

Highlighting Conflict Dynamics in Event Data*

Ulrik Brandes[†]

Daniel Fleischer[‡]

Jürgen Lerner[§]

Department of Computer & Information Science
University of Konstanz, Germany

ABSTRACT

We present a method for visual summary of bilateral conflict structures embodied in event data. Such data consists of actors linked by time-stamped events, and may be extracted from various sources such as news reports and dossiers. When analyzing political events, it is of particular importance to be able to recognize conflicts and actors involved in them. By projecting actors into a conflict space, we are able to highlight the main opponents in a series of tens of thousands of events, and provide a graphic overview of the conflict structure. Moreover, our method allows for smooth animation of the dynamics of a conflict.

Keywords: information visualization, text mining, event analysis, time-dependent visualization

1 INTRODUCTION

Event data are among the most widely used indicators in quantitative international relations research. While such data can be extracted efficiently from news reports and dossiers, there is a need to automatically process the obtained series of events. Even when aggregating actors (like different people representing the same organization) and focusing on specific global regions (like the Balkans or the Persian Gulf) and periods of time (like 10 or 20 years) there are typically several hundreds of actors engaged in tens of thousands of events.

Of special interest in social and political event analysis is the identification of conflicts and the division of parties engaged into opposing sides. While the ultimate goal is conflict prediction, a first step consists in the detection and analysis of historical or ongoing conflicts. This is particularly difficult if the parties involved are not, or only partially, known to the analyst.

Previous work is overwhelmingly statistical (see, e.g., [10]), but there is a growing interest in combining event data analysis with techniques from network analysis (see, e.g., [9]). The idea is that the macroscopic structure emerging from bilateral events provides additional information that can be exploited using network-analytic techniques.

We present a method that, given a list of events, constructs a sequence of networks which in turn is converted into an animated scatterplot. (Scatterplots are widely used in statistical graphics, see, e.g., [2, 3].) The resulting video summarizes graphically the dynamics of major conflicts over a potentially long period of time. From this video, an analyst can recognize or discover the major actors engaged in conflict during certain periods of time. The observer is also enabled to detect time-points where the conflict structure changes significantly. Since our animation is smooth by design, it

can be recognized easily which actors enter or leave a conflict during transitions.

Thus, usage scenarios for our method include the exploration of unfamiliar political situations, identification of crucial turning points, and briefing of decision makers. As a by-product, we also experienced that the video can draw attention to inconsistencies in the input data.

This paper is organized as follows. In Sect. 2 we provide background information on the type of data that is being analyzed. Our method for visualizing the conflict structure embodied in a set of events is introduced in Sect. 3 and extended to smooth animation of event series in Sect. 4. The utility of our method is illustrated on event data from the Balkans and the Persian Gulf in Sect. 5. We conclude with a discussion of open problems and future work.

2 EVENT DATA

Our method is applicable to event data given as a series of pairwise interactions. Although it is independent of the data format, we will focus on a particular coding scheme to make the exposition more concrete.

The Kansas Event Data System (KEDS) [8] is a software tool that automatically extracts events from text. In Sect. 5 we will use KEDS data for the Persian Gulf and Balkans regions.

Formally, an event series is a sequence a_1, \dots, a_k of tuples $a_i = (t_i, s_i, o_i, c_i)$, where

- t_i is the *time-stamp* (date),
- s_i is the *subject* (source),
- o_i is the *object* (target), and
- c_i is the *code* (type)

of event a_i . Dates are given by the day and actors may be aggregated. We say that actors s_i and o_i are *involved* in event a_i . Events are classified using to the World Event/Interaction Survey (WEIS) codes [7]. Each event is assigned Goldstein weights [4] $-10 \leq \omega(a_i) = \omega(c_i) \leq 10$, which are psychometrically determined scores depending only on the type of event. A positive (negative) weight indicates the degree of *cooperation* (*hostility*) of the corresponding type of event.

The following excerpt indicates the coding of actors in the Balkans data.

```
NATO_OFFICIAL [NAT]
NATO-LED_STABILIZATION_FORCE_IN_BOSNIA [NAT]
BELGRADE [YUG <920427] [SER]
MILOSEVIC [SERGOV 890101-971230]
[FRYGOV 971231-001005]
[SERSM >001006]
```

Several tokens may be interpreted as referring to the same aggregated actor. In the above excerpt, NATO (NAT) is represented by (among others) potentially unnamed officials and SFOR. On the other hand, the same token may represent different actors at different times. As its capital, Belgrade is understood to represent

*Research partially supported by DFG under grant Br 2158/1-2

[†]e-mail:Ulrik.Brandes@uni-konstanz.de

[‡]e-mail:fleische@inf.uni-konstanz.de

[§]e-mail:lerner@inf.uni-konstanz.de

Yugoslavia until the end of April 1992, but after that it is coded as referring to Serbia. Slobodan Milosevic switches roles more often, representing the Serbian government until December 1997, the government of the Federal Republic of Yugoslavia until October 2000, and after being captured only himself.

Given an actor coding, textual statements are parsed into events like the following, all dated 22 June 1991.

910622	SLO	YUG	198	(WITHDRAW FROM)
910622	USA	ALB	032	(VISIT)
910622	ALB	USA	033	(RECEIVE)
910622	YUGCRO	SLO	031	(MEET)
910622	SLO	YUGCRO	031	(MEET)

The first event is an action initiated by Slovenia (active) and directed at Yugoslavia (passive). In addition to the event code, a textual description of the type of event is given in parentheses. The next two lines show how asymmetric events are coded when both actors are active (receiving a visitor is a cooperative action), and the last two lines serve to illustrate that symmetric events are split in two directed ones.

