

# Generating Linear Orders of Text-Based Events

**Kathleen Hornsby**

National Center for Geographic  
Information and Analysis  
University of Maine  
Orono, ME 04469-5711  
khornsby@spatial.maine.edu

**Suzannah Hall**

Department of Spatial Information  
Science and Engineering  
University of Maine  
Orono, ME 04469-5711  
shall@spatial.maine.edu

## Abstract

Events described in textual narratives do not always occur in neat, chronological order but occur, for example, during or overlapping each other or as simultaneous events. Summarizations of narratives, however, benefit from a simpler, linear ordering of events. This paper describes an approach for modeling events in text as event intervals and for generating linear orders of event intervals, useful for the summarization of events or as the basis for question answering systems. Linear orders are derived through reducing the set of thirteen possible event interval relations to a set of only *before* or *equal* relations. The mapping of event interval relations into *before/after* sequences requires the support of additional constraints in order to preserve the original semantics of the events presented in the text and to derive plausible orders of events.

## 1 Introduction

The world is a dynamic place and the activities and actions that are part of our everyday experience include such phenomena as the flow of traffic on the morning drive to work, a person walking across a street, or the fluctuation of water bodies due to seasonal change. An interest in developing computational models that convey the dynamic aspects of the world has resulted in a more explicit focus on modeling *events*, i.e., actions that result in some change to an entity over time, such as the creation, elimination, or transmission of entities (Al-Taha and Barrera, 1994; Claramunt and Theriault, 1995; Claramunt and Theriault, 1996; Medak, 1999; Hornsby and Egenhofer, 2000). The segmentation of real-world happenings into events facilitates the processing and

conveying of natural language information to a user (Zacks and Tversky, 2001). An understanding of text is improved if the event structure of the text can be reconstructed. Alfonseca and Manandhar (2002) extract events based on verbs that represent actions, verbs associated with the verb *to be* which represent states, and occurrences of nouns that are specifications of verbs, such as *arrival* or *accident*. Events are anchored in time relative to either the time the text is written or to the main event, and reordered based on this anchoring and verb tenses. Indexing and information retrieval for current and historical reports on infectious disease outbreaks is improved through an approach where events relating to occurrences of infectious diseases are automatically extracted from news sources on the Internet (Grishman et al., 2002). These events are then used to populate a database with the view that coupling the text-based event extraction with a database approach, offers better indexing for reports on infectious disease outbreaks.

Reasoning about events commonly requires assembling the events into a sequence or order of events such that a temporal pattern of events becomes distinguishable (Frank, 1998) and more understandable. For many reasoning tasks involving events, users require a simple, *total order* of events, where for every pair of events, *A* and *B*, either *A is before B* or *B is before A* (or both) (Frank, 1998). The case where *A is before B* and *B is before A* describes the case where *A* occurs *at the same time as B*. This linear sequencing helps us to understand and communicate in a simpler fashion how events occur over time.

In a typical database, the values of one or more attributes of data may be ordered through queries to the database using a database query language such as SQL. Dynamic scenarios as captured in text-based narratives, for example, require alternative approaches to ordering, where orders are based on events or the relations among events even though the knowledge about these relations may be incomplete or uncertain (Pustejovsky et al.,

2003). This paper presents methods to generate linear orders of events from more complex orderings of events in text. These linear orders provide simpler, summarized views of the events in a narrative as well as a basis for event-based question and answer systems. Automatic text summarization strategies are necessary to support decision making from large textual narratives as well as the large number of information resources available via the Web (Mani and Maybury, 2001). Temporal semantics and events have been considered in the context of time-based summarization strategies that assist users to monitor changes in news coverage over time (Allan et al., 2001).

In this paper, *entities* refer to phenomena in the real world, and an *event* results in some change to an entity over time. Events are assumed to be linear with both a start point and an end point. Events that are ongoing, i.e., have no end, are not treated here, nor are pre-existing events, i.e., those that have no known start point.

