Two Visualization Tools for Analysis of Agent-Based Simulations in Political Science

R. Jordan Crouser* — Tufts University
Daniel E. Kee† — Tufts University
Dong Hyun Jeong‡ — University of the District of Columbia
Remco Chang§ — Tufts University

Abstract—Agent-based modeling (ABM) has become a key technique for modeling and simulating dynamic, complicated behaviors in social and political sciences. Analyzing and interpreting the results of these simulations can help scientists to better understand the forces at work in social and political systems, which can in turn enable them to better inform decision-makers and international policy. While there exist many robust toolkits tailored to developing and running these simulations, supporting systems for analyzing the results of these simulations are few and tend to be overly general. Lacking the appropriate tools and support, it has become difficult for social scientists to interpret the results of their increasingly complex simulations. To begin to bridge this gap between data generation and interpretation, we present two systems specifically designed to support inquiry and inference by social scientists using agent-based simulations to model political phenomena. In collaboration with domain experts, we designed these systems to provide interactive exploration and domain-specific data analysis tools. When tested by analysts in political science, we validated that these systems provide an efficient framework to explore individual trajectories and the relationships between variables as well as supporting hypothesis generation by enabling analysts to group simulations according to multidimensional similarity and drill down to investigate further.

Index Terms—Visual analytics systems, agent-based simulation, political science.

1 Introduction

Modeling and simulating complex systems of behavior is an integral component of research in the social and behavioral sciences. Research in these areas often utilizes a technique called *agent-based modeling* (ABM) for simulating and exploring behaviors such as collaboration [2], conflict [10], violence [5], and population change [3]. Agent-based models have also been used to identify a country's political patterns, which might indicate the imminence of civil unrest and help predict catastrophic events [9]. In ABM, a behavioral system is modeled as a collection of autonomous entities or *agents*. Each agent interacts with other agents according to a set of rules and goals, and over time it may influence and be influenced by the agents around it.

As computing power becomes more widely available, scientists are able to simulate increasingly complex systems, which in turn generate increasingly large data sets which must then be analyzed and interpreted. Understanding these simulation results can help social and political scientists to better understand the forces at work in complicated social behaviors, such as those leading to patterns of violence and socioeconomic repression, political unrest and instability, and even help identify factors that might lead to catastrophic events. Unfortunately, the existing methods and tools available to social scientists for analyzing simulation results are not able to support data sets of this magnitude, making it difficult for scientists to effectively interpret and analyze the results of these simulations [8].

Data size and dimensionality are not the only challenges facing social scientists when using large-scale agent-based simulations to model complex behaviors. ABM is a stochastic simulation technique, utilizing small random perturbations to the interaction rules and running each simulation hundreds or even thousands of times times to avoid local minima and to generate a distribution of sample behavioral patterns. Because of this, it is critical for analysts to be able to compare simulated behaviors between and across distinct runs, and to be able to piece together many simulation runs into a single, cohesive overview.

For these reasons, computational support and effective, domain-specific visualization tools are critical for effective analysis of these simulations. By understanding the patterns being modeled by the simulation, scientists can better understand the sociopolitical forces at work in real-world social and political systems, which can in turn enable them to better inform decision-makers and international policy. To begin to address this need, we formed a collaborative partnership with domain experts to investigate novel approaches for supporting the analysis process.

Our first action was to identify areas of critical need for our collaborators in political science. Through informal brainstorming sessions with a group of domain experts, we identified three areas that are insufficiently addressed by existing analytical support systems for use in exploring agent-based simulation data:

- Support for exploring the data set as a whole to generate initial hypotheses,
- Efficient mechanisms for the comparison of individual simulation runs,
- And the incorporation of domain expertise into the data analysis tool.

Using these three design considerations as a foundation, we designed and developed two interactive exploratory visual analytics systems to support analysis of agent-based models in political science. Each of these systems utilizes a coordinated multi-views architecture, allowing the analyst to customize the views to suit his or her analytical process. To evaluate these systems, we performed an expert analysis with a group of analysts working with data from an agent-based simulation of political violence and unrest in Thailand. From this analysis, we found that most analysts believed our systems to be invaluable tools that would significantly streamline their analytical processes. In collaboration with these experts, we also identified areas for further refinement of these tools.

