A better tool than Allen's relations for expressing temporal knowledge in interval data

Fabian Morchen
Siemens Corporate Research
755 College Road East
Princeton NJ 08540, USA
fabian.moerchen@siemens.com *

ABSTRACT

Temporal patterns composed of symbolic intervals are commonly formulated with Allen's interval relations originating in temporal reasoning. We show that this representation has severe disadvantages for knowledge discovery. The patterns are not robust, in the sense that small disturbances of interval boundaries lead to different patterns for similar situations. The representation is ambiguous since the same pattern can have quantitatively widely varying appearances. For all but very simple cases the patterns are not understandable because the textual descriptions are lengthy and unstructured. We present the Time Series Knowledge Representation (TSKR), a new hierarchical language for interval patterns to express the temporal concepts of coincidence and partial order. We demonstrate the superiority of this novel form of representing temporal knowledge over Allen's relations for data mining. Results on a real data set support our claims and show a successful application.

Categories and Subject Descriptors

I.5 [**Computing Methodologies**]: Pattern Recognition

General Terms

Models

Keywords

knowledge discovery, time series, interval patterns, Allen's interval relations

1. INTRODUCTION

Symbolic interval time series are an important data format in temporal knowledge discovery [11, 13, 29, 7, 14, 12, 5, 19, 30]. In particular, numerical time series are often converted to symbolic interval time series by segmentation [14, 12], discretization [29] or clustering [11]. Alternatively, the intervals can be obtained directly from other temporal data, e.g., video [9] or association rules over time [21].

Mining such data for patterns has largely been performed based on Allen's interval relations [3], e.g., in [13, 7, 12, 5, 19, 30]. The relations were originally developed in the context of temporal reasoning where inference about past, present, and future supports applications in planning, understanding, and diagnosis. The input usually consists of exact but incomplete input data and temporal constraints, often expressed by Allen's relations. Typical problems include determining the consistency of the data and answering queries about scenarios satisfying all constraints. But these problems do not occur in the data mining context: almost the complete interval data is given and meaningful and understandable patterns are searched [12]. One may have to cope with some missing data, but more importantly with possibly noisy and incorrect data.

We show that Allen's relations have severe disadvantages when used for pattern discovery from interval time series. We introduce an alternative representation for temporal knowledge in interval data and compare it to approaches using Allen's relations conceptually and experimentally. We show that our representation is more expressive, more robust, and arguably more understandable than the patterns of Höppner [12] using Allen's relations.

2. MOTIVATION

The 13 interval relations of Allen are [3]: *before*, *meets*, *overlaps*, *starts*, *during*, *finishes*, the corresponding inverses and *equals*. They can describe any relative positioning of two intervals. The relations are commonly used for the formulation of temporal rules involving intervals [13, 7, 12, 22, 5, 19, 30].

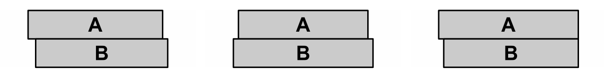
Three methods use Allen's relations for unsupervised rule mining from symbolic interval data with variants of the Apriori algorithm. In [13] interval patterns are restricted to right concatenations of intervals to existing patterns, using one of Allen's relations to describe the relative positioning of the new interval to the interval of the complete pattern. Similarly, in [7] the patterns are restricted to composites of two already significant patterns or single intervals. Both approaches thus use at most $k-1$ relations for k intervals. In contrast, the patterns of [12] list all $\frac{k(k-1)}{2}$ pairwise relations of the intervals within a pattern. In [19] the problem of mining patterns from interval data was rediscovered. The pattern format of [12] is used with a subset of Allen's relations. The patterns are mined with a tree-based enumer-

TDM'06, August 20, 2006, Philadelphia, Pennsylvania, USA.

ation algorithm [4]. In [30] the patterns of [12] are mined with a modified sequential pattern algorithm [15].

We think that using Allen's relation to represent interval patterns has severe disadvantages for knowledge discovery and first formulate and justify our claims with examples.

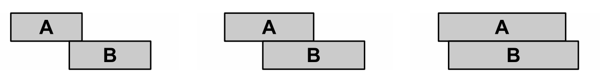
(1) Patterns from noisy interval data expressed with Allen's interval relations are not robust: Several of Allen's relations require the equality of two or more interval endpoints. Small disturbances can create patterns where a very similar relationship between intervals is fragmented into different relations as defined by Allen's relations. Figure 1 shows several examples of almost equal intervals.



(a) overlaps (b) during (c) finishes

Figure 1: Examples for different patterns according to Allen that are fragments of the same approximate relation *almost equals*.

(2) Patterns expressed with Allen's interval relations are ambiguous: The same relation of Allen can visually and intuitively represent very different situations. In Figure 2 three very different versions of the *overlaps* relation are shown as an example. Even more ambiguous is the compact representation of patterns from [13] or [7], several different descriptions are valid for the exact same pattern (see Figure 3).



(a) small (b) medium (c) large

Figure 2: Three instances of Allen's *overlaps* relation with large quantitative differences.

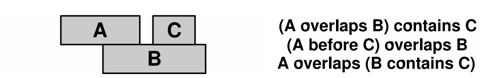


Figure 3: Pattern that can be described by three different compact rules using Allen's relations.

(3) Patterns expressed with Allen's interval relations are not easily comprehensible: The representation of patterns with Allen's relations does not follow the Gricean maxims [10] suggested for the representation of knowledge discovery results to humans [25]. For example, the maxim of manner is violated. Since the compact format of [13, 7] is ambiguous, patterns need to be expressed with an unstructured list of pairwise relations of all intervals [12] that grows quickly with the number of intervals.

We present the Time Series Knowledge Representation (TSKR), a new hierarchical language for the representation of temporal knowledge based on interval times series. The TSKR extends the Unification-based Temporal Grammar (UTG) that was proposed by Ultsch in [26, 27] and applied to the analysis of sleeping disorders in [11]. The central pattern elements of the UTG are Events and Sequences shown in Figure 4. Events combine several *more or less simultaneous* intervals, a robust version of Allen's *equals*. The number of intervals in an Event is restricted, however, to the dimensionality of the interval series. Sequences describe an ordering of several Events with *immediately followed by* or *followed by after at most t time units*.

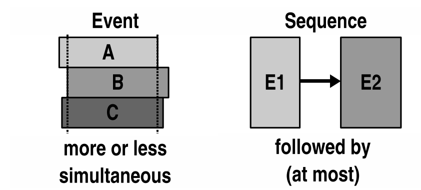


Figure 4: Core patterns of the UTG: Events describe a group of almost simultaneously starting and ending intervals, Sequences describe an order of Events.