The objective of this paper is to introduce an approach for automatically generating plausible linear orders of events from partially-ordered sets of event intervals drawn from text descriptions. The remainder of this paper is structured as follows: Section 2 describes events modeled as event intervals and the relations that hold between event intervals. Section 3 presents an approach to generating linear orders of event intervals where the set of 13 possible event interval relations are reduced to either *before* or *equals*. In Section 4 an example is introduced to demonstrate this approach. The next section further refines the ordering process by incorporating constraints derived from the semantics of the original event interval relations, and Section 6 uses the example scenario to illustrate the use of these semantics in the ordering process. Section 7 presents the conclusions and discusses future work.

## 2 Events and relations among events

Events are often modeled as being instantaneous such as, for example, an update to a bank account or the transmission of an electronic message, i.e., changes of state having no duration (Hinze and Voisard, 2002). Alternatively, events may be modeled as occurring over a period of time and therefore have duration. These events are typically associated with a specific point in time (Motakis and Zaniolo, 1995), usually the point at which the event finishes (Galton and Augusto, 2002). In linguistics and cognitive psychology, an event is most often modeled as occurring over a period of time (Larson, 1999; Pedersen and Wright, 2002), and human perceptions of this event time are of particular interest to researchers. For example, in a court case, a suspect's claim about events during a particular time period *ver-*

*sus* a witness' perception of the same events may affect the jury and make a difference to the outcome of a trial (Pedersen and Wright, 2002).

The events described in this paper are treated as intervals with a start point and end point, and are assumed to have some duration. For example, LowPressure-Moves refers to an event—the movement of a low pressure system—modeled as an interval. Text, such as a paragraph describing the weather over the past twenty-four hours, can be processed to yield a set of event-relation combinations. Our focus in this paper is not so much on event extraction from text but rather the methods to automatically generate a plausible linear order of events. As event intervals are assumed to be linear, Allen's temporal interval relations (Allen, 1983) describe the set of possible relations that hold among event intervals (Figure 1). For example, *scattered showers occur this afternoon before drier air arrives from the west* relates two events, *ScatteredShowersOccur* and *DrierAirArrives*, by the temporal interval relation *before*. This approach assumes an underlying linear model of time and excludes cyclic or branching time (Frank, 1998; Hornsby et al., 1999).

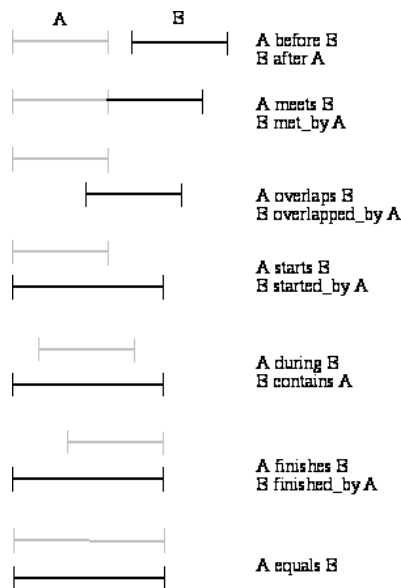


Figure 1. Thirteen event interval relations (after Allen 1983).

Retrieving information about event intervals that shows how events relate to each other is required for an understanding of how entities described in a narrative evolve over space and time. Generating *orders of events* is necessary such that useful information about events is presented for users, either as a summarization tool for large information resources or as a method for question answering.

### 3 Ordering events

To generate linear orders of events, the set of 13 possible event interval relations is reduced to a set comprising only *before* or *equal* relations. Given two event intervals, *A* and *B*, for example, if the start point of *A* is *before* or *equal* to the start point of *B* and the end point of *A* is *before* the end point of *B*, then *A* is *before B* in the linear order. This holds for the cases of: *A before B*, *A meets B*, *A overlaps B*, and *A starts B* (Figure 2a). Event *A* is also *before B* in cases where the start point of *B* is *after* the start point of *A*, and the end point of *B* is *before* or *equal* to the end point of *A*, as in the occurrence of *A contains B* and *A ended\_by B*.