The remainder of this paper is structured as follows: Section 2 characterizes the analytical processes of domain experts working with agent-based models and describes our target users. Section 3 provides

*e-mail: rcrous01@cs.tufts.edu
†e-mail: dan.kee@tufts.edu
‡e-mail: djeong@udc.edu
§e-mail: remco@cs.tufts.edu

Manuscript received 1 May 2011.
For information on obtaining reprints of this article, please send email to: tvcg@computer.org.

more detail about the considerations for designing visual analytics systems for the domain of political science. Section 4 presents the visual analytics systems we developed in collaboration with political scientists. Section 5 provides several scenarios in which our system can support the analysis of political science simulation data. In Section 6, we present an expert evaluation of the systems. In Section 7, we discuss the implications of these systems and present plans for future work in this area, and Section 8 concludes the paper.

2 Domain Characterization

Behavioral simulation analysis is an important component of social and political science research. In studying these models, scientists seek to uncover the sociopolitical and socioeconomic forces at work in controlling and influencing group behaviors and to make predictions about behavioral patterns using data collected in the real world. Better understanding of how these forces influence group behavior and the ability to make more accurate predictions can greatly influence how we view real-world behavioral systems and better inform decisions regarding domestic stability, foreign policy and more.

The first step in this process is constructing an accurate model. Existing political theories based on observed behaviors and interactions are developed into an agent-based model that is seeded with data collected in the field about political party affiliation, level of violence, protest, regional and local conflict, and more [1]. Using this data, the agent-based model then produces a large amount of data representing a distribution of possible behavioral patterns over a period of time. Analysts will then study this data to try to extract a cohesive story explaining the relevant interactions and to identify interesting or highly-likely outcomes.

While statistical analysis of the resulting data can be performed, it often proves insufficient. Because of the complex nature of these simulations, expert analysis of the resulting data sets is required to interpret the results as valid behavioral patterns and fully understand the forces controlling the interactions observed in the simulation. The size of the data is so large that it would require countless hours to examine by hand, and so the data must often be simplified and some of the subtlety sacrificed in the interest of conserving time and energy.

It is clear that computational support for these analytical processes is of critical importance to social and political science. Systems must be designed to support expert analysis that present the data in a manner that preserves information and allows for deep, low-level exploration while still maintaining a manageable overview. Further considerations for these systems will be presenting in the following section.

3 Design Considerations

There are several important considerations for the design of our systems. As indicated by our early conversations with our collaborators in political science, there are three main challenges in their analysis of agent-based simulation data: exploring the data as a whole to generate initial hypotheses at both the global and single-simulation level, comparing simulation runs, and incorporating domain expertise into data analysis.

3.1 Supporting Hypothesis Generation and Exploration

To begin to support the analytical process, we must first enable analysts to view the data in a meaningful way. At the overview level, this means giving the analysts an intuitive mechanism for examining the aggregated data, determine the similarities and differences between high-dimensional simulation runs, and helping them to identify trends and outliers for further exploration. At the detail level, we must provide an organized mechanism for drilling down into a single run, enabling analysts to explore the behaviors of a single set of conditions, as well as providing a useful tool for debugging the simulation. At present, analysts report using line graphs and statistical plots of each dimension in order to make comparisons, and comparing the values of individual variables to drill down into a single run. This process is laborious, highly error-prone, and fails to provide a real overall sense of how the dimensions interact with one another.

3.2 Comparison of Simulation Runs

As indicated by our collaborators, there are many instances where it is useful to be able to compare distinct simulations runs. For example, analysts might want to explore outliers to determine whether or not they represent legitimate but unlikely outcomes, or whether they are simply noise. However, at present there exists no supporting technology for analysts in this task. To compare simulation runs, the values of each variable must be compared independently, leaving the analyst without a holistic overview of the similarities and differences between the compared runs.