Examples for weights associated with event types are the following.

072	EXTEND MIL AID	8.3
093	ASK MATERIAL AID	3.4
120	ACCUSE	-2.0
223	MIL ENGAGEMENT	-10.0

Apparently, extending military aid is a highly cooperative action, whereas accusations are mildly and military engagement is extremely hostile. To analyze conflict, we will only make use of negatively weighted events.

To detect emergent patterns and utilize indirect relations, we transform the data into a network. Any set $A = \{a_1, \dots, a_k\}$ of events gives rise to an undirected interaction graph $G(A) = (V, E)$ in the following way. Every actor involved in any event constitutes a vertex, i.e. $V = \bigcup_{i=1}^k \{s_i, o_i\}$. There is an (undirected) edge $e = \{u, v\}$ if there is an event involving both u and v , and we assign a weight $\omega(e)$ that is minus the sum of all negative weights on events involving these two actors. Figure 1 shows an example of a conflict graph drawn by standard force-directed layout techniques [6]. The complexity of Fig. 1 already indicates the insufficiency of general-purpose graphdrawing techniques and the need for other analysis and visualization methods that are more appropriate for this application. In Sect. 3 we develop a method that is able to extract the dominant conflict structure, filters out minor actors, and finally produces a less complex image that is easy to understand.

It is unlikely that a focused data set yields an interaction graph with more than one significant non-trivial connected component. However, since connected components can be analyzed separately, we may safely assume that all interaction graphs are connected, anyway.

3 VISUALIZING BILATERAL CONFLICTS

In this section we focus on extracting the structure of conflicts from static event data, i.e. we ignore time-stamps and consider the data to be given as a set (rather than a sequence). We will use the corresponding interaction graph to determine the main opposition of actors, where the assumption is that conflicts are predominantly bilateral. The static methods that are developed in this section will be augmented to include dynamics in Sect. 4.

A straightforward attempt to determine the two opponent groups of a bilateral conflict is to try to divide the actor set V into two disjoint subsets U and W , such that for each edge $e = \{u, w\}$ we have $u \in U$ and $w \in W$, i.e. conflicts are only between and not within

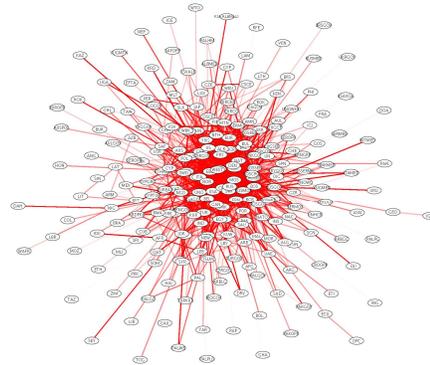


Figure 1: Hostile interaction in the Balkans from 1991 until 1997. Saturation of the edges is proportional to cumulative hostility weights.

the two groups. See Fig. 2 for an example partition of a small subset of actors in the Balkans according to events of 1991.

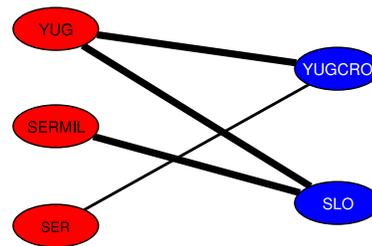


Figure 2: The subgraph of hostile interaction induced by selected Balkan actors during 1991 constitutes a bipartite conflict structure.

The existence of a perfect partition into such sets corresponds to the graph being bipartite and is thus easy to test for. However, larger data sets will seldomly result in purely bipartite structures. In fact, our experiments provide evidence that the interaction graphs are far from bipartite for almost all reasonable selections of actors and periods of time. Thus, in order to make the idea applicable to empirical data, the concept of a bipartition must be relaxed.

A possible relaxation is to determine a partition $V = U \cup W$ such that the sum of edge weights between the two sets is maximized (in a perfect bipartition, this sum is over all edges), or, equivalently, the sum of edge weights inside of the two sets is minimized. However, this is the well-known MAXCUT problem, which is NP-hard. We make use of a different relaxation that poses no such algorithmic problems and filters out unimportant actors on the fly.

3.1 Structural Projections

To arrive at a relaxed bipartition, we employ the recently introduced concept of structural projections (and the closely related structural similarities) [1], which have a sound theoretical basis and lead to an efficient algorithm that is easy to implement. We first give an intuition of structural projections, and then introduce them formally.

Instead of mapping actors to one class or the other, structural projections yield real-valued *degrees of membership* in classes. For a relaxed bipartition, actors are projected into a two-dimensional conflict space such that actors mapped mostly into one dimension have major conflicts with actors mapped mostly into the other dimension, but only minor conflicts with actors mapped into their own dimension.

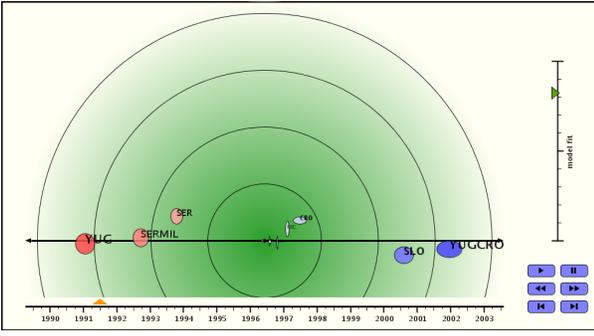


Figure 3: Structural projection of hostile Balkan events during 1991. The dominant conflicts are exactly those of Fig. 2.