Where the start point of *A* is *after* the start point of *B* and the end point of *A* is *before* or *equal* to the end point of *B*, then *B* is *before A* in the linear order. This holds for *A during B* and *A ends B*, as well as for cases, *A after B*, *A met\_by B*, *A overlapped\_by B*, and *A started\_by B*, where the start point of *B* is *before* or *equal* to the start point of *A*, and the end point of *B* is *before* the end point of *A* (Figure 2b).

<i>A before B</i>	□	<i>A before B</i>
<i>A meets B</i>	□	<i>A before B</i>
<i>A overlaps B</i>	□	<i>A before B</i>
<i>A starts B</i>	□	<i>A before B</i>
<i>A contains B</i>	□	<i>A before B</i>
<i>A ended_by B</i>	□	<i>A before B</i>

(a)

<i>A during B</i>	□	<i>B before A</i>
<i>A ends B</i>	□	<i>B before A</i>
<i>A after B</i>	□	<i>B before A</i>
<i>A met_by B</i>	□	<i>B before A</i>
<i>A overlapped_by B</i>	□	<i>B before A</i>
<i>A started_by B</i>	□	<i>B before A</i>

(b)

<i>A equals B</i>	□	<i>A equals B</i>
-------------------	---	-------------------

(c)

Figure 2. The set of event interval relations are reduced to (a) *A before B*, (b) *B before A*, and (c) *A equals B*.

Finally, where the start point of *A* is *equal* to the start point of *B* and the end point of *A* is *equal* to the end point of *B*, then *A equals B* in the linear order (Figure 2c). For these cases, the event intervals are considered to be simultaneous.

If all possible relations that hold between events are known through the narrative, then only one plausible linear order will result. More commonly, however, it may not be known for certain how each event interval is related to the other intervals. For these cases, *partial orders* exist, and the events in the narrative may correspond to multiple possible linear orders.

In these cases, without filtering or abstracting some of the events, the number of possible orders generated can easily become too large for a user to comprehend. A method for filtering or abstracting is necessary. One method of abstraction is to remove any events that occur at the same time as another event with a longer duration, and about which no other information is known. Thus any event that occurs *during* another event would be discounted if that event has no known relation to another event. In addition, events that *start* other events are of shorter duration than the events that they start, and may also be abstracted if no other relation exists between the *starting* event and a third event. An event that *ends* another event and about which no other information is known would also be abstracted. This filtering will prune many of the orders from the set.

The next section presents an example scenario where orders of events are abstracted from a short text.

### 4 Example Scenario

Consider a narrative describing vehicles traveling on a bridge and boat traffic maneuvering in the harbor below the bridge, as well as activities on land in preparation for a ferry's arrival:

*While the car was crossing over the bridge, a ferry passed underneath and an ambulance went rushing past. A plane flew over as the ferry passed under the bridge. As the ferry reached the dock on the other side of the bridge, a truck arrived to pick up goods from the boat.*

Events described in a narrative can be reduced to a set of event-relation combinations, where two event intervals are related by one relation. There are  $n$  event intervals and  $m$  relations in the set, where  $m \leq 13$  is the number of possible event interval relations.

In this example, there are  $n=6$  event intervals including *CarCrossesBridge*, *FerryUnderBridge*, *AmbulancePassesCar*, *FerryDocks*, and *TruckArrives*. There are also  $m=4$  (unique) relations generating the following event-relation combinations:

FerryUnderBridge *during* CarCrossesBridge  
 AmbulancePassesCar *during* CarCrossesBridge  
 FerryDocks *after* FerryUnderBridge  
 FerryUnderBridge *overlaps* PlaneFliesOver  
 FerryDocks *meets* TruckArrives

From this set, the event intervals are extracted and combined pairwise (Figure 3) such that a square matrix  $E$  is formed with rows  $i$  and columns  $j$ , where  $i=1..n-1$  and  $j=1..n-1$ . Cells in  $E$  are denoted as  $e_{ij}$ .

