships. It should be noted that time instants (as ticks or stamps) play no role in these boolean operators.

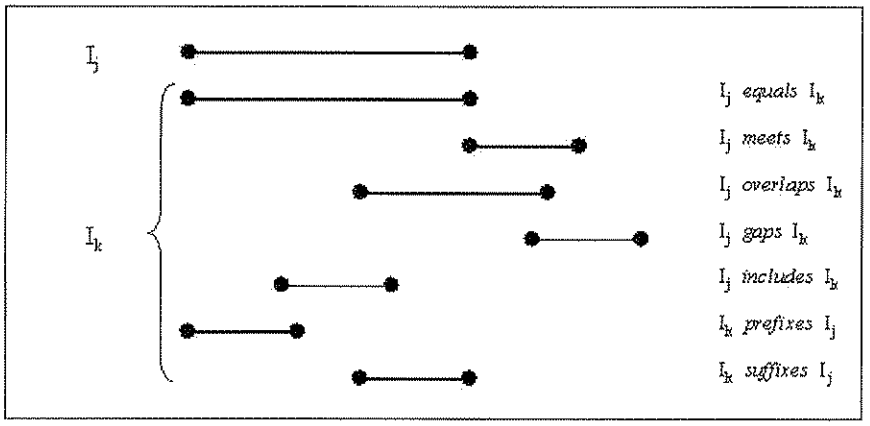

Figure 1: Boolean Operations on Time Intervals

3 Associated Data and Interval Classes

Environmental data is generally location and time-dependent. Weather data is reported as a set of parameter values measured at a particular place at a specific date or time. Similarly, researchers in other fields such as oceanography, glaciology, and limnology report data that are bound to a time and place. When examining data, the temporal

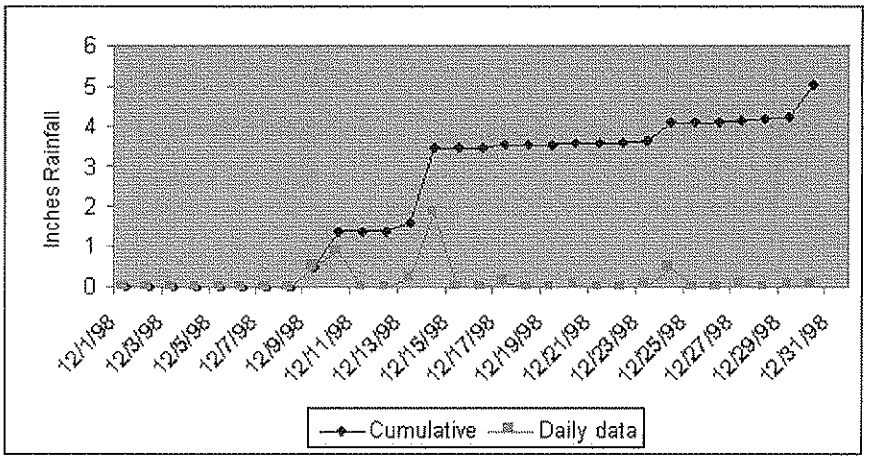

Figure 2: Cumulative and Daily Precipitation for Richmond, Virginia, December 1998

and spatial dimensions provide a context to permit the comparison of data points and the development of interpretations of the data. Time intervals can provide the temporal dimension necessary to provide such a context for data.

We have already briefly mentioned interval classes. The class of a time interval is defined by the set of process parameters that are associated with it. For example, if we have an interval, spanning from t_1 to t_2, and associated with that interval are values of average temperature and total precipitation, then its class includes all intervals that contain the exactly the same parameter set. The class is not dependent upon any property of the interval other than its associated parameters, so intervals may have dramatically different durations but still belong to the same class.