3.3 Leveraging Domain Expertise

As the result of early offline explorations of the data, Lustick et al. developed a model known as the Dynamic Political Hierarchy (DPH) [9] which combines theories of cross-cutting cleavages, nested institutions, and dynamic loyalties. In the referenced work by Lustick et al., this model was found to be incredibly useful in political forecasting. However, the ability to observe changes in DPH structures over time was limited and involved the laborious process of generating and comparing diagrams one at a time.

4 Visual Analytics Systems

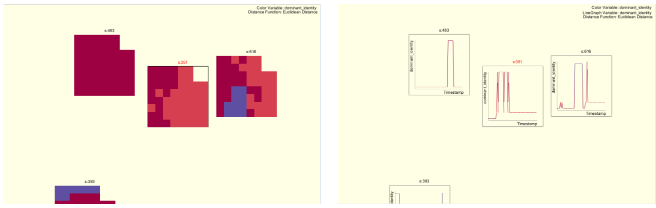
In order to address these challenges, we designed and implemented two systems to support political scientists in their analysis: MDSViz and SocialViz. With MDSViz, the analyst can explore the simulation space, finding patterns of similarity at an aggregated level and finding factors that dominantly affect to other agents. With SocialViz, the user's analysis can be more closely focused to a single simulation run, exploring relationships between time-steps and geographic regions of a selected simulation. Because the two systems are closely connected and maintain a consistent visual metaphor whenever appropriate, the analyst can easily transition from the analysis of multiple simulations with MDSViz to closely exploring a selected simulation with SocialViz. To better support the exploration and understanding complex political simulations, the overall interface designs of the two systems are based on a coordinated multi view (CMV) framework. The systems are developed using C++, OpenGL, and wxWidgets, and as such are deployable to any machine regardless of its operating system. In the following sections, we address the design of the two systems to reflect the needs identified by political scientists.

4.1 Data Overview and Dimension Reduction

Because the simulations maintain high dimensionality (roughly 1,000 simulations $\times$ 60 time steps $\times$ 351 attributes), a distance function is necessary to describe the similarity of two given states (see Section 7.1 for a detailed discussion on selecting a distance function). With an appropriate distance function, the multi-dimensional scaling (MDS) method is applied to reduce dimensionality of the data. MDS creates a distance matrix to explore similarities between pairs of states. Since the dimensionality is high in our input data (a distance matrix of 60,000 $\times$ 60,000 is possible), the system computes mean variance of each simulation by referencing all 60 time steps. Since every simulation is represented as mean values of 351 variables, the size of the distance matrix can be reduced to 1,000 $\times$ 1,000. Based on this generated distance matrix, MDS is performed to reduce the dimensionality of the simulations further.

4.2 Systems Overview

To support the analysis on complex political simulations, both systems are designed by following a coordinated multiple view (CMV) framework. MDSViz consists of two windows with two views in each window, and SocialViz contains three windows with either one or two views each. Within the CMV framework, any interaction with one view is immediately reflected to all the other views. To effectively coordinate each view, we implemented an interaction manager which handles all keyboard and mouse interactions. In addition, the selection operation in all views and the zooming-in/out mechanism in the Projection View and the Cluster View help users focus on their interested



(a) (b)

Fig. 2. In the Projection View, the glyph representation can be switched while navigating the projection space. (a) Pixel-oriented glyph that displays 60 time steps of each simulation by following the Hilbert curve ordering method. (b) Line graph that represents the temporal changes through time on a selected variable.

simulations or time steps. A detailed explanation about supported interactions in each system is included in following sections.

4.3 MDSViz System

MDSViz is designed to have a specific goal as to support the user perform a global understanding of all simulations. To support this global analysis on simulations, all simulations are managed to find similarities among simulations by applying a distance function and multi-dimensional scaling (MDS). To show relationships among simulations, the system consists of four views and two control panels (see Fig. 1).

4.3.1 Projection View

All simulations are represented by applying a distance function and multi-dimensional scaling (MDS) in the Projection View. Because there are limitations on applying MDS directly to large-scale input data, a statistical variance analysis is performed. Mean variance is computed to determine the center of the variable distribution for each simulation, and a distance function is then applied. Although finding a semantically meaningful distance function is important, identifying the appropriate contribution of all variables requires significant computational time. We use a simple Euclidean distance function and allow the user to manually control the weighted contribution of each dimension. MDS is then applied to reduce dimensionality of the simulations. By default, MDS runs 1,000 iteration times. Figure 1(a)-top shows all 1,000 political simulations.