The matrix  $E$  is populated with the  $m$  relations that exist between the events in the set. Event-event combinations that do not exist in the set are represented by  $\sim$ . All inverse relations are included in the matrix, thereby allowing all information about an event to be captured by a single row. The inverse of an *equals* relation is another *equals*. If orders are generated from the example matrix at this point, 15 linear orders are possible. To avoid such large result sets, filtering is performed on the matrix. Each row of the matrix is checked for a single *during*, *starts*, or *ends* relation, and these rows are eliminated from the matrix. The row for *Ambulance-PassesCar* contains only a *during* relation, so this row and its corresponding column are eliminated. The revised matrix  $E_1$  (Figure 4) becomes the foundation for computing the orders.

Applying the mappings to *before* for the relations in the matrix  $E_1$  now yields a total of 3 possible orders:

- (1) *CarCrossesBridge before FerryUnderBridge before Ferry-Docks before TruckArrives before PlaneFliesOver*
- (2) *CarCrossesBridge before FerryUnderBridge before Ferry-Docks before PlaneFliesOver before TruckArrives*
- (3) *CarCrossesBridge before FerryUnderBridge before Plane-FliesOver before FerryDocks before TruckArrives*

A key aspect to generating the linear orders is that the result set comprises orders that are plausible, i.e., capture as closely as possible the semantics of the original text. The next section describes a method for invoking additional constraints that increase the plausibility of all orders that are derived. Keeping the plausibility of the automatically generated linear orders as high as possible, has the added benefit of reducing the number of linear orders that are generated.

	FerryUnderBridge	CarCrossesBridge	AmbulancePassesCar	FerryDocks	PlaneFliesOver	TruckArrives
FerryUnderBridge	$\sim$	<i>during</i>	$\sim$	<i>before</i>	<i>overlaps</i>	$\sim$
CarCrossesBridge	<i>contains</i>	$\sim$	<i>contains</i>	$\sim$	$\sim$	$\sim$
AmbulancePassesCar	$\sim$	<i>during</i>	$\sim$	$\sim$	$\sim$	$\sim$
FerryDocks	<i>after</i>	$\sim$	$\sim$	$\sim$	$\sim$	<i>meets</i>
PlaneFliesOver	<i>overlapped_by</i>	$\sim$	$\sim$	$\sim$	$\sim$	$\sim$
TruckArrives	$\sim$	$\sim$	$\sim$	<i>met_by</i>	$\sim$	$\sim$

Figure 3. Matrix based on a set of event-relation pairs

	FerryUnderBridge	CarCrossesBridge	FerryDocks	PlaneFliesOver	TruckArrives
FerryUnderBridge	$\sim$	<i>during</i>	<i>before</i>	<i>overlaps</i>	$\sim$
CarCrossesBridge	<i>contains</i>	$\sim$	$\sim$	$\sim$	$\sim$
FerryDocks	<i>after</i>	$\sim$	$\sim$	$\sim$	<i>meets</i>
PlaneFliesOver	<i>overlapped_by</i>	$\sim$	$\sim$	$\sim$	$\sim$
TruckArrives	$\sim$	$\sim$	<i>met_by</i>	$\sim$	$\sim$

Figure 4. Matrix of events and relations after abstraction

## 5 Applying additional constraints based on semantics of relations

Each event interval relation is associated with particular semantics that should be maintained in order to generate plausible linear orders. For example, the *meet* relation describes a scenario in which two event intervals, *A* and *B*, occur such that the start time of *B* is simultaneous with the end-time of *A*. No other events would be expected to occur between these two events in a resulting linear order. Combine *A meets B* with *A before C*, however, and one resulting linear order of events is *A before C before B*, in which event *C* occurs between *A* and *B*. Incorporating the semantics of the relations offers a way to increase the plausibility of linear orders of events. Based on this work, mapping rules are defined that reduce the thirteen event interval relations to either *before* or *equals*, and constraints are applied to allow the preservation of key semantics associated with any given event interval relation. These constraints provide the basis for mapping event intervals and relations to a linear order of events.