The TSKR extends the UTG significantly by allowing an *arbitrary number* of coinciding *parts of intervals* in constrast to the *fixed* number of *complete intervals* within Events and by relaxing the *total order* in Sequences to a *partial order*. For more details on the relation of the TSKR to the UTG we refer the interested reader to [18]. Other interval mining methods are not as powerful as Allen's relations or the UTG. In [29] only containments of intervals and in [14] only associations of successive intervals are searched.

3. TIME SERIES KNOWLEDGE REPRESENTATION

The TSKR is a hierarchical interval language describing the temporal concepts of duration, coincidence, and partial order in interval time series. The basic primitives are labeled intervals called *Tones* representing duration. A Tone pattern describes a property of the temporal process that is (repeatedly) observed during time intervals. For example, a Tone labeled *temperature increasing* could be obtained by segmenting a numerical time series and considering the slope on each segment. In the following paragraphs we describe the higher level TSKR patterns composed of Tones. Simultaneously occuring Tones form a *Chord*, representing coincidence. Several Chords connected with a partial order form a *Phrase*. We only briefly outline the mining algorithms, for more details see [17, 18].

In data mining the input data is usually measured at discrete time points of a certain resolution representing a sample of the generating time continuous process. Without loss of generality, we define the following patterns based on the natural numbering $\mathbb{T}$ of a set of uniformly spaced time points. Let Σ be a set of unique *symbols*.

Definition 1. *A* symbolic time interval *is a triple* $[\sigma, s, e]$ *with* $\sigma \in \Sigma$, $[s, e] \in \mathbb{T}^2$, $s \leq e$. *The duration of a symbolic interval is* $d([s, e]) = e - s + 1$. *We write* $[\sigma, s, e] \subseteq [\sigma', s', e']$ *if* $s' \leq s$ *and* $e \leq e'$ *with equality iff* $s = s'$ *and* $e = e'$. *A symbolic time interval* $[\sigma, s, e]$ *is* maximal *if* $\forall [\sigma', s', e'] \supset [\sigma, s, e]$ $\sigma' \neq \sigma$. *If* $\{s, ..., e\} \cap \{s', ..., e'\} \neq \emptyset$ *we say that the intervals* $[\sigma, s, e]$ *and* $[\sigma', s', e']$ *overlap.*

Definition 2. *We define a* partial order $\prec$ *of intervals as* $[s_1, e_1] \prec [s_2, e_2] \Leftrightarrow e_1 < s_2$. *We say that* $[s_1, e_1]$ *is* before $[s_2, e_2]$.

Definition 3. *A* symbolic interval series *is a set of non-overlapping symbolic time intervals* $\mathcal{I} = \{[\sigma_i, s_i, e_i] | \sigma_i \in \Sigma, [s_i, e_i] \in \mathbb{T}^2, i = 1, ..., N; e_j < s_{j+1}, j = 1, ..., N-1\}$. *The duration of an interval series is* $d(\mathcal{I}) = \Sigma_{i=1}^{N} d([s_i, e_i])$.

Definition 4. *A* symbolic interval sequence *is a set of symbolic time intervals* $\mathcal{I} = \{[\sigma_i, s_i, e_i] | \sigma_i \in \Sigma, [s_i, e_i] \in \mathbb{I}, i = 1, ..., N\}$

At the core of our knowledge representation stand the patterns and the occurrences of patterns in the input data.

Definition 5. *Let* Λ *be a finite set of* labels λ. *Let* $l : \Sigma \mapsto \Lambda$ *be the function assigning each symbol a label. Let* Φ *be a set of characteristic functions* $\phi_{\mathcal{X}} : \mathbb{I} \rightarrow \{\text{TRUE}, \text{FALSE}\}$, *where* $\mathcal{X}$ *is some arbitrary input data. A* pattern $(\sigma, \lambda, \phi_{\mathcal{X}})$ *is a semiotic triple [26] composed of:*

- $\sigma \in \Sigma$, *a unique symbol representing the pattern in higher levels constructs (*syntax*),*
- $\lambda = l(\sigma) \in \Lambda$, *a label providing a textual description of the practical meaning of the pattern (*pragmatic*),*
- $\phi_{\mathcal{X}} \in \Phi$, *a characteristic function determining when the pattern occurs (*semantic*).*

The semantic of ϕ *and the data type of* $\mathcal{X}$ *will differ for the specific patterns defined below. We write* ϕ *for* $\phi = \text{TRUE}$ *and* $\neg\phi$ *for* $\phi = \text{FALSE}$.

The concept of semiotic triples is consistently used on all levels of the TSKR pattern hierarchy. It enables the pragmatic annotation of each pattern with labels to aid the later interpretation when used as parts of larger patterns. A value-based Tone obtained by discretizing a univariate time series $v_t \in \mathbb{R}\ t \in \mathbb{T}$ has a characteristic function of the form $\phi_A([s,e]) \leftarrow v_{min} < v_i \leq v_{max} \forall i \in = \{s, ..., e\} \subset \mathbb{T}$ where $v_{min}, v_{max} \in \mathbb{R}$ and could be labeled *high* or *medium*.

Definition 6. *An* occurrence *of a pattern is an interval* $[s,e] \in \mathbb{I}$ *with* $\phi_{\mathcal{X}}([s,e])$. *A* maximal occurrence *is an occurrence* $[s,e]$ *such that* $\forall [s',e'] \supset [s,e]\ \neg\phi_{\mathcal{X}}([s',e'])$.

In this study we assume a set of Tones given. The observations according to the characteristic function of a single Tone form a symbolic interval series, the observations of several Tones form a symbolic interval sequence.

Definition 7. *A* Chord *pattern is a semiotic triple* $c = (\sigma, \lambda, \phi_{\mathcal{T}}^{T})$ *with* $\sigma \in \Sigma$, $\lambda \in \Lambda$, *and a characteristic function* $\phi_{\mathcal{T}}^{T} \in \Phi$ *indicating the simultaneous occurrences of k Tones* $T = \{t_i = (\sigma_i, \lambda_i, \phi_i) | i = 1, ..., k, k > 0\}$ *on a given time interval according to the interval sequence* $\mathcal{T}$ *with occurrences of the Tones* T:

$$\phi_{\mathcal{T}}^{T}([s,e]) \leftarrow \phi_1([s,e]) \wedge ... \wedge \phi_k([s,e]) \quad (1)$$

We say the Chord $c_i \supset c_j$ *is a super-Chord of* c_j *if* c_i *describes the coincidence of a superset of the Tones from* c_j.