We classify data as one of two types, **cumulative** and **distributive**. We consider

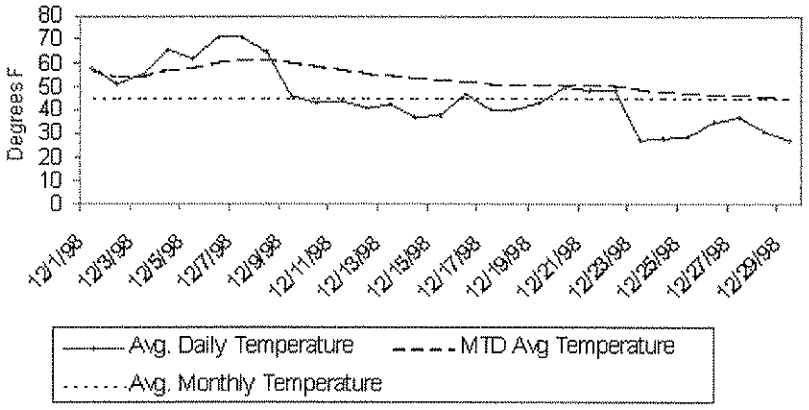

Figure 3: Daily and Average Monthly Temperature for Richmond, Virginia, December 1998

that cumulative data are those measurements that are typically examined as totals over a period of time. For example, total monthly rainfall is a cumulative parameter, since it is simply the sum of the daily rainfall measurements. Figure 2 shows an example of cumulative precipitation over one month. Similarly, we consider distributive data to be those measurements that are typically examined as an average over a time period, e.g. monthly average temperature. Figure 3 shows an example of temperature measurements over one month. This distinction between parameter types is necessary since our temporal operators affect parameter values differently depending on whether the parameter is distributive or cumulative.

We can consider constant data as a special case of distributive data, where the dis-

tribution of values over time is uniform. If, however, the constant represents an extreme measurement,this uniform assumption is seldom correct.

In our global change model, we are using data from many different sources, and the time intervals that the data cover are often inconsistent, or do not correspond neatly to the intervals we require for the model. For example, if we need data on three parameters (*i.e.* average temperature, average hours of sunlight, and total precipitation) for December 1998, but all that is available is daily measurements for one parameter, monthly measurements for another parameter, and seasonal measurements for the third parameter, it is going to be difficult to extract the data we need, and it will also be difficult to assess how confident we can be with the our results. Our interval operators provide a mechanism to obtain the data needed by the global change model.

4 Manpulating Inverval-Based Data

To make interval operators useful, the operations must yield the best possible data. Observed data are naturally preferred, but in their absence, we must be able to obtain either derived values or a best estimate. An operation that generates a parameter value for any interval must not alter existing observed data. In this section, we present methods of determining parameter values that are generated as the result of applying class-dependent operators on time intervals.

In order to apply temporal operators to physical measurements, we need to know how these measurements relate to time. Some relationships are intrinsic to the parameter as indicated by its dimension (*e.g.* velocity in feet/second). Other relationships are artifacts of the measuring process, as in rainfall measured in, for example, inches per day (or month, or year). Our approach is to describe a set of operations that can be applied to process parameters, and then determine, based on the nature of those process parameters, which operations apply. The rules we abide by in our operations are:

1. Observed data are not modified.
2. Derived data are flagged as such.
3. Estimated data are flagged as well.

Previously, we discussed two classifications of data: cumulative and distributive. A characteristic of cumulative data is the parameter value in any given subinterval will never be greater than that parameter value in its parent interval. However, other than that, we often are not certain as to the nature of the change over time. If we have a monthly rainfall measurement of 30 inches, unless we have additional information we cannot know if the 30 inches fell in a day or if one inch fell every day for 30 days, or if it fell in some other manner.

Distributive data is similar in that often we do not know the nature of the distribution of the parameter values over time. Our approach is to use what data we do have to make an approximation, and flag it as such. Here, we present an example of how a decomposition might affect associated parameters:

As defined earlier, decomposition is the division of a time interval into n subintervals. The process to perform a decomposition is as follows:

1. Create a set S of n subintervals (I'_j) from the parent interval (I) by determining the temporal boundaries on the new subintervals. The boundaries are calculated using a simple division of I.
2. Assign the class of all subintervals I'_j to be equal to the class of I. All subintervals I'_j now have the same parameter set as I.
3. For each subinterval ($I'_j, 1 \leq j \leq n$) in S, determine if there exists any other interval (I'') in our database with the same temporal and spatial coordinates. If there exists a parameter p_k in I'' that is also in I'_j and the value of p_k is not flagged as estimated, then the value of $I''.p_k$ is assigned to $I_j.p_k$.
4. After assigning parameter values for each of the subintervals from existing data, we then estimate the values of the remaining parameters. Where a newly estimated value is of higher quality than the existing value, or if no previously determined value exists, the newly estimated value is assigned to that parameter. If the newly estimated value is of lesser quality than the existing value, then the newly estimated value is discarded.