### 5.1 Semantics involving *during* and *contains* relations

The semantics associated with *during* and *contains* relations capture cases where one event begins and ends within the time that another event is occurring. Applying the mapping rules to a case where, for example, *A before B* and *C during B*, with no regard to the semantics of the *during* relation, returns *A before B* and *B before C*, i.e., the linear order, *A before B before C*. If, in addition, *D after A*, one linear order becomes *A before B before D before C* and the events that are originally related by *during* are no longer together. In order to preserve the semantics relating to *during* and *contains* for a linear order of events, therefore, a constraint is applied where any events related by *during*, are always sequential in the resulting linear order of events and no intermediate events can occur between them, i.e. *A during B*  $\square$  *B*  $\prec$  *A* and  $\square$  *C* | *B*  $\prec$  *C*  $\prec$  *A*, read *A during B* leads to *B before A* and there does not exist a *C* such that *B before C before A*. For cases where *A contains B*, *A contains B*  $\square$  *A*  $\prec$  *B* and  $\square$  *C* | *A*  $\prec$  *C*  $\prec$  *B*.

For cases where more than one event occurs *during* another event interval, for example if *C during A* and *A during B*, the events can be put in order and the semantics preserved. *C during A* reduces to *A before C* and *A during B* reduces to *B before A*, resulting in the order *B before A before C*. If it is also known that event *D* occurs *before A* and *during B*, the linear order then becomes *B before D before A before C*. In

this case, *B* and *A* are separated by event interval *D* since both *A* and *D* are *during* event *B* but *both* events cannot immediately follow *B*. Formally, *A during B*  $\square$  *B*  $\prec$  *A* except if  $\square$  *D* | *D during B*, *D before A* then *B*  $\prec$  *D*  $\prec$  *A*.

### 5.2 Semantics involving *meets*, *met\_by*, *starts*, *started\_by*, *ended\_by*, and *ends*

Applying the mapping rules to any of the relations *meets*, *starts*, or *ended\_by* results in the relation being replaced by *before*. I.e., *A meets B*  $\square$  *A before B*, *A starts B*  $\square$  *A before B*, and *A ended\_by B*  $\square$  *A before B*. In all of these cases, it is implausible that a third event would occur between events *A* and *B* in a linear order. A constraint is applied to prevent this, and thus when *R = meets, starts* or *ended\_by*, *A R B*  $\square$  *A*  $\prec$  *B* and  $\square$  *C* | *A*  $\prec$  *C*  $\prec$  *B*.

It is equally implausible that a third event would occur between two events related by an inverse of one of the above three relations, and a similar constraint is applied: when *R = met\_by, started\_by* or *ends*, *A R B*  $\square$  *B*  $\prec$  *A* and  $\square$  *C* | *B*  $\prec$  *C*  $\prec$  *A*. Because the constraints for *during* take precedence over those for all other relations, some exceptions to the above constraints are necessary. In the case of *meets* and *met\_by*, if, in addition to an event-relation pair that *meets*, for example *A meets B*, there are events that occur *during A*, additional rules are necessary. For these cases we allow events that *meet* to be separated by other events in the resulting linear order. For example, given *A meets B* and *D during A*, these event-relation pairs are mapped to *A before B* and *A before D*. Two linear orders result, *A before D before B*, and *A before B before D*. The latter order is implausible, however, because event *B* occurs between two *during*-events, *D* and *A*. To prevent this type of implausible order, an addition is made to the *meets* constraint, if  $\square$  *D* | *D during A* then *A*  $\prec$  *D*  $\prec$  *B*, or if  $\square$  *E* | *B during E* and not *A during E*, then *A*  $\prec$  *E*  $\prec$  *B*. Similarly, *A met\_by B*  $\square$  *B*  $\prec$  *A* and  $\square$  *C* | *B*  $\prec$  *C*  $\prec$  *A*, except if  $\square$  *D* | *D during B* then *B*  $\prec$  *D*  $\prec$  *A*, or if  $\square$  *E* | *A during E* and not *B during E*, then *B*  $\prec$  *E*  $\prec$  *A*.

One exception to the constraints for *starts* and *started\_by* occurs if, given *A starts B* or *B started\_by A*, an event (e.g., *C*) or a set of events occurs *during* event *A*. Here the resulting order is *A before C before B*. Another exception is in the case where event *B* is *during* a third event. Given *A starts B* and *D contains B*, the resulting order is *A before D before B*. Thus *A starts B*  $\square$  *A*  $\prec$  *B* except if  $\square$  *D* | *D contains B* then *A*  $\prec$  *D*  $\prec$  *B*. For *A started\_by B*, *A started\_by B*  $\square$  *B*  $\prec$  *A* and  $\square$  *C* | *B*  $\prec$  *C*  $\prec$  *A*, except if  $\square$  *D* | *D during B* then *B*  $\prec$  *D*  $\prec$  *A*.