Definition 8. *Let the* support set *of a Chord be the symbolic interval series of all maximal intervals. The* support $sup(c)$ *of a Chord c is the duration of the support set.*

Often, the support set will be restricted to intervals with a duration of at least $\delta > 0$, to exclude very short temporal phenomena that are not meaningful for the application under study. In Figure 5 the occurrences of the Tones A, B, and C are shown in the top segment. In the middle segment all maximal Chords of size 2 and 3 are shown. All support sets consist of two intervals. The occurrence of a Chord pattern implies the occurrence of all sub-Chords on the same interval, e.g. AB, BC, and AC for ABC. In general larger Chords can be considered more interesting, because they are more specific. We use the concept of closedness to non-redundantly represent a set of Chords motivated by closed itemsets [20].

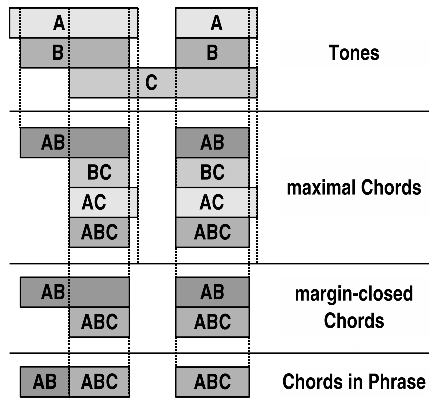


Figure 5: Simultaneous occuring Tones form maximal Chords. Phrases describe a (partial) order of (not neccessarily maximal) Chords.

Definition 9. *A Chord* c_i *is* closed *if there are no super-Chords that have the same support, i.e.,* $\forall c_j \supset c_i, sup(c_j) < sup(c_i)$.

The definition of closedness considers a Chord as closed even if a larger Chord has only a slightly smaller support set. With Tone patterns mined from possibly inexact and erroneous time series this is a harsh restriction. We therefore introduce the relaxed concept of margin-closedness to prune patterns with very similar support.

Definition 10. *A Chord* c_i *is* margin-closed *w.r.t. a threshold* $\alpha < 1$ *if there are no super-Chords that have almost the same support, i.e.,* $\forall c_j \supset c_i, \frac{sup(c_j)}{sup(c_i)} < 1 - \alpha$.

The third segment in Figure 5 shows all margin-closed Chords ($\alpha = 0.1$). The Chord BC is not closed, because whenever it is observed, so is the super-Chord ABC. The Chord AC is closed, but not margin-closed, because the support set of the super-Chord ABC is only slightly smaller.

Chords are similar in structure to the well known itemsets [1]. Each Tone symbol is an item, while a Chord is a subset of all items. The set of all margin-closed Chords can be mined with a modified version of the CHARM [32] algorithm for mining closed itemsets [17].

Definition 11. *A* Phrase *pattern is a semiotic triple* $p = (\sigma, \lambda, \phi_{\mathcal{C}}^{C,E})$ *with* $\sigma \in \Sigma$, $\lambda \in \Lambda$, *and a characteristic function* $\phi_{\mathcal{C}}^{C,E} \in \Phi$ *indicating the occurrences of the k Chords* $C = \{c_i = (\sigma_i, \lambda_i, \phi_i) | i = 1, ..., k, k > 0\}$ *according to a partial* *order $E \subseteq \{\sigma_i\}^2$ on a given time interval according to the interval sequence $\mathcal{C}$ with occurrences of the Tones C:*

$$\begin{aligned}
\phi_C([s,e]) \quad \leftarrow \quad & (\forall i=1,...,k\ \exists [s_i,e_i] \subseteq [s,e]\ \phi_i([s,e])) && (2)\\
& \wedge(\exists i \in \{1,...,k\}\ s_i = s) && (3)\\
& \wedge(\exists i \in \{1,...,k\}\ e_i = e) && (4)\\
& \wedge(\forall i \neq j \in \{1,...,k\}^2 && (5)\\
& [\sigma_i, s_i, e_i] \prec [\sigma_j, s_j, e_j] && (6)\\
& \Leftrightarrow (\sigma_i, \sigma_j) \in E) && (7)
\end{aligned}$$

We say the Phrase $p_i \supset p_j$ is a super-Phrase of p_j if p_i describes the partial order of a superset of of the Chords of p_j and all common Chords have the same partial order.

The characteristic function for a Phrase consists of four necessary conditions. Line 2 ensures that the intervals of all Chords are within the Phrase interval, while Line 3 and Line 4 prevent extra room before the first and after the last Chord, respectively. The occurrence of a Phrase is thus maximal by definition. A Phrase occurs on a particular interval but not on the sub-intervals. Lines 5-7 require the Chords to be in the partial order specified by E. Note, that any two intervals that have an order relation in E are not allowed to overlap. This restriction makes sense, because Chords already describe the concept of coincidence. Allowing overlapping Chords within a Phrase in general would mean to repeatedly represent the same concept and can in fact be equally represented by a larger Chord on the interval where two Chords overlap. The bottom segment of Figure 5 shows Chord intervals that are part of a Phrase. The interval for AB is non-maximal to avoid overlap with the interval of ABC.

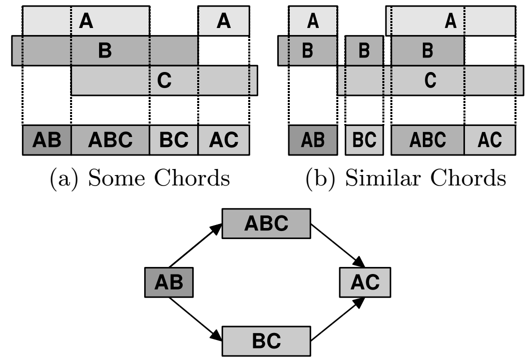
The power of partial order is shown in Figure 6. The two similar Chord sequences in the bottom rows of Figure 6(a) and Figure 6(b) are summarized by the partial order graph of a Phrase shown in Figure 6(c). Note, that similar to [12] multiple observation intervals of a Tone (here A) are allowed and treated as distinct intervals in the Phrase. In contrast to [12] and similar to similar to [9] we further allow that only a part of an observed interval is used in a pattern. Different parts of the long Tone interval B in Figure 6(a) are used in the Chords AB, ABC, and BC. The Phrase in Figure 6(c) thus summarizes cases where the Tone B in the three Chords AB, BC, and ABC stems from one (Figure 6(a)) or more (Figure 6(b)) occurrences of the Tone B.