For the estimation of cumulative data under decomposition, we assume that parameter values increase linearly with time. For each parameter p_k, if n is the total number of subintervals in the decomposition, j is the number of subintervals that were assigned values for p_k in step 3 above, S is the sum of the values of p_k assigned in step 3, and $I.p_k$ is the value of parameter p_k for the parent interval I, then we can estimate the value of p_k in each subinterval I' that was not assigned a value in step 3 above:

$$I'.p_k = \frac{(I.p_k - S)}{(n - j)}.$$

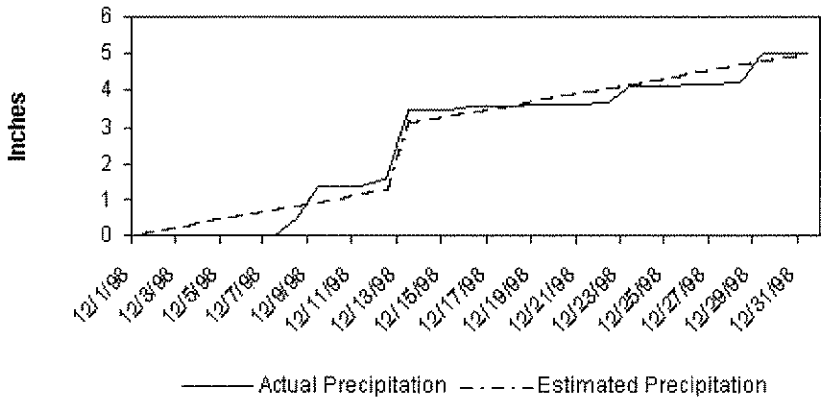

Figure 4: Actual and Estimated Daily Precipitation for Richmond, Virginia, December 1998

Example: For Richmond, Virginia, we have monthly total rainfall data and daily data for those days where the rainfall was at least 1 inch. We wish to decompose the interval corresponding to December 1998 into 31 daily intervals.

In December 1998, total rainfall in Richmond was 5.02 inches. On the 13th, there was 1.84 inches of rain. All other days had less than 1 inch of rain. We can estimate the daily rainfall in Richmond to be $\frac{5.02-1.84}{31-1}$, or 0.106 inches, except for the 13th where we know the rainfall to be 1.84 inches. Figure 4 shows graphically the results of this decomposition. With no daily data, the result of this decomposition is at best a crude estimate. As we include additional daily data, the estimate improves. In Figure 5, we show the result of the decomposition when all daily rainfall totals greater than 0.5 inches

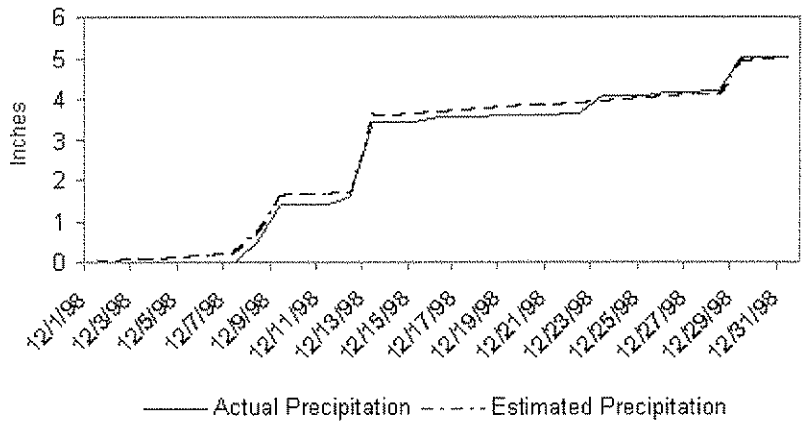

Figure 5: Actual and Refined Estimated Daily Precipitation for Richmond, Virginia, December 1998

are included in the calculation.