As shown in Figure 1(a)-top, each simulation in the Projection View is displayed as a glyph. Each glyph is represented a pixel-oriented glyph by following a Hilbert curve ordering method to arrange each time step with color information. This technique has the advantage of providing continuous curves while maintaining good locality of information. For mapping each time step, we set the Hilbert curve order to 8 which covers up to 8×8 sizes. Color coding is then used to represent the selected variable at each time step. This parameter can be selected by the user in the the control panel (see Fig. 1(c)-top). Alternatively, the user can switch from the pixel-oriented glyph to a line graph representation (see Fig. 2).

4.3.2 Data View

Each simulation is controlled by 351 variables. To present the variables, we utilize a well-known visualization technique called parallel coordinates visualization. Although understanding 1,000 simulations with 351 variables through a parallel coordinates visualization can prove difficult because of a *cluttering* problem, this visualization technique is useful when the data exhibit patterns or underlying structure. Within the parallel coordinates visualization, a color attribute is selected by referencing the political structure of each simulation. Since most variables are mapped by the Dynamic Political Hierarchy (DPH), which characterizes the political structure of a country based on the relationships and strengths of individual political, racial, ideological, and religious groups, the frequency analysis counts the political structure in order to determine the most dominant political structure present in each simulation. For instance, if about 60% of the dominant identities in a simulation falls into a given category, that category is going to be selected as a representative political structure of the simulation. The corresponding color attribute is then used to represent the simulation as a line graph.

In the Data View, each line denotes one of the simulations. When the user highlights or selects simulations in the Projection View, the highlighted or selected simulations are emphasized by hiding all other simulations in the parallel coordinates. In addition, the mean variance of the highlighted simulation is displayed with a gradient color mapping method (see Fig. 1(a)-bottom). With this feature, the user can intuitively identify the variance over the course of 60 time steps in each simulation.

4.3.3 Cluster View

Once the analyst has identified and selected interesting simulations in the Projection View, all time steps in the selected simulations are represented in the Cluster View. Each simulation spans 60 time steps, and each time step is mapped to a unique circle in this view (see Fig. 1(b)-top). Similar to the Projection View, we apply MDS to reduce dimensionality. However, instead of applying MDS to all simulations, MDS is applied to all time steps in the selected simulations. Since each time step is considered as an individual data element in the Cluster View, similarities among 120 data elements will be computed when two simulations are selected. When multiple simulations are selected, representing all corresponding time steps in this Projection View makes it difficult for the the user determine which simulation produced each time step. To avoid this ambiguity, the convex hull is computed to form a group boundary around each simulation as shown in Figure 1(b)-top. Each convex hull indicates a cluster of each selected simulation. If the user highlights an item (i.e. time step) by hovering over the item, the convex hull of the corresponding simulation will also be highlighted.