Definition 12. *Let the* support set *of a Phrase be the symbolic interval series of all maximal intervals. The* support $sup(p)$ *of a Phrase c is the size of the support set.*

Definition 13. *A Phrase p_i is* closed *if there are no super-Phrases that have the same support, i.e., $\forall p_j \supset p_i, sup(p_i) = sup(p_j)$.*

Definition 14. *A Phrase p_i is* margin-closed *w.r.t. a threshold $\alpha < 1$ if there are no super-Phrases that have almost the same support, i.e., $\forall p_j \supset p_i, \frac{sup(p_j)}{sup(p_i)} < 1 - \alpha$.*

Just as Chords relate to itemsets, Phrases relate to episodes [16]. The mining of margin-closed Phrases can be performed in several steps [17, 18]: First, the interval sequence of Chords is converted to an itemset sequences with one itemset per interval where no Chords change containing



(a) Some Chords (b) Similar Chords

(c) Phrase for both

Figure 6: Chords summarize several overlapping Tones. Phrases summarize similar sequences of Chords.

all currently active Chords. Next, an algorithm for closed sequence mining, e.g. CloSpan [31], is applied using a windowing of the itemset sequence. Similar to [6] the closed sequences are grouped according to their transaction lists. Each group is then converted to a partial order. We perform the grouping according to margin-closedness with a modified version of the CHARM [32] algorithm considering closed sequences as items and the groups as itemsets.

4. ANALYSIS

Having introduced a new language for the description of interval patterns, we compare the TSKR to the patterns of [12, 30] using Allen's interval relations. We show that the pattern language of the TSKR has a higher temporal expressivity, is more robust, and has advantages in interpretability.

4.1 Expressivity

In data mining the purpose of a pattern expressed in a knowledge representation language is to collectively describe many similar or possibly equal situations observed in the data. We say a pattern is more *expressive* than another, if it summarizes more similar, yet qualitatively different situations.

We first show that all patterns expressible with pairwise Allen's interval relations can also be expressed with the TSKR by construction. That means that all occurrences of a *single* interval pattern described with the pattern format of [12] can also be described by a *single* TSKR pattern. We write AB for a Chord with coinciding Tones A and B. We write $AB \rightarrow CD$ for the Phrase expressing the total order of the Chord AB followed by the Chord CD.

Let $\mathcal{I} = \{(\sigma_i, s_i, e_i) | i = 1,...,k\}$ be all involved symbolic intervals with arbitrary pairwise relations according to Allen. Let $B = \{b_j\} = \bigcup_{i=1}^{k}\{s_i, e_i\}$ the sorted set of all interval boundaries. We construct a Phrase with at most $|B| - 1$ Chords where the j-th Chord describes the non-empty coincidence of all σ_i where $s_i \leq b_j$ and $e_i \leq b_{j+1}$ or is skipped otherwise. The Chords have a complete ordering according to the indices j.

Consider the example pattern in Figure 7 consisting of six intervals and at least one representative of each of Allen's operators within the pairwise relations. The six resulting Chords are shown in the bottom row.

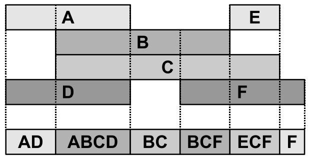
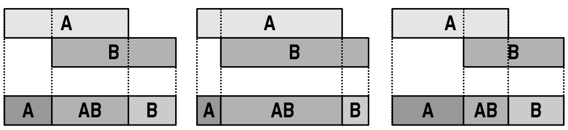
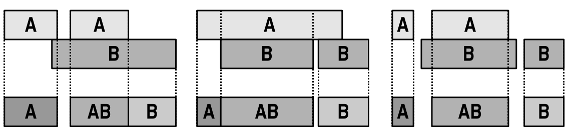


Figure 7: Conversion of complex pattern using all of Allen's relations to a TSKR pattern.

Note, that a TSKR patterns constructed in this way describes even more similar situations in the data than the original pattern using Allen's relations. In Figure 8(a) several instances of Allen's relation A *overlaps* B are shown with the corresponding TSKR pattern $A \rightarrow AB \rightarrow B$. In Figure 8(b) we give some examples of instances that are also covered by the TSKR pattern but not by the Allen pattern. The Phrase also matches observations where the long interval of A and B in Figure 8(a) are interrupted by noise.



(a) Examples of A *overlaps* B and $A \rightarrow AB \rightarrow B$.



(b) Examples also described by $A \rightarrow AB \rightarrow B$.

Figure 8: The TSKR pattern $A \rightarrow AB \rightarrow B$ can describe all instances of Allen's *A overlaps B* pattern and more.

On the other hand it is not always possible to find a *single* pattern defined with Allen's relations to describe all situations covered by a certain *single* TSKR pattern. In particular Phrases that summarize similar situations as in Figure 6 by utilizing the concept of partial order, cannot be expressed by a single pattern using *all* pairwise relations of Allen. Consider the Chords AB, ABC, BC, and AC as shown in Figure 6(a) at the bottom resulting from the Tone patterns A, B, C in the top rows. In Figure 6(b) the same Tone intervals result in the same Chords but with the middle two Chords exchanged. Both versions can be captured in a *single* Phrase (see Figure 6(c)) using the concept of partial order. The order relation of ABC and BC is simply not specified, both versions of the pattern match this description. The two similar instances cannot be captured with a single pattern using the pattern format of [12]. Further, the instances of the TSKR pattern $A \rightarrow AB \rightarrow B$ in Figure 8(a) and Figure 8(b) cannot be described with a single pattern of [12] because they have a different number of intervals.

Even if we restrict ourselves for the moment to TSKR patterns using only a total order among the Chords in a Phrase and exactly the same set of participating Tone intervals, constructing a pattern with pairwise Allen's relations that describes the same situations in the data is not always possible. For example the simple Phrase $AB \rightarrow C$, i.e., Tones A and B occur simultaneously followed by an interval where the Tones C occurs, covers instances that correspond to very different relations of Allen. The relation between A and B could be *overlaps*, *starts*, *during*, *finished*, *equals* plus the corresponding inverses. The relation between A (or B) and C could be *before*, *meets*, or *overlaps*. See Figure 9 for examples.

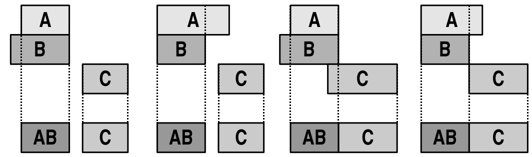


Figure 9: Example of the same simple TSKR Phrase $AB \rightarrow C$ with different relations according to Allen.