An example of such a real-valued projection is shown in Fig. 3. The figure shows the conflict structure of all actors involved in the Balkans in the mid of 1991. The five most important actors are exactly those of Fig. 2. Note that the degree of membership assigned to actors varies. E.g., Serbia (SER), though closer to the red group, does not fit exactly into the bipartite structure, because the reported hostile activity is less than that of the other red actors. Many of the unimportant actors close to the origin are filtered out because their level of hostility is not sufficient to place them prominently in one group or the other. Thus, our method not only determines a relaxed bipartition, but also indicates which actors are most responsible for the division.

The method sketched above is a specific use of a more general framework. We next introduce its essentials, and refer to [1] for further details.

A weighted graph $G = (V, E)$ can be represented by its symmetric adjacency matrix $A = (a_{uv})_{u,v \in V}$ with rows and columns indexed by V and entries $a_{uv} = \omega(\{u, v\})$ corresponding to the weight of the edge between the two endpoints. The following definition specifies how vertices (i.e., actors) are projected continuously to k classes, and thus generalizes k -partitions.

Definition 1 Let G be a graph with n vertices and adjacency matrix A . A projection is a real $k \times n$ matrix P that has orthonormal rows. The quotient of G modulo P is the graph that has the k different classes as vertex set and whose adjacency matrix is $B = PAP^T$.

Quotients model the relation between classes induced by P and G . Two classes C_1 and C_2 are connected by an edge whose weight is the average weight of the edges connecting actors in C_1 to actors in C_2 (weighted by their respective degree of membership). In Fig. 2, e.g., the quotient of the projection to the red and blue class is a single edge connecting the two.

The next condition ensures that a projection is derived from the graph structure. It formalizes the idea that actors are equivalent when they are equally connected to equivalent actors. In Fig. 2, e.g., two red actors are both connected to some (but not necessarily the same) blue actors.

Definition 2 A projection P is called structural for G if $PA = BP$.

The following theorem is a characterization of all structural projections for a given quotient and hence essential for our purposes. It shows how a projection can be chosen such that it yields a pre-specified quotient.

Theorem 3 ([1]) Let G and R be two weighted graphs with adjacency matrices A and B . A projection P is structural for G with quotient R , if and only if the row-space of P is generated by eigenvectors of A associated with all eigenvalues of B .

We will use this theorem in the next section to determine a structural projection into two-dimensional space such that the resulting quotients are “as bipartite as possible.”

3.2 Projecting into Conflict Space

Consider the parameterized graph in Fig. 4. It consists of two vertices representing the opposition of two groups, an edge with large weight w representing the hostility of events involving actors in opposite groups, and two edges with small weight c representing the hostility of events involving actors within a group. The two-dimensional vector space associated with its adjacency matrix is called *conflict space*, and we will apply Theorem 3 to determine a structural projection into it.

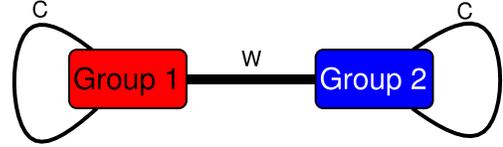


Figure 4: Quotient R representing the conflict space. Weights w and c are average weights of edges between and within the two groups.

The single-edge quotient of a bipartite graph is augmented by loops of weight c , because the interaction graph will almost never be bipartite at any given point in time. For this quotient, we need to find a (continuous) projection of actors onto these two classes, such that w is as large as possible and c as small as possible.

The eigenvalues of the adjacency matrix B of the quotient shown in Fig. 4 are

$$\begin{aligned}\lambda &= w + c \\ \mu &= -w + c.\end{aligned}$$

From a different perspective, parameters w and c are given by the two eigenvalues λ and μ of B as

$$\begin{aligned}w &= \frac{\lambda - \mu}{2} \\ c &= \frac{\lambda + \mu}{2}.\end{aligned}$$

By Theorem 3, a projection P onto the graph R is found by eigenvectors of A associated with λ and μ . Since our goal is to maximize w and minimize c , we choose λ as large and μ as small as possible, i.e. we take eigenvectors of A associated with the largest and smallest eigenvalue λ_{\max} and λ_{\min} . We thus have the following result.

Theorem 4 The structural projection with quotient R from Fig. 4 that maximizes w (and minimizes c) is the orthogonal projection onto the two eigenvectors of A associated with the largest and smallest eigenvalue.

The above derivation also shows that $c = 0$ if and only if $\lambda_{\min} = -\lambda_{\max}$, which is well-known to hold if and only if the graph is bipartite. Although this is almost never the case in empirical data, our experiments provide evidence that often c is very small compared to w . To assess the degree to which the data matrix A is bipartite, we introduce the following index.

Definition 5 Let λ_{\max} and λ_{\min} be the largest and smallest eigenvalue of the adjacency matrix of a graph G . The bipartiteness of G is defined as

$$\beta(G) = \left| \frac{\lambda_{\min}}{\lambda_{\max}} \right|.$$

Index $\beta(G)$ is between zero and one, and it is one if and only if the graph is bipartite.

3.3 Graphing Conflict Space

Using the projection specified in Theorem 4, we obtain a representation of the interaction graph in a two-dimensional conflict space, i.e. a scatterplot of the vertices with coordinates given by the eigenvectors of the adjacency matrix of the graph.