An exception is made in the case of *ends* and *ended\_by* if, given *A ends B* or *B ended\_by A*, a set of

events occurs *during* the event that is *ended\_by* the other. In this case these events occur between the two events related by *ends* in the linear order. For example, consider *A ends B* and *C during B*. These map to *B before A* and *B before C*, and the resulting plausible order is *B before C before A*. Event interval *C* falls between *B* and *A* because *C* is related to event *B* by *during*. Thus *A ends B*  $\square$  *B*  $\prec$  *A* and  $\square$  *C*  $\mid$  *B*  $\prec$  *C*  $\prec$  *A*, except if  $\square$  *D*  $\mid$  *D during B* then *B*  $\prec$  *D*  $\prec$  *A*, or if  $\square$  *E*  $\mid$  *A during E* and not *B during E*, then *B*  $\prec$  *E*  $\prec$  *A*. In the case where *A* is *ended\_by B*, *A ended\_by B*  $\square$  *A*  $\prec$  *B* and  $\square$  *C*  $\mid$  *A*  $\prec$  *C*  $\prec$  *B*, except if  $\square$  *D*  $\mid$  *D during A* then *A*  $\prec$  *D*  $\prec$  *B*, or if  $\square$  *E*  $\mid$  *B during E* and not *A during E*, then *A*  $\prec$  *E*  $\prec$  *B*.

### 5.3 Semantics involving overlaps and overlapped\_by

The semantics involving *overlaps* and *overlapped\_by* probably pose the most challenges for automatically generating a linear sequence of events. When one event interval *overlaps* another, the duration of the overlap is not always known. It is possible that two events almost coincide, approximating an *equals* relation. It is also possible that the overlap is very small, such that one event is almost *before* the other, or that the two event intervals almost *meet*. In this work, two event intervals, *A* and *B*, that *overlap* are reduced to *A before B*. No additional constraint is applied because it is assumed that the relation will hold even if there are intermediate events between the events that *overlap*, and therefore, *A overlaps B*  $\square$  *A*  $\prec$  *B*. *Overlapped\_by* is represented as *A overlapped\_by B*  $\square$  *B*  $\prec$  *A*.

### 5.4 Semantics involving before and after

When one event is *before* another event, this order should be preserved in the linear order of events. In contrast to the other relations, it is acceptable to have additional events occurring between any *before*-events, since the relation *before* continues to hold regardless of the number of events between the two events. Therefore, no constraints are necessary and *A before B*  $\square$  *A*  $\prec$  *B*. This same reasoning holds for any event intervals related by *after*, such that *A after B*  $\square$  *B*  $\prec$  *A*.

### 5.5 Semantics involving equals

If two events are *equal* to each other this relation is preserved in the ordering of events. As orders are built, the events that are *equal* will remain together. Events that are not *equal* in the initial set of events and relations will not be *equal* in the final linear orders.

## 6 Using a matrix to generate orders

The matrix  $E_l$  is the basis for generating linear orders. The events are arranged in a linear order using the constraints and mapping rules presented in the previous section. Parsing each row of  $E_l$ , the first event-event combination ( $e_{1,1}$ ) represents either a relation linking the two event intervals or is an empty cell,  $\sim$ . Empty cells obviously do not contribute to any linear order. In this example, there is no valid relation for  $e_{1,1}$ .

Cell  $e_{1,2}$  returns *FerryUnderBridge during CarCrossesBridge*, and based on the mapping rules, an order is instantiated, *CarCrossesBridge before FerryUnderBridge*.