In summary, a single TSKR pattern can express many observations in the data where time intervals have a very similar relative positioning but different relations according to Allen. In contrast each pattern according to Allen can also be described with a single TSKR pattern. The TSKR achieves a higher level of abstraction over small disturbances in the data while offering clear temporal semantics with the concepts of coincidence and partial order.

4.2 Robustness

Symbolic interval series or sequences obtained from numeric time series inherit the noise present in the original data. The interval boundaries gained from preprocessing steps like discretization of values or segmentation are subject to noise in the measurements. Such time points should thus be considered approximate.

Using Allen's relations to formulate patterns such slight variations of interval boundaries can create fragmented patterns that describe the same intuitive relationship between intervals with different operators. An example for two almost equal intervals was given in Figure 1. Many more examples can be constructed. Any pattern using one of Allen's relations that requires equality of two interval boundaries can be destroyed by changing one boundary by one time unit only.

There are approaches to relax the strictness of Allen's relations by using a threshold to consider temporally close interval boundaries equal [2]. This has not been used in temporal pattern mining. Fragmented patterns are still possible if noise causes interval boundaries to be shifted around the threshold value.

In contrast, the Chord and Phrase patterns of the TSKR are designed to be insensitive to small changes of the interval boundaries. The TSKR operator *coincides* is extremely robust. It only considers the intersection of all participating intervals, any interval can individually be stretched to infinity without changing the pattern at all. All three cases in Figure 1 could be represented well with a Chord describing the coincidence of A and B. The leading and trailing intervals where only A or B is observed would not be considered of significant duration for appropriate choices of the minimum Chord duration.

When making intervals smaller, the pattern breaks down as soon as the smallest interval disappears - as is true for both temporal knowledge representations. The robustness of Phrases directly depend on the Chords. Other aspects of noisy data like missing intervals similarly affect both representations. This can be compensated by allowing alternatives within a pattern [26, 12].

4.3 Interpretability

Interpretability is hard to specify or measure. In [25] natural language descriptions of time series are generated and interpreted by experts. The authors suggest that the presentation should follow the Gricean maxims of quality, relevance, manner, and quantity. We describe the implications of some of the the maxims in the context of interval patterns and judge how well the different pattern languages support this paradigm.

The maxim of relevance requires, that only patterns relevant to the expert are listed. This is a rather application dependent requirement, still there are differences in how the pattern languages support this task. The hierarchical structure of the TSKR enables the user to view and filter lower level patterns before the next level constructs are searched. Mining Phrases from only the relevant Chords will be much faster, fewer and smaller Phrase patterns need to be analyzed in the next step. This form of pattern representation supports the human analysis according to the zoom, filter, and details on demand paradigm [23]. The patterns expressed with Allen's relations are commonly mined in a single step, resulting in a much larger set of patterns for manual analysis.

The maxim of manner suggests to be brief and orderly and to avoid obscurity and ambiguity. The TSKR patterns can be longer or shorter than the Allen patterns of [12]. A pattern of [12] with k intervals always consists of $\frac{k(k-1)}{2}$ pairwise relations. When counting each Chord and each consecutive order relation in a Phrase separately the worst case for the number of atomic relations in a TSKR pattern is $(2k-1)(2k-2)$ because with $2k$ interval boundaries there can be at most $2k-1$ Chords. These Chords would then occur consecutively involving $2k-2$ binary order relations. In the best case there is only a single relation if all intervals are equal and thus form a single Chord. The typical size of a TSKR pattern given k intervals depends on the data set at hand (see Section 5).

The textual representation of the TSKR can be argued to be more orderly and to avoid obscurity because is uses a hierarchical structure with a partial order relation on the highest level that offers details on demand. In contrast, there is no inherent order in the list of pairwise Allen relations. They could be ordered by the intervals of the first argument of each relation according to an temporal order of the intervals or the symbolic labels. In either case, the list needs to be considered as a whole to understand the pattern.

To avoid ambiguity, a given pattern should semantically describe a single intuitive notion of the relationship of several intervals. This is not the case for all of Allen's relation. The instances of a single pattern can quantitatively vary significantly and represent intuitively different patterns. This was already demonstrated for the *overlaps* relation in Figure 2, similar examples can be constructed for other relations. Whether different versions of a pattern are semantically equivalent depends on the application. The separation of temporal concepts in the TSKR, however, completely avoids this problem. Chord patterns simply describe the overlapping parts and ignore the rest of the intervals. Phrase patterns express the concept of partial order, allowing variation only in the length of Chords and possible gaps. The ambiguity problem is even worse for the compact pattern format used in [13] and [7]. The *exact same* instance of a pattern with k intervals can be written in $\frac{k!}{2}$ different ways by consecutively choosing an interval for the next position in the pattern and excluding duplicates caused by inverse operators. Again, this is not the case for the TSKR, given a set intervals and a minimum duration for Chords there is only one Phrase of all maximal Chords.

5. EXPERIMENTS

We compare the TSKR with Allen's relations on symbolic interval data describing videos[1]. Different scenarios involving a hand moving colored blocks were analyzed with the LEONARD system for recognition of visual events from video camera input [24]. The scenes include simple actions like putting one block on another and more complex scenes where a whole stack of blocks is built. In [9] each scene is described with a logical formula found by inductive logic programming in a supervised process. We use the data in an unsupervised manner and use the ground truth on the scenarios for evaluation purposes only. We preprocessed the descriptions by normalizing the argument order, filtering out some redundant descriptions, and merging scenarios that were equivalent.

We mined Chords with a minimum support of 1%, minimum size of 1, and a minimum count of 10. This leads to 19 closed Chords and 15 margin-closed Chords using $\alpha = 0.1$. We further dropped three Chords of size one not involving the hand, but rather one of the *contacts* Tones. These Tones are only interesting when combined with a hand action. The lattice of the remaining Chords is shown in Figure 10.

The trivial Chord C9 describes the hand holding the red block. This Chord has very high support, covering 50% of all time points and has several super-Chords. C13 and C14 describe the hand holding the red block, while it sits on top of the green or blue block, respectively. Both have further super-Chords C11 and C12 where yet another block is part of the stack. Note, that the support of the super-Chords is always at least 10% smaller than that of any immediate sub-Chords, according to $\alpha = 0.1$. The Chords C2 and C3 in the upper left represent a stack of three blocks not touched by the hand. Without looking at the original Video data, it is unclear at this point which block sits on top and which at the bottom. For C2 we only know that blue is in the middle, because it is an argument of both *contacts* relations. The co-occurrences of the Chords with the known scenarios are striking, a selection is listed in Table 1. The Chord patterns are not sufficient, however, to discriminate the scenarios, because they always appear in at least two different scenes that are reversed versions of one another, e.g. *stack* and *unstack*. In order to distinguish each pair, we need patterns expressing an order among the Chords.