The coordinates of this scatterplot are transformed to indicate the degree of membership to each of the two classes. To do so, first consider what happens when the ideal, single-edge, conflict graph is projected onto itself. The normalized eigenvectors of the adjacency matrix of a graph consisting of only an edge are

$$x = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \text{and} \quad y = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix},$$

so that

$$\frac{x+y}{\sqrt{2}} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{and} \quad \frac{x-y}{\sqrt{2}} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

If x, y are the first and last eigenvectors of an arbitrary input graph, the lefthand side transformations thus yield real-valued degrees of membership in the two conflict groups. To obtain a horizontal opposition, we finally rotate the result by 90 degrees and scale the opening angle between the two coordinate axes by a factor of two (compare e. g., Fig. 6).

Because of an intricate relation between structural similarities and vertex centrality [1], the distance from the origin is an indicator of the actors importance. As is apparent in Fig. 3, our projections thus not only classify actors to one group or the other, but also distinguish between major and minor members of each group.

The graphical attributes of our visualization are determined as follows. Color coding is used to emphasize group membership (the degrees of membership determine a red and blue component) and importance (saturation). Vertex shape and size are used to add information about activeness of actors. *Activeness* is defined as the net weight of the events in which an actor is involved as the subject initiating the event. Symmetrically, *passiveness* adds weights of events received. The ratio between activeness and passiveness determines the aspect ratio of a vertex, so that aggressive actors that initiate hostile interactions, but are not the subject of retaliation are high and narrow. The size of a vertex is the sum of the two and thus represents an actors *involvement* in a conflict structure.

We summarize the algorithm for scatterplots representing the conflict structure of a graph G of hostile interactions with adjacency matrix A :

1. Compute maximum and minimum eigenvalues λ_{\max} and λ_{\min} of A together with associated eigenvectors v_{\max} and v_{\min} .
2. The projection is given by the $2 \times n$ matrix P with $x = (v_{\max} + v_{\min})/\sqrt{2}$ in the first and $y = (v_{\max} - v_{\min})/\sqrt{2}$ in the second row. Rotate coordinates by 90 degrees and scale the opening angle between the two coordinate axes as described above.
3. The bipartiteness is given by $\beta = -\lambda_{\min}/\lambda_{\max}$.
4. Draw each actor v as a ellipse with coordinates and color values proportional to the two values in the v 'th column of P , height proportional to activeness and width proportional to passiveness.

Any eigenvector algorithm for real symmetric matrices can be used in Step 1 (see, e.g., [5]), and there are many readily available software packages. In our examples we indicate the fit of the bipartite model using a bipartiteness gauge.

Figure 5 shows the projection into conflict space for events of the Balkan Conflict from 1989 until 2003. The bipartiteness of this

projection is rather low (only around 0.42), indicating many conflicts within the two groups. Despite of the low bipartiteness, our method still yields two reasonable opponent groups: Serbia (SER) and the Serbs in Bosnia (BOSSER) opposed to Bosnia (BOS) and Croatia (CRD). The NATO (NAT) is opposed to (SER) and (BOSSER), due to the massive air strikes in 1994 and 1995.

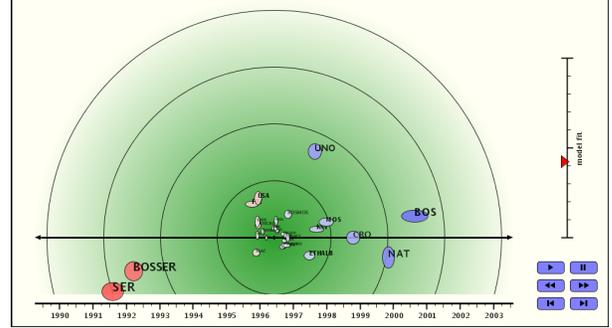


Figure 5: Balkan Conflict 1989–2003.

Figure 6 shows the projection for the conflicts in the Persian Gulf from 1979 until 1999. The two major opponents for the whole period of time are clearly the Iran and Iraq, due to their war from 1980 to 1988. The bipartiteness is, although much higher than in Fig. 5, considerably distant from one. The reason for this is that the USA has strong negative ties to both Iran and Iraq, thus these three actors form a negative triangle. The fact that the aggregated weight of negative events between the USA and Iraq (mostly following Iraq's invasion of Kuwait in 1990) is larger than the negative weight between the USA and Iran, is the cause that USA is mapped farther from Iraq and thus, necessarily, closer to Iran.

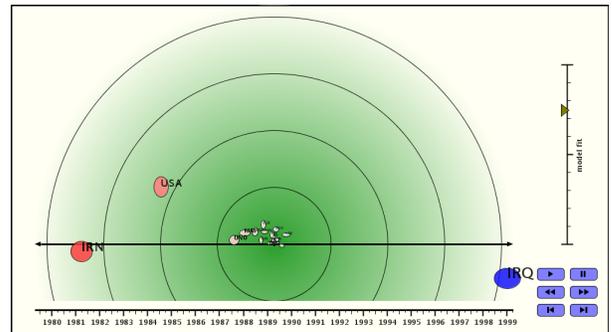


Figure 6: Gulf Conflict 1979–1999.

To obtain more detailed insight in the evolution of both conflicts, we will introduce a technique for smooth animation of the above type of scatterplots for limited periods of time.