4.3.4 Temporal View

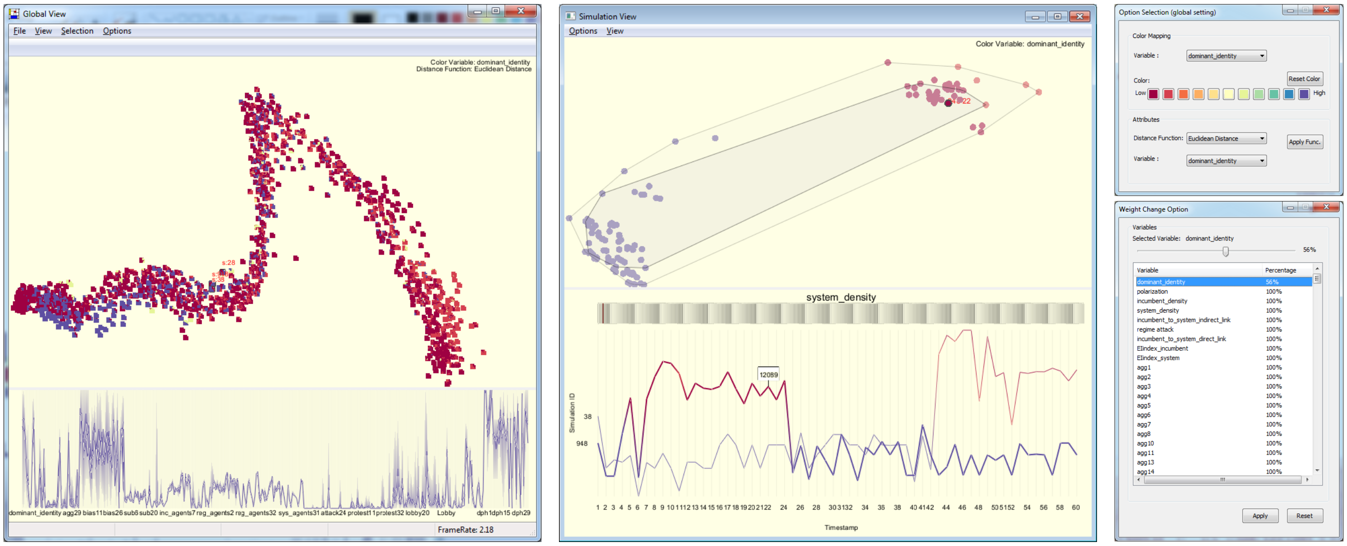
In the Temporal View, all attributes related to each time step are displayed in a parallel coordinates visualization. As shown in Figure 1(b)-bottom, the layout is has two components: a variable selector and a parallel coordinates visualization. The variable selector is positioned above the parallel coordinates visualization. Since each small sub-region of the parallel coordinates view is mapped directly to a variable, the user can interactively select a variable by simply choosing a sub-region. Alternatively, the user can select a variable from the control panel. Based on the selection, the corresponding information is displayed in the parallel coordinates visualization. In this visualization, time steps are indicated intuitively along x-axis. As shown in Figure 1(b), the color attributes from the Cluster View are used when rendering lines in the parallel coordinates. From this, the user is able to perform an analysis of identifying what factors cause the changes in political DPH structures.

4.3.5 Control Panels

Two control panels are designed to allow the user to manage input parameters to the visualization. The first is used to modify the attributes of the visualization. In this panel, the user is able to change variables and modify the color mapping. Since the color mapping is created by referencing the selected variable, whenever the user select a different variable in the control panel, the corresponding information will be represented to the visualization. The other panel is used for controlling the amount of contribution of a variable in the MDS calculation. The contribution change from 100% to 50% indicates that the weight change of the selected variable to 0.5. When the contribution is diminished to 0%, the selected variable will not be used in computing similarity.