Using a minimum frequency of 12, 20 closed sequential patterns were found and grouped into 10 margin-closed Phrases ($\alpha = 0.1$). The co-occurrences of the Phrases with the known scenarios showed an almost perfect correspondence, see Table 2. The Phrases further *explain* the actions in the videos. The Phrase for the *disassemble* scenario is shown with pictures from the original Videos in Figure 11.

For mining patterns expressed with Allen's relations we used the format of [12] to avoid ambiguity and counted sup-

[1] `ftp://ftp.ecn.purdue.edu/qobi/ama.tar.Z`

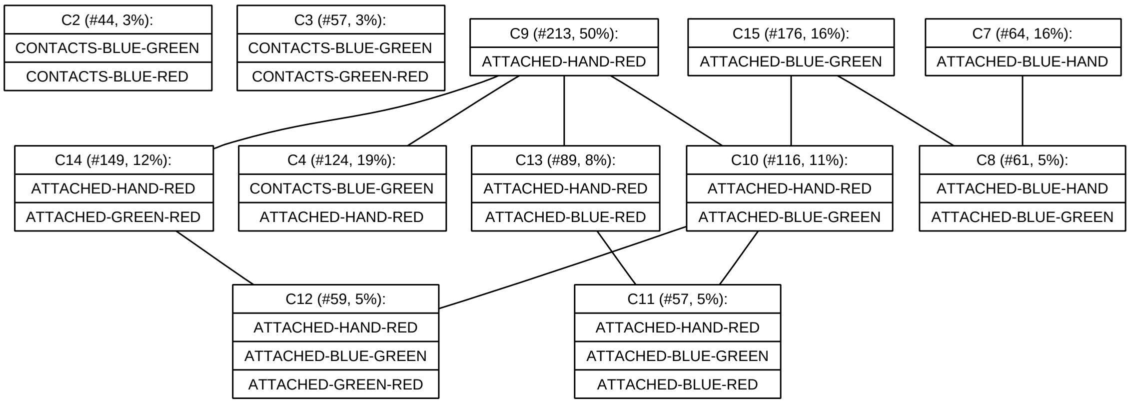


Figure 10: The lattice of the most interesting margin-closed Chords from the Video data annotated with frequency (#) and support (%). The three Chords in the upper right represent the hand holding one of the three blocks. The two most specific Chords with three Tones describe a stack of three blocks with the hand holding the topmost block.

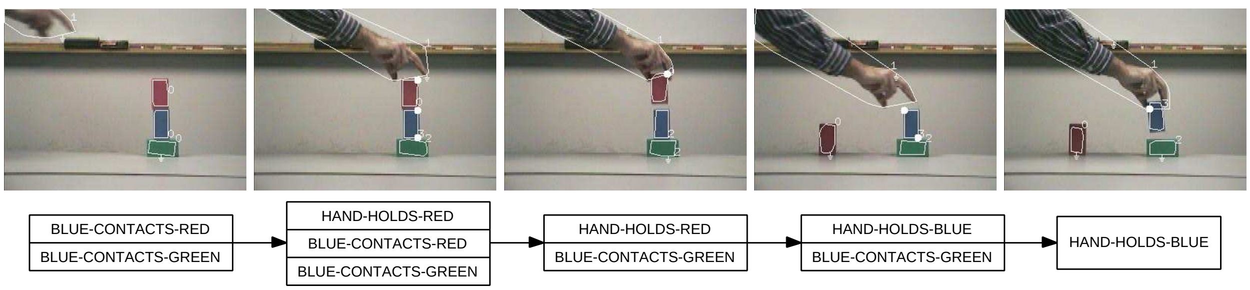


Figure 11: Video frames [9] and Phrase explaining the complete *disassemble* scene.

Scenes/Chords	BLUE-CONTACTS-GREEN, BLUE-CONTACTS-RED	BLUE-CONTACTS-GREEN, BLUE-CONTACTS-RED	BLUE-CONTACTS-GREEN, HAND-HOLDS-RED
stack	0	28	30
unstack	0	29	30
assemble	14	0	34
disassemble	30	0	30

Table 1: Co-occurrences of Chords and Video scenes.

port as occurrences in windows [13] to be comparable with the Phrase mining. We further pruned the set of patterns applying the concept of margin-closedness in a brute force post-processing step. Using minimum size of two and a minimum frequency of 10, 363 patterns were found, 174 of which were margin-closed for $\alpha = 0.1$. This large amount of patterns could not be analyzed manually, some filtering needed to be applied.

We first looked at the 14 largest patterns with five or more intervals. Patterns of these sizes were *only* observed within the *(dis)assemble* scenes with at most 21 out of 30 repetitions. The patterns for the *assemble* scene further described only fragments of the true temporal phenomena. None of them contained any information about the hand taking the red block and placing it on top of the stack. We alternatively filtered the mining results by a minimum frequency of 24 and a minimum size of three. These patterns were all composed of only three symbolic intervals, not sufficiently explaining the complex events of these scenes. As an example we show a pattern found in 26 out of 30 of the *disassemble* scenes in Figure 12. The intervals only correspond to the first three video frames of Figure 11 and fail to mention important parts of the scene. The TSKR Phrases consist of up to 10 (sub-)intervals providing much more information.

Figure 12: Example of frequent Allen pattern A1 explaining only fragments of the *disassemble* scene in Figure 11.

Apart from the more specific explanations that the TSKR patterns provide, they are also almost always more precise and more distinctive than the Allen patterns of [12], as shown in Table 3. For each video scene, precision and recall of the best pattern from each representation was calculated. Only the recall for the complex *assemble* scene is better for Allen/Höppner. In this case the TSKR pattern

Scenes/Phrases	(CONTACTS-BLUE-GREEN, ATTACHED-HAND-RED) → (ATTACHED-HAND-RED, ATTACHED-BLUE-GREEN, ATTACHED-GREEN-RED) → (CONTACTS-BLUE-GREEN, CONTACTS-GREEN-RED)	(CONTACTS-BLUE-GREEN, CONTACTS-GREEN-RED) → (ATTACHED-HAND-RED, ATTACHED-BLUE-GREEN, ATTACHED-GREEN-RED) → (CONTACTS-BLUE-GREEN, ATTACHED-HAND-RED)	(ATTACHED-BLUE-HAND) → (ATTACHED-BLUE-HAND, ATTACHED-BLUE-GREEN) → (CONTACTS-BLUE-GREEN, ATTACHED-HAND-RED) → (ATTACHED-HAND-RED, ATTACHED-BLUE-GREEN, ATTACHED-BLUE-RED)	(BLUE-CONTACTS-RED, BLUE-CONTACTS-GREEN) → (HAND-HOLDS-RED, BLUE-CONTACTS-RED, BLUE-CONTACTS-GREEN) → (HAND-HOLDS-RED, BLUE-CONTACTS-GREEN) → (HAND-HOLDS-BLUE, BLUE-CONTACTS-GREEN) → (HAND-HOLDS-BLUE)
stack	28	0	0	0
unstack	0	28	0	0
assemble	0	0	28	0
disassemble	0	0	0	30

Table 2: Co-occurrences of Phrases and Video scenes.

is much more complex and provides better insight into the temporal phenomena of the scene sacrificing some discrimination power for explanation.