The next cell encountered,  $e_{1,3}$ , contains *FerryUnderBridge before FerryDocks*, and the order is updated to *CarCrossesBridge before FerryUnderBridge before FerryDocks*. Cell  $e_{1,4}$  contains *FerryUnderBridge overlaps PlaneFliesOver*, and because the relation between *PlaneFliesOver* and *FerryDocks* is unknown, two possible orders result:

- (1) *CarCrossesBridge before FerryUnderBridge before PlaneFliesOver before FerryDocks*
- (2) *CarCrossesBridge before FerryUnderBridge before FerryDocks before PlaneFliesOver*

At  $e_{1,5}$ , there is no valid relation and no updates are made to the orders. The next event-event combination considered is  $e_{2,1}$ , i.e., *CarCrossesBridge contains FerryUnderBridge*. This is redundant since *FerryUnderBridge during CarCrossesBridge* has already been considered, and so no changes are necessary to the orders. No valid relations are present in the remainder of the second row, and the next relation encountered is  $e_{3,1}$ , *FerryDocks after FerryUnderBridge*. The inverse of this relation has also already been considered and no changes are made. However,  $e_{2,5}$  contains *FerryDocks meets TruckArrives*, which does result in an update to each of the orders. Adding the event *TruckArrives* such that the constraints of *meets* are satisfied returns two orders:

- (1) *CarCrossesBridge before FerryUnderBridge before PlaneFliesOver before FerryDocks before TruckArrives*
- (2) *CarCrossesBridge before FerryUnderBridge before FerryDocks before TruckArrives before PlaneFliesOver*

The remaining two event interval relations extracted from the matrix (*PlaneFliesOver overlapped\_by FerryUnderBridge* in  $e_{4,1}$  and *TruckArrives met\_by FerryDocks* in  $e_{5,3}$ ) are redun-

dant because their inverses have already been considered, and thus require no additional changes to the orders. When all relations have been processed, the result is a set  $O$  of all possible orders that are plausible and maintain as closely as possible the original semantics of the relations:

$$O = \{ \text{CarCrossesBridge before FerryUnderBridge before PlaneFliesOver before FerryDocks before TruckArrives,} \\ \text{CarCrossesBridge before FerryUnderBridge before FerryDocks before TruckArrives before PlaneFliesOver} \}$$

Using the semantics of the relations, both orders generated are plausible. *PlaneFliesOver* always occurs *after* the ferry is under the bridge, as does *FerryDocks*. *FerryDocks* is always directly before *TruckArrives* with no intermediate events between them. If the result set  $O$  is compared to the set of orders generated in section 4, i.e., before the semantic constraints were applied,  $O$  is shown to consist of a smaller number of orders. These correspond to the most plausible orders given the original set of event-relation combinations present in the text. Orders that do not meet the constraints based on semantics are eliminated and not presented to a user.

## 7 Conclusions and future work

This paper introduces a method for abstracting events from text and generating a linear order of these events based on the temporal semantics associated with them. These simpler orders are useful as summaries of the events contained in the textual description. They also provide a basis for reasoning about the events and performing operations on the orders, such as returning, for example, all events that happened before event  $X$ , or returning the first event, last event, etc. The approach presented in the paper models events as event interval relations and maps the thirteen possible event interval relations to *before* and *equals*. A system of constraints that preserve the semantics associated with the original event relations assists in generating orders that are plausible. For example, when one event is *during* another, no intermediate events will fall between these two events in a linear order. Since for many cases not all relations among events are known, it is possible that there are many candidate linear orders produced for a given narrative. In an effort to reduce the number of orders, methods are presented for filtering the orders based on event relations.

This linear sequencing helps us to understand and communicate how events described in text occur over time and serves as the basis for establishing a *timeline* of events. Timelines organize events into

chronological order and provide a method of encapsulation or summarization where detailed data are rendered in a simpler, temporally ordered fashion. Future work will focus on additional approaches for filtering possible orders, for example, if the narrative includes the locations of events, this spatial information can be used to refine further the orders. Events that happen beyond a certain distance, or outside a known area of interest, for example, can be abstracted from the orders.

## Acknowledgements

This research is supported by the National Geospatial-Intelligence Agency under grant number, NMA201-00-1-200. Kathleen Hornsby's research is further supported by a grant from the National Institute of Environmental Health Sciences, NIH, under grant number 1 R 01 ES09816-01.