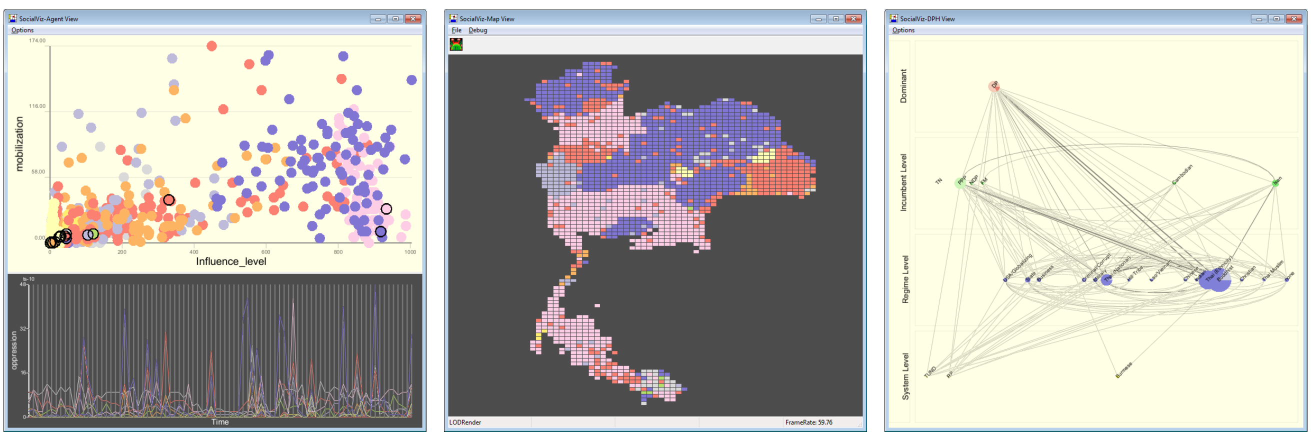
4.4 SocialViz System

To complement the MDSViz system, the SocialViz system enables analysts to perform analyses on the detailed, lower-level information of an individual simulation. In SocialViz, the analyst has access to information about the variables controlling each individual agent at every



(a) (b) (c)

Fig. 1. The MSDViz system consists of two windows and two control panels. (a) The Global View window that shows all simulations. (b) The Simulation View window that presents only those simulations selected in the Global View window. When the user select one of the simulations in the Global View window, the detailed information about the selected simulation will be displayed in the Simulation View window. (c) Two control panels allow the analyst to modify variables and colors (top), and control the amount of contribution of a dimension in the MDS calculation (bottom). In (a), a Projection View (top) shows the relationship between simulations based on a given distance function, and a Data View (bottom) displays all attributes on the simulations. In (b), the similarity of the selected simulations are computed and displayed within a Cluster View (top), and the state changes within the selected simulations are represented within a Temporal View (bottom).



(a) (b) (c)

Fig. 3. Four views of the SocialViz system. (a) A Bubblechart View (top) and a Temporal View (bottom) are designed to support correlation and temporal analysis. The Bubblechart View shows the correlation between mobilization and influence level of each group of agents. The Temporal View represents the activities of each group over time. (b) A geospatial view of the overall system including all the agents. (c) The Dynamic Political Hierarchy (DPH) view. Each group of agents is assigned a level in the hierarchy: dominant, regime, system, and anti-system.

timestep of the simulation. As shown in Figure 3, the four views (Bubblechart view, temporal view, geographical view, and DPH view) are designed to support the analysis of correlation, temporal, geographical, and DPH, respectively. All views are interconnected to support a user's interactions between different views.

4.4.1 Bubblechart View

The Bubblechart View displays the correlation between two intersecting variables. If the two variables maintain a positive correlation, the slope of the pattern of dots will be from lower left to upper right. With this approach, the user is able to examine the actions and interactions of each agent or political group by comparing the correlation between its controlling variables. The analyst can select variable to comparethrough a control panel. The color attribute is determined by referencing the activated identity within each group, utilizing the same encoding metaphor used in MDSViz.

4.4.2 Temporal View

In the Temporal View, the activities of each agent of political group over time can be represented as line, with the color of the line matched with the color of each group. The line indicates the activities of each group over time. By highlighting the line or time dimension, the corresponding information will be reflected in all other views.

4.4.3 Geographical View

Geographical information corresponding to each agent is represented in this Geographical View. Because the political simulation run in this case was performed on data gathered in Thailand, the geographical map of Thailand is used. In here, each region is mapped with an agent, whose color corresponds to the activated identity with each group.

4.4.4 DPH View

The DPH view shows the groups of agents and how their relationships impact the structure and stability of a system. The configuration of the Dynamic Political Hierarchy (DPH) characterizes the political structure of a country based on the relationships and strengths of individual political, racial, ideological, and religious groups. In this model, each group of agents is assigned a level in the hierarchy: dominant, incumbent, regime, system, and anti-system. Since agents represent states of the DPH (there are 56 possible identities), each agent can be found in 1 of 5 DPH levels. The line between groups represents the relationships, and the thickness of the line indicates how strongly the two groups are connected. By default, all linkages among groups are displayed. Since the DPH View uses a graph drawing approach, commonly known limitations (i.e. *cluttering* and *line crossing*) in graph drawing approaches also present in the DPH View. To minimize these limitations, an approach of B-spline curves is used to create a curvy line. In addition, only highlighted linkages are emphasized when the user interact with group(s).