A detailed analysis of the pattern for *put-down* revealed five visually very similar patterns with the same intervals but slightly different pairwise relations (see Figure 13). Only the first combination is frequent enough to appear in the mining result, the others would commonly be filtered out from the large amount of patterns found because they are small and rare.

We further evaluated the typical size of TSKR patterns on this data set. We ordered the intervals by their start points and end points. Each interval was used to generate a pattern of size k by selecting the next $k-1$ intervals as long as consecutive intervals shared at least one time point. Such a pattern can be described by $\frac{k(k-1)}{2}$ of Allen's relations. For the creation of TSKR patterns we considered all intervals between any two consecutive time points taken from the de-duplicated set of the $2k$ start and end points

Scenes	TSKR		Allen/Höppner	
	Precision	Recall	Precision	Recall
pick-up	100.0	100.0	42.3	36.7
put-down	97.8	62.0	41.9	60.0
stack	100.0	93.3	80.0	66.7
unstack	100.0	93.3	100.0	90.0
assemble	100.0	93.3	100.0	100.0
disassemble	100.0	100.0	76.5	86.7

Table 3: Precision and recall for Allen and TSKR patterns.

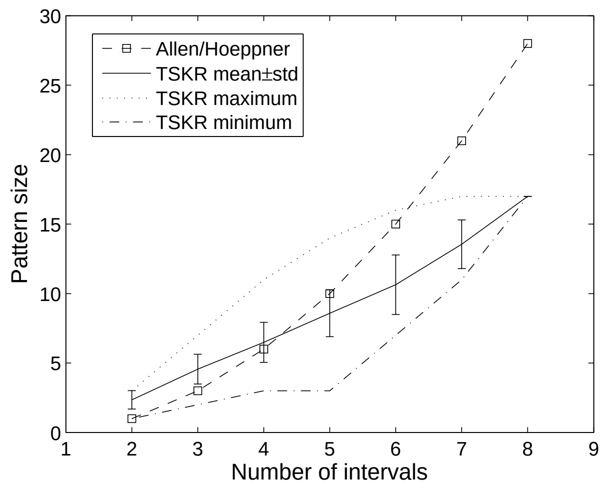


Figure 14: Fixed size of Allen patterns compared to typical size of TSKR patterns given k intervals of the video data.

of the k intervals. Assuming the strictest minimum Chord duration of 1, we created a Chord for each of these intervals describing the coincidence of all original intervals that intersect it. Finally, a Phrase describing the total order of these Chords was considered. Let the size of the de-duplicated set of time points be l. Counting each Chord and each consecutive order within a Phrase as an elementary relations, the size of the TSKR pattern is $(l-1)+(l-2)$. In Figure 14 the mean and standard deviation, as well as the minimum and maximum of the TSKR pattern sizes are compared to the fixed size of Allen patterns for several values of k.

For very small k Allen's relations can usually more compactly describe a group of intervals. As an explanation consider the example *A overlaps B* from Figure 8(a). While there is only one relation of Allen for the two intervals, the TSKR considers 3 Chords with 2 order relations (size 5). For larger patterns the opposite effect is observed on this data set. For $k = 4$ the typical size of TSKR patterns is about the same than that of the Allen patterns and for larger patterns it is much smaller. Even the maximum TSKR pattern size observed on this data set is well below the size of Allen patterns for $k \geq 7$. The more intervals are involved, the more the pattern size seems to profit from Chords that group more than 2 concurrently observed intervals. For $k = 8$ only few TSKR patterns, all of the same size, could be constructed with the process described above, for $k \geq 9$ no such patterns were available.

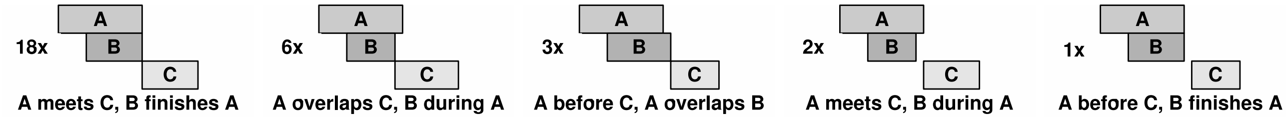


Figure 13: Five similar patterns for *put-down* only one of which is frequent. A is the hand holding the red block, B is the placing of the red on top of the green block, C is the red on the green block without the hand.

6. DISCUSSION

The descriptions found by the TSKM in the Video data in an unsupervised process directly explained the known experimental setups. Several valid patterns based on Allen's relations were also found, but they mostly explained only fragments of the scenarios and showed less correspondence to the ground truth. This was partly explained by pattern fragmentation.

In [12] the rule sets are defragmented with disjunctions of patterns but this makes the result even longer and harder to understand. Take for example a rule disjunctively combining the 5 cases of Figure 13. Another solution to the problem is to use thresholds, effectively merging the problematic patterns. This does not solve the problems of ambiguity and comprehensibility, however, and poses the problem of threshold selection. Ambiguity could be reduced by splitting patterns with potential high variability into several different patterns, e.g. using the mid points [22]. But using 49 instead of 13 relations with many additional conditions requiring equality of time points will in turn increase effects of pattern fragmentation.

The reason for the TSKR being more expressive than Höppner's representation using Allen's relation is the permission of partial order and of subintervals in patterns. One could also mine partially related Allen's patterns by allowing blanks in the matrix of pairwise relations. This would make the mining even more costly. Further, imagine being given a listing of 10 pairwise relations of 5 intervals and the task to draw a qualitative example of the pattern. This is not at all a trivial task, many dependencies need to be considered. In fact, this corresponds to the constraint satisfaction problem of temporal reasoning which is NP-complete [28]. The same task is rather easy given a description in the TSKR language and shouldn't this be the case in order to call a pattern *understandable*?