In our application, it is not unusual for our data to be incomplete in this manner. For example, we might have an annual rainfall measurement, and additional data only for catastrophic or unusual events such as hurricanes or monsoons. Other normal rainfall events may be missing. Since our global change model employs rainfall on a daily basis, we must estimate rainfall values as closely as possible. This mechanism provides a reasonable method of estimation.

For the decomposition of intervals that include distributive data, we perform a process similar to the above four steps. However, we use a different calculation to estimate the value of distributive data where the data set is incomplete. In the worst case, when we

have no data for the subintervals, we assign the same value for distributive data in the parent interval to all subintervals. In cases where we have some supporting data, we can refine the estimate. The assumption we make for decomposing intervals with associated distributive data is that the average of the values of a parameter for all subintervals will equal the value in the parent interval. For each parameter p_k, if n is the total number of subintervals in the decomposition, j is the number of subintervals that were assigned values for p_k in step 3 above, S is the sum of the values of p_k assigned in step 3, and $I.p_k$ is the value of parameter p_k for the parent interval I, then we can estimate the value of p_k in each subinterval I' that was not assigned a value in step 3 above:

$$I'.p_k = \frac{(I.p_k \times n) - S}{(n - j)}.$$

The other operations act in similar ways. Since there is such a rich set of operations and relations on time intervals, and since we need to be aware of how the data is affected by these operations, there remains much more work to be done in this area.

5 Conclusion

The global change simulation embraces 65,875 biologically active land elements, each with different process characteristics. Modeling this, even assuming a parallel database implementation as described in [PHF98], is computationally intensive. Our represen-

tation of process parameters with respect to natural process intervals is a key step to managing this computational complexity. While development of these operators is still in the early stages, we believe that they will provide additional flexibility in manipulating temporal data. In particular, we are exploring a re-entrant capability that allows selective subdivision of the simulation interval over specified critical regions.

References

[AF94] James F. Allen and George Ferguson. Actions and events in interval temporal logic. Technical report, The University of Rochester, July 1994.

[All83] James F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, November 1983.

[All91] James F. Allen. Time and time again: the many ways to represent time. *International Journal of Intelligent Systems*, 6(4):341–355, 1991.

[ATSS93] Khaled K. Al-Taha, Richard T. Snodgrass, and Michael D. Soo. Bibliography on spatiotemporal databases. *SIGMOD RECORD*, 22(1):59–67, March 1993.

[CDI+97] James Clifford, Curtis Dyreson, Tomas Isakowitz, Christian S. Jensen, and Richard T. Snodgrass. On the semantics of "now" in databases. *ACM Transactions on Database Systems*, 22(2):171–214, June 1997.

[Lam78] Leslie Lamport. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, 21(7):558–565, July 1978.

[LG93] Thomas D. C. Little and Arif Ghafoor. Interval-based conceptual models for time-dependent multimedia data. *IEEE Transactions on Knowledge and Data Engineering*, 5(4):551–563, August 1993.

[Mat89] Friedemann Mattern. Virtual time and global states of distributed systems. *Parallel and Distributed Algorithms*, pages 215–226, 1989.

[PF93] John. L. Pfaltz and James. C. French. Scientific database management with ADAMS. *Data Engineering*, 16(1):14–18, March 1993.

[PHF98] John L. Pfaltz, Russell F. Haddleton, and James C. French. Scalable, parallel, scientific databases. pages 4–11, Capri, Italy, July 1998.

[PPE+90] W. Post, T. Peng, W. Emanuel, A. King, V. Dale, and D. DeAngelis. The global carbon cycle. *American Scientist*, 78:310–326, 1990.

[WSE95] F. I. Woodward, T. M. Smith, and W. R. Emanuel. A global land primary productivity and phytogeography model. *Global Biochemical Cycles*, 9:471–490, 1995.