## References

- Khaled Al-Taha and Renato Barrera. 1994. Identities through Time. In *Proceedings of the International Workshop on Requirements for Integrated Geographic Information Systems*, pages 1-12, New Orleans, LA.
- Enrique Alfonseca and Suresh Manandhar. 2002. A Framework for Constructing Temporal Models from Texts. In *proceedings of the FLAIRS Conference*, pages 456-460, Pensacola Beach, FL.
- James Allan, Rahul Gupta, and Vikas Khandelwal. 2001. Temporal Summaries of News Topics. In *proceedings of SIGIR'01*, New Orleans, Louisiana, September 9-12, 2001.
- James F. Allen. 1983. Maintaining Knowledge About Temporal Intervals. *Communications of the ACM* 26(11): 832-43.
- Christophe Claramunt and Marius Theriault. 1995. Managing Time in GIS: An Event-Oriented Approach. In *Proceedings of the International Workshop on Temporal Databases*, pages 23-42, Zurich.
- Christophe Claramunt and Marius Theriault. 1996. Toward Semantics for Modelling Spatio-Temporal Processes within GIS. In *Proceedings of the 7th International Symposium on Spatial Data Handling*, pages 47-63, Delft, NL.
- Andrew Frank. 1998. Different Types of "Times" in GIS. In *Spatial and Temporal Reasoning in Geographic Information Systems*, eds. M.J. Egenhofer and R.G. Golledge. Oxford University Press, New York.

- Antony Galton and Juan Carlos Augusto. 2002. Two Approaches to Event Definition. In *Proceedings of the 13th International Conference on Database and Expert Systems Applications (DEXA'02)*, pages 547-556, Aix Provence, France.
- Ralph Grishman, Silja Huttunen, and Roman Yangarber. 2002. Real-Time Event Extraction for Infectious Disease Outbreaks. In *Proceedings of Human Language Technology Conference*, San Diego, CA, March 24-27,2002.
- Annika Hinze and Agnes Voisard. 2002. *A Flexible Parameter-Dependent Algebra for Event Notification Services*. Freie Universitat Berlin, Berlin.
- Kathleen Hornsby and Max J. Egenhofer. 2000. Identity-Based Change: A Foundation for Spatio-Temporal Knowledge Representation. *International Journal of Geographical Information Science*, 14 (3): 207-224.
- Kathleen Hornsby, Max J. Egenhofer, and Patrick Hayes. 1999. Modeling Cyclic Change. *Advances in Conceptual Modeling, ER'99 Workshop on Evolution and Change in Data Management*, pages 98-109, Paris, France.
- Richard Larson. 1999. Time and Event Measure. In *Proceedings of the Linguistic Society of America Meeting*, Los Angeles, CA.
- Inderjeet Mani and Mark T. Maybury, eds. 2001. *Advances in Automatic Text Summarization*. Cambridge, Massachusetts: MIT Press.
- Damir Medak. 1999. Lifestyles - an Algebraic Approach to Change in Identity. In *Proceedings of the International Workshop, Spatio-Temporal Database Management, STDBM'99*, pages 19-38, Edinburgh, Scotland.
- Iakovos Motakis and Carlo Zaniolo. 1995. *A Formal Semantics for Composite Temporal Events in Active Database Rules*. U.C.L.A. Computer Science Department, Los Angeles.
- Alice C. I. Pedersen and Daniel B. Wright. 2002. Do Differences in Event Descriptions Cause Differences in Duration Estimates? *Applied Cognitive Psychology* 16: 769-83.
- James Pustejovsky, José Castaño, Robert Ingria, Roser Saurí, Robert Gaizauskas, Andrea Setzer, and Graham Katz. 2003. TimeML: Robust Specification of Event and Temporal Expressions in Text. In *Proceedings of AAAI Spring Symposium*, pages 28-34, Palo Alto, CA.
- Jeffrey Zacks and Barbara Tversky. 2001. Event Structure in Perception and Conception. *Psychological Bulletin* 127(1): 3-21.