Each group is represented as a piechart with holding two information: the number of activated identity and total number of subscribed identities. The darker region in the piechart indicates the proportional percentage to the number of activated identity.

5 Usage Scenarios

When analyzing agent-based simulation data, our collaborators in political science have identified two main goals. First, they need to be able to identify several different potential outcomes and estimate their likelihood. Second, they need to be able to identify and analyze outlier runs with high potential impact, and determine the factors that cause the simulation to behave differently during these runs than in other runs. In the following usage scenarios identified together with our collaborators in political science, we show how these goals are supported by MDSViz and SocialViz.

5.1 Identifying Potential Outcomes

The MDSVis system was initialized with data from the VirThai [1] simulation data created by our expert analysts as part of their DARPA ICEWS project. They began by representing the data with pixel-oriented glyphs of the Dominant Identity attribute in the Projection View (see Fig. 4) to explore how the simulation runs are clustered and how the clustering correlates to the Dominant Identity attribute. Because the Dominant Identity attribute has a small contribution to the distance function we are using, it can be utilized as a label for each simulation in this context.

In Figure 4, analysts observed that runs that more prominently feature Buddhist (red) or Thai Ethnic groups (light purple) as the Dominant Identity are clustered on the right side, whereas runs that more prominently feature the Red Shirts (dark purple) or Yellow Shirts (pink) are clustered on the left. Because the Buddhist/Thai Ethnic clustering is roughly the same size as the Red Shirts/Yellow Shirts clustering, the probability of Thailand's future resembling either of the two outcomes is similar.

The analysts then selected one run from each of the Dominant Identities present in the two clusters to see how the attributes of each run differ. They looked specifically at the Lobby (Fig. 5(a)), Protest (Fig. 5(b)), and Attack (Fig. 5(c)) attributes. As indicated by the graphs shown in Figure 5, there are significant differences between the two clusters for the Lobby and Protest attributes, but not the Attack attribute.

While the analysts could not make strong predictions about Thailand's future from this analysis, they hypothesized that one important distinction between the two clusters is that runs in the Buddhist/Thai Ethnic clustering exhibit high levels of legal lobbying and low levels of protest, whereas runs in the Red Shirts/Yellow Shirts clustering exhibit the opposite. To confirm their hypothesis, our collaborators then selected ten runs from each cluster and observed similar patterns for