4 ANIMATING CONFLICT DYNAMICS

In order to show how conflicts evolve over time, the event graph G is used to generate a sequence of graphs G_t , each of which represents the view on the set of events at the specific time t . A graph G_t yields one frame of the final video. How the events are viewed at a certain time-point is determined by a *scaling function* $\delta: \mathbb{R} \rightarrow \mathbb{R}_{\geq 0}$, which models how events move into the data when time increases and how they fade out. Examples of possible scaling functions are

“rectangular” functions of radius r

$$\delta_r(t) = \begin{cases} 1 & \text{if } |t| \leq r \\ 0 & \text{else,} \end{cases}$$

having the meaning that events move in all at once if their time-stamp is within r of the current time and that they have no influence if it is outside. We will later use a continuous scaling function that results in a smoother animation.

For a fixed δ and t , the graph $G_t = (V, E, \omega_t)$ is defined as follows. The actor set V and the edge (or event) set E are the same as for the input graph G . The weight $\omega_t(e)$ of an event e at time t is defined to be $\omega_t(e) = \omega(e) \cdot \delta(t_e - t)$, that is, the weight of e at time t is its absolute weight $\omega(e)$ times a scaling factor which is dependent on the difference between the time-stamp t_e of the event and the fixed time point t . The graph G_t may be reduced by removing events with zero weight, as well as isolated actors, since these do not influence the analysis and would be invisible in the final video.

Given a graph G , representing a list of events, the movie is generated by the following steps.

1. Select a sequence of time-points $t_1 < \dots < t_N$ in a given time interval.
2. For each i from 1 to N

Compute the visualization of the graph G_{t_i} .

3. The images for all time-points yield the frames of the video.

In order to maintain the overall appearance of the frames one further detail has to be taken into consideration. If v is an eigenvector of A associated to eigenvalue λ , then so is $-v$. Thus the eigensolver algorithm could return either v or $-v$ as a solution to the eigenvalue problem. To prevent that this assignment switches from one frame to another (which would result in reversing one axis of the coordinate system from one frame to another) we have to ensure that the eigenvectors we use point in a well-defined direction.

The canonical direction for the eigenvector v_{\max} associated to the largest eigenvalue is simply the direction in which each entry of v_{\max} is positive. (It is standard knowledge in algebraic graph theory that all entries of this eigenvector have the same sign.) We define the canonical direction for the eigenvector v_{\min} for time-step t recursively by the direction of this eigenvector for time-step $t - 1$. The direction of v_{\min} is chosen such that the angle between v_{\min} at time t and v_{\min} at time $t - 1$ is smaller than 90 degrees. Thus, only the direction of v_{\min} for the very first time-step is arbitrary. This translates to the fact that there is no absolute meaning attached to the two opponent groups. A second computation of the movie could reverse the red and the blue groups, but then it has to reverse the assignment for all actors, which results in the same opponents.

5 APPLICATION EXAMPLES

We apply our method to visualize two selected data sets from the Kansas Event Data System (KEDS)¹ in a prototypical implementation. We generate the graphs G_t with a triangular shaped scaling function with time radius r

$$\delta(t) = \begin{cases} (t+r)/r & \text{if } |t| \leq r \text{ and } t < 0 \\ -(t-r)/r & \text{if } |t| \leq r \text{ and } t \geq 0 \\ 0 & \text{if } |t| > r \end{cases}$$

This function does not consider events with a time-stamp more than r away from the current time-point. Events move into G_t linearly until t is larger than their time-stamp. Then, they fade out linearly

¹<http://www.ku.edu/~keds/>

until they have zero weight. For the two investigated data sets we used for r a period of 90 days.

This simple but continuous scaling function already yields a smooth animation, since the coordinates depend continuously on the entries of the input matrix. The usage of more sophisticated functions (like Gaussian curves) could be considered in future research.

In the video, the current time is shown by a triangle. The current bipartiteness β_t is shown in a scale ranging from zero to one. This scale is green if β_t is close to one and red if it becomes low.

The two coordinate-axes both point into horizontal directions, one to the left and one to the right. We have chosen to take this coordinate system instead of the more usual one (one axis vertical and one horizontal) to prevent misguided interpretation of superiority of one group over the other.

The animations are realized in SVG (Scalable Vector Graphics, see W3C Recommendation²) format, thus they can be viewed on any web browser with an appropriate plug-in.³ The file for the Balkan region, e. g., covers about 5000 days using 1000 key frames to ensure smooth animation. Buttons in the lower right corner allow to pause, fast forward, rewind, etc. the animation.

5.1 Balkan

The KEDS Balkans data⁴ contains 78,668 events for the major actors (including ethnic groups) involved in the conflicts in the former Yugoslavia. Coverage is April 1989 through July 2003. The following figures show selected time-points of the Balkans conflict.

Figure 3 shows the war in Slovenia and Croatia in 1991. Yugoslavia (YUG) and the Serbian military (SERMIL) are clearly opposed to Slovenia (SLO) and Croatia (YUGCRO).

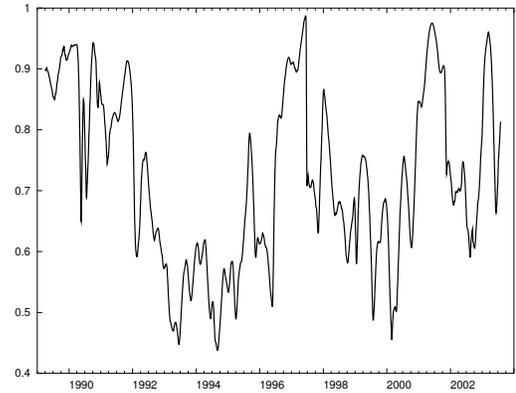


Figure 7: Bipartiteness profile of the Balkan Conflict.