If the semantics of Allen's relations are explicitly desired for an application, one should be aware of the negative effects of noise and the general disadvantages of this representation for knowledge discovery. For small patterns a qualitative visualization of the pattern might not adequately represent all instances found as there can be large quantitative differences. For larger patterns this effect may be weaker because with more pairwise relations there are fewer possibilities for quantitative variance. In addition to interval based pattern visualizations, the TSKR also offers a more abstract graph visualization of Phrases and Chords.

The TSKR was designed for unsupervised learning in noisy interval data. It can just as well be used with data where exact relations of interval endpoints matter. It is assumed in general, however, that the temporal process producing the symbolic interval is not chaotic and the properties described by Tones can be observed during intervals of a minimum length meaningful to the analyst. We believe that other data mining tasks like classification or anomaly detection can also profit from the high robustness and high understandability of the TSKR. Other possible extensions of the TSKR include the use of quantitative information [12], generation of implication rules [1, 12, 5, 30], and fuzzy itemsets and association rules [8].

7. SUMMARY

We presented a novel way of extracting understandable patterns from multivariate temporal data with the aim of knowledge discovery. We defined the Time Series Knowledge Representation (TSKR), a new pattern language for expressing temporal knowledge. Efficient algorithms for mining such patterns are available [17, 18]. The TSKR is more robust, more expressive, and better interpretable than previously proposed approaches using Allen's relations. Unsupervised pattern mining with the TSKR was demonstrated using video data. Compared to Allen, many fewer patterns needed to be analyzed with the TSKR and they were more specific and more accurate.

Acknowledgements: This work was done in the Databionic Research Group at the Philipps-University Marburg, Germany. We thank Alfred Ultsch for guidance and discussion during this time. We further thank Alan Fern for donating and explaining the video data.

8. REFERENCES

[1] R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. In *Proc. SIGMOD*, pages 207–216, 1993.

[2] M. Aiello, C. Monz, L. Todoran, and M. Worring. Document understanding for a broad class of documents. *IJDAR*, 5(1):1–16, 2002.

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

[4] R. J. Bayardo. Efficiently mining long patterns from databases. In *Proc. PODS*, pages 85–93, 1998.

[5] R. Bellazi, C. Larizza, P. Magni, and R. Bellazi. Temporal data mining for the quality assessment of hemodialysis services. *Artificial Intelligence in Medicine*, 34:25–39, 2005.

[6] G. Casas-Garriga. Summarizing sequential data with closed partial orders. In *Proc. SDM'05*, pages 380–391, 2005.

[7] P. R. Cohen. Fluent learning: Elucidating the structure of episodes. In *Proc. IDA'01*, pages 268–277, 2001.

[8] D. Dubois, E. Hüllermeier, and H. Prade. A systematic approach to the assessment of fuzzy association rules. *Data Mining and Knowledge Discovery*, 2006. to appear.

[9] A. Fern. *Learning Models and Formulas of a Temporal Event Logic*. PhD thesis, Purdue University, 2004.

[10] H. Grice. *Studies in the Way of Words*. Harvard University Press, 1989.

[11] G. Guimarães and A. Ultsch. A method for temporal knowledge conversion. In *Proc. IDA'99*, pages 369–380, 1999.

[12] F. Höppner. *Knowledge Discovery from Sequential Data*. PhD thesis, Technical University Braunschweig, Germany, 2003.

[13] P.-S. Kam and A. W.-C. Fu. Discovering temporal patterns for interval-based events. In *Proc. DaWaK'00*, pages 317–326, 2000.

[14] M. Last, Y. Klein, and A. Kandel. Knowledge discovery in time series databases. *IEEE Trans. on Systems, Man, and Cybernetics*, 31(1):160–169, 2001.

[15] M.-Y. Lin and S.-Y. Lee. Fast discovery of sequential patterns by memory indexing. In *Proc. DaWaK*, pages 150–160, 2002.

[16] H. Mannila, H. Toivonen, and I. Verkamo. Discovery of frequent episodes in event sequences. In *Proc. KDD'96*, pages 210–215, 1995.

[17] F. Mörchen. Algorithms for time series knowledge mining. In *Proc. KDD*, 2006.

[18] F. Mörchen. *Time Series Knowledge Mining*. PhD thesis, Philipps-University Marburg, Germany, 2006.

[19] P. Papaterou, G. Kollios, S. Sclaroff, and D. Gunopoulos. Discovering frequent arrangements of temporal intervals. In *Proc. ICDM*, pages 354–361, 2005.

[20] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. In *Proceeding ICDT'99*, pages 398–416, 1999.

[21] C. Rainsford and J. Roddick. Adding temporal semantics to association rules. In *Proc. PKDD'99*, pages 504–509, 1999.

[22] J. F. Roddick and C. H. Mooney. Linear temporal sequences and their interpretation using midpoint relationships. *IEEE TKDE*, 17(1):133–135, 2005.

[23] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proc. IEEE Symposium on Visual Languages*, page 336, 1996.

[24] J. M. Siskind. Grounding the lexical semantics of verbs in visual perception using force dynamics and event logic. *Journal of Artificial Intelligence Research*, 15:31–90, 2001.

[25] S. G. Sripada, E. Reiter, and J. Hunter. Generating English summaries of time series data using the Gricean maxims. In *Proc. KDD'03*, pages 187–196, 2003.

[26] A. Ultsch. Eine unifikationsbasierte Grammatik zur Beschreibung von komplexen Mustern in multivariaten Zeitreihen. personal notes, 1996. German.

[27] A. Ultsch. Unification-based temporal grammar. Technical Report 37, CS Dept., Philipps-University Marburg, Germany, 2004.

[28] M. Vilain, H. A. Kautz, and P. G. van Beek. Constraint propagation algorithms for temporal reasoning. *Reading in Qualitative Reasoning about Physical Systems*, pages 373–381, 1989.

[29] R. Villafane, K. Hua, D. Tran, and B. Maulik. Knowledge discovery from series of interval events. *Journal of Intelligent Information Systems*, 15(1):71–89, 2000.

[30] E. Winarko and J. F. Roddick. Armada - an algorithm for discovering richer relative temporal association rules from interval-based data. *Data & Knowledge Engineering*, 2006. to appear.

[31] X. Yan, J. Han, and R. Afshar. CloSpan: Mining closed sequential patterns in large datasets. In *Proc. SDM'03*, pages 166–177, 2003.

[32] M. J. Zaki and C.-J. Hsiao. Efficient algorithms for mining closed itemsets and their lattice structure. *IEEE TKDE*, 17(4):462–478, 2005.