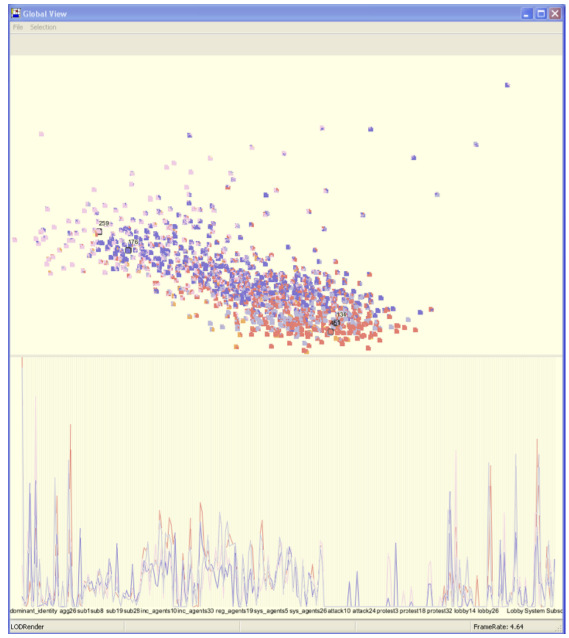


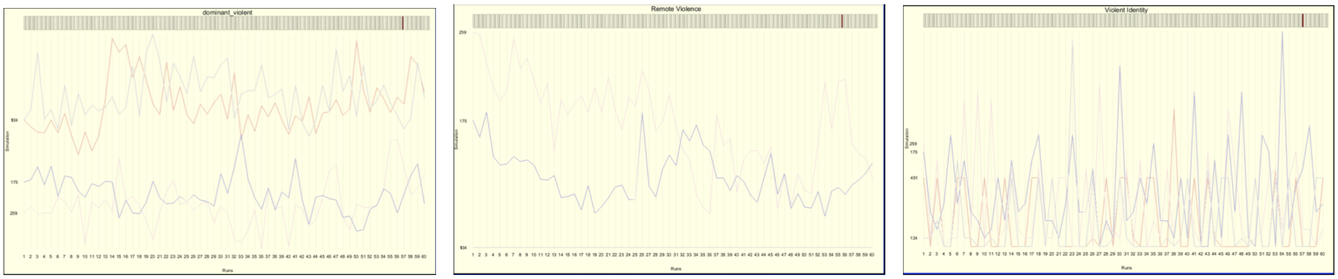
Fig. 4. A representation of the data with pixel-oriented glyphs of the Dominant Identity attribute in the Projection View. Because the Dominant Identity attribute has a small contribution to the distance function we are using, it can be used as a label for each simulation in this context. Note that runs that more prominently feature Buddhist (red) or Thai Ethnic groups (light purple) as the Dominant Identity are clustered on the right side, whereas runs that more prominently feature the Red Shirts (dark purple) or Yellow Shirts (pink) are clustered on the left.

each attribute (Lobby in Fig. 6(a), Protest in Fig. 6(b) and Attack in Fig. 6(c).)

5.2 Identifying Unlikely Yet High Impact Outcomes

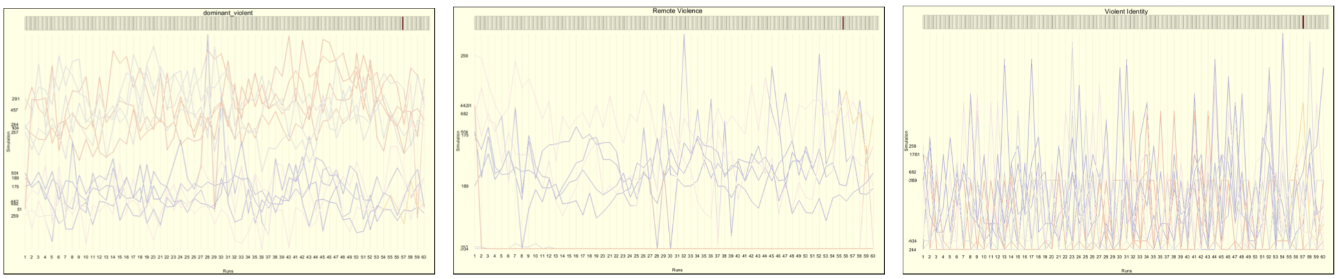
To analyze unlikely, yet potentially high impact outcomes, the analysts returned to the Projection View (Fig. 4) and focused their attention on outliers. Adding two of these outliers to the subset of runs selected in the previous scenario, analysts obtained the Temporal View of the Attack attribute shown in Figure 7. In the four runs from the "Identifying Potential Outcomes" usage scenario, there was little noticeable difference between the level of the Attack attribute for the runs, but the additional outlier runs show several spikes of very high level of Attack relative to the runs from within the clusters.

To further explore *why* these outliers display such a high level of attack, the analysts switch to using the SocialVis tool which allows them to quickly explore an individual history at an in-depth level. They begin their analysts by using the Time Series and Bubble Chart views to confirm the spikes in attach that they observed using MDSViz. To understand why the spikes occur, they then examine the DPH View of the time steps immediately preceding the increase in attacks. In this view, they observe a pattern: in the two time steps immediately before the attacks, there is a movement in the DPH level of the Thai Ethnic identity from the Regime Level (Fig. 8(a)) to the System Level (Fig. 8(b)). Additionally, there is also a shift in the Isan group, bringing them from the System Level (Fig. 8(a)) up to the Incumbent Level (Fig. 8(b)). Both of these patterns occur immediately before nearly all of the spikes in attack. From this, the analysts leverage their domain expertise to conclude that, for this run, the very high levels of violent attacks probably result from the alienation of the Thai Ethnic group whenever the Red Shirts align themselves closely with the minority Isan ethnicity.