The varying degree of polarization can be inferred from the model fit indicator curve in Fig. 7.

An important change in the conflict structure is shown in Figs. 8, 9, and 10. Figure 8 shows the NATO bombings in 1995. The Dayton peace talks (Fig. 9) took place in December 1995. After this, events in the Balkans calmed down and the media focused on the conflict between Turkey and the Kurds (Fig. 10).

²<http://www.w3.org/TR/SVG/>

³The animations referred to in this paper are available from <http://www.inf.uni-konstanz.de/algo/research/conflict/>

⁴<http://www.ku.edu/~keds/data.dir/balk.html>

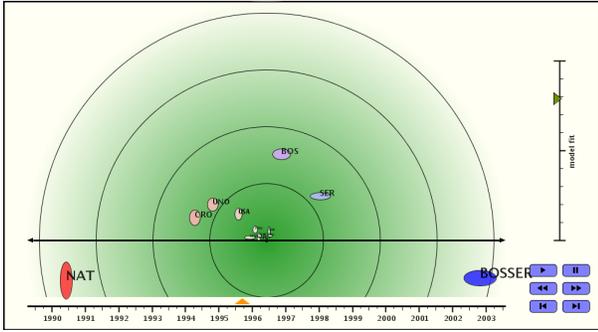


Figure 8: NATO bombing in Bosnia (1995).

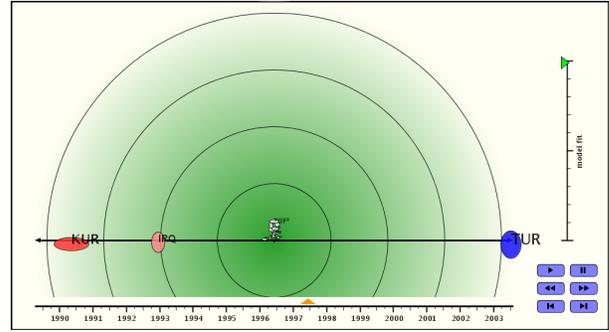


Figure 11: Conflict between Turkey and the Kurds.

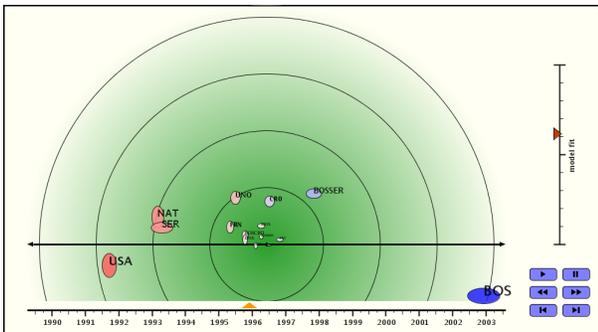


Figure 9: Dayton peace talks (December 1995).

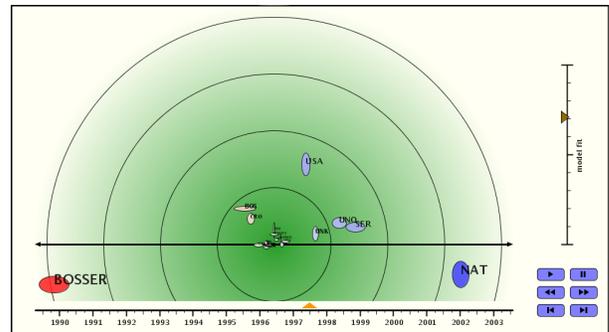


Figure 12: After the change of data source from Reuters North America to Reuters Business Briefing.

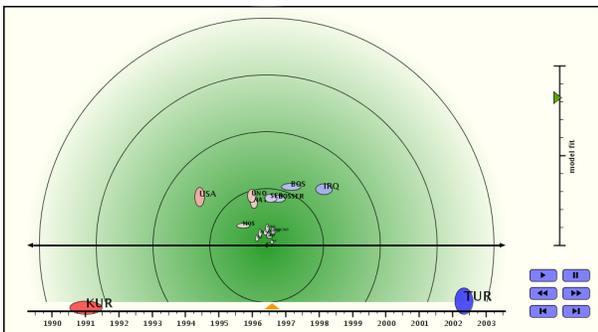


Figure 10: After the Dayton peace agreement, a new conflict emerges.

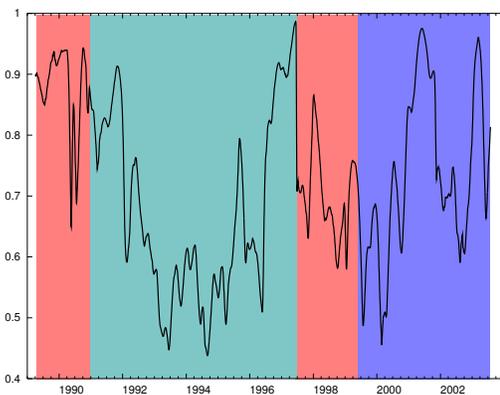


Figure 13: Sources for KEDS data on the Balkans. *Red*: Reuters Business Briefing, *green*: Reuters North America, *blue*: Agence France Press.

The conflict between Turkey and the Kurds also exemplifies a problem with the data that we were not aware of before seeing the animation. In July 1997, there is an abrupt change in media coverage. Figures 11 and 12 show the conflict structure before and after this change, with only sixteen days in between. The change is also visible in the bipartiteness curve, where the highly bipolar situation rapidly changes into a more complex one.

Since we knew of no prominent historic events explaining the change, we went back to the data and found that this is precisely the time when KEDS sources change from Reuters North America to Reuters Business Briefing, with the latter apparently not covering the conflict.

5.2 Persian Gulf

The KEDS Gulf data set⁵ contains 57,000 events for the states of the Gulf region and the Arabian peninsula for the period 15 April 1979 to 31 March 1999.

Figure 14 shows the so-called Iran hostage crisis (U.S. diplomats taken hostage after/during the Iranian revolution) in 1979.

In September 1980, the Iraqi invasion of Iran (shown in Fig. 15) took place. In the movie it can be seen between the two frames in Fig. 14 and Fig. 15 how the USA moves out and Iraq moves into the group opposing Iran. Bipartiteness remains very high during this period, i. e., the conflicts are almost all between the two opposing groups. The projection shown in Fig. 15 remains almost unchanged until 1987, although Iraq and Iran switch several times from being the main source to being the main target of the hostilities and back.

In 1987 and 1988 the U.S. engaged in a series of naval battles with Iranian forces. Figure 16 shows both the USA and the Iraq being opposed to Iran. Again, the bipartiteness for this projection is close to one; in particular there are no conflicts between the USA and Iraq during that time.

On 2 August 1990, Iraqi troops invaded Kuwait. A coalition of 34 countries, most notably the U.S., mobilized troops in Saudi Arabia and in the Persian Gulf. Figure 17 shows the Kuwait as a very broad and flat actor (target of aggression) on the same side as the USA and opposite to Iraq.

Starting from the end of 1991 a repeated switch between the two conflict pairs USA vs. Iraq on one hand and Israel vs. the Lebanon on the other can be observed. Figures 18 and 19 shows such a change which took place in 1996.

This oscillation is not an artefact of our method but can actually be found in the data. We calculated the aggregated negative weights of the edges (ISR,LEB) and (IRQ,USA) and plotted them in Fig. 20. For example, the weight of (ISR,LEB) is significantly higher than the weight of (IRQ,USA) at the beginning of 1996. This priority becomes inverted in the second half of 1996, which is the cause for the change from Fig. 18 to Fig. 19.

6 DISCUSSION

We have presented a method for designing scatterplots that represent the conflict structure embodied in event data, and a method for smoothly animating these scatterplots to highlight conflict dynamics. There are several interesting avenues for extending and improving our visualizations which will be addressed in further research.

A straightforward extension is secondary analysis of our dynamic projections. For instance, it might be interesting to focus on particular actors and follow their trajectories in the animated scatterplot and relate it to their involvement profile. Sensitivity analysis may point to crucial events or actors that trigger major changes

in the structure. Ultimately, combinations of these analyses could serve as early warning indicators.

The use of structural projections is not limited to two dimensions. If, for instance, the bipartiteness indicator is low, a third dimension could be opened in conflict space, yielding three groups that are in pairwise opposition. For example the situation in the Persian Gulf for the period from 1979 to 1999 (shown in Fig. 6) could be represented much better as a triangle formed by the USA, Iraq, and Iran, since these actors have mutually strong negative ties. The main difficulty will be to identify the appropriate eigenvectors to be used in the projection. Moreover, if a time-window graph contains several, say k , independent major conflicts, it should be projected using eigenvectors of the k largest and smallest eigenvalues. For example, the two independent conflict pairs (ISR,LEB) and (IRQ,USA), which are shown in Figs. 18 and 19 could both be represented in one image if we project to the two largest and two smallest eigenvalues. The main challenge will be to represent the resulting four-dimensional conflict space in a two-dimensional image.

On the other hand, we may extend our data basis by also considering events with positive weights. Natural groups are identified by clustering actors based on cooperation, and these can be compared to or integrated with a partitioning based on hostility.

Acknowledgement. We thank the anonymous reviewers and Michael Balzer for comments on an earlier version of the animation design.

REFERENCES

- [1] Ulrik Brandes and Jürgen Lerner. Structural similarity in graphs. In *Proc. 15th Intl. Symp. Algorithms and Computation (ISAAC'04)*, pages 184–195, 2004.
- [2] John M. Chambers, William S. Cleveland, Beat Kleiner, and Paul A. Tukey. *Graphical Methods for Data Analysis*. Wadsworth, 1983.
- [3] William S. Cleveland and Robert McGill. The many faces of a scatterplot. *Journal of the American Statistical Association*, 79(388):807–822, 1984.
- [4] Joshua S. Goldstein. A conflict-cooperation scale for WEIS international events data. *Journal of Conflict Resolution*, 36(2):369–385, 1992.
- [5] Gene H. Golub and Charles F. van Loan. *Matrix Computations*. John Hopkins University Press, 1996.
- [6] Michael Kaufmann and Dorothea Wagner, editors. *Drawing Graphs*. Springer Verlag, 2001.
- [7] Charles A. McClelland. World event/interaction survey codebook (icpsr 5211), 1976.
- [8] Phillip A. Schrodt, Shannon G. Davis, and Judith L. Weddle. Political science: KEDS—a program for the machine coding of event data. *Social Science Computer Review*, 12(3):561–588, 1994.
- [9] Thomas Widmer and Vera Tröger. Event data based network analysis. Paper presented at the 45th Annual Convention of the International Studies Association, Montréal, 2004.
- [10] Kazuo Yamaguchi. *Event History Analysis*. Sage, 1991.

⁵<http://www.ku.edu/~keds/data.dir/gulf.html>

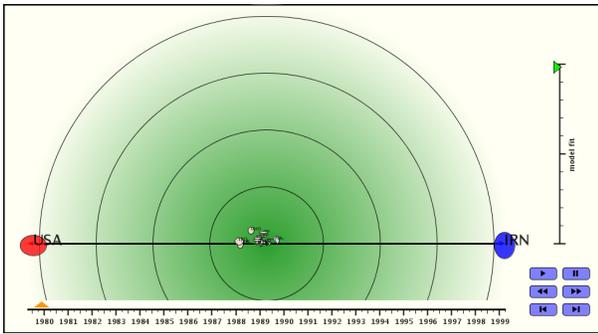


Figure 14: Iran hostage crisis.

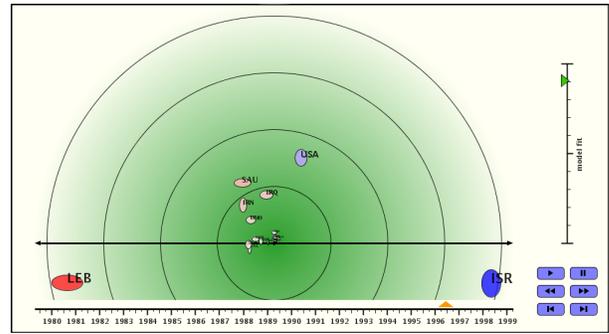


Figure 18: Conflict pair Lebanon vs. Israel.

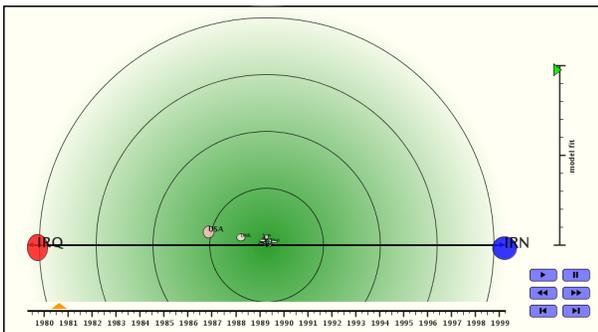


Figure 15: Iraqi invasion of Iran.

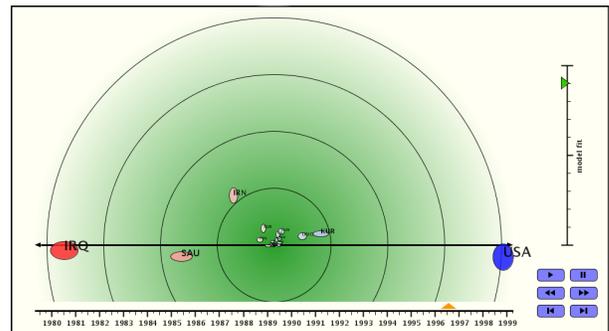


Figure 19: Conflict pair Iraq vs. USA.

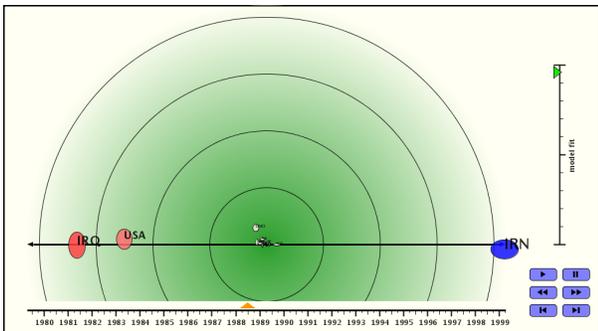


Figure 16: U.S. engaged in battles with Iranian forces.

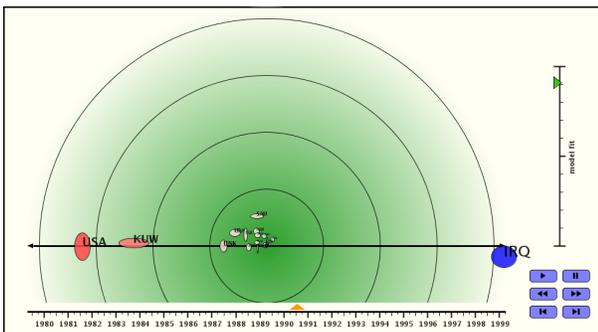


Figure 17: Iraqi invasion of Kuwait.

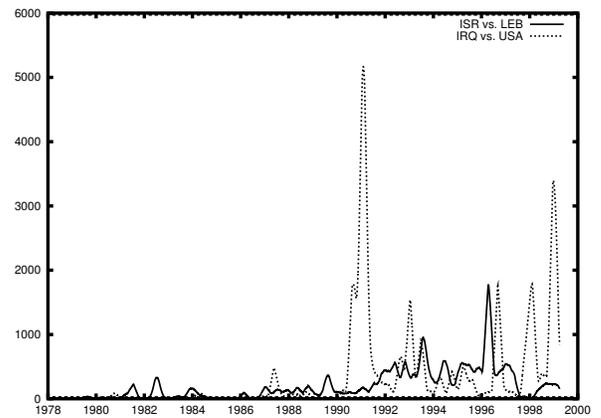


Figure 20: Aggregated weight of edges (ISR,LEB) and (IRQ,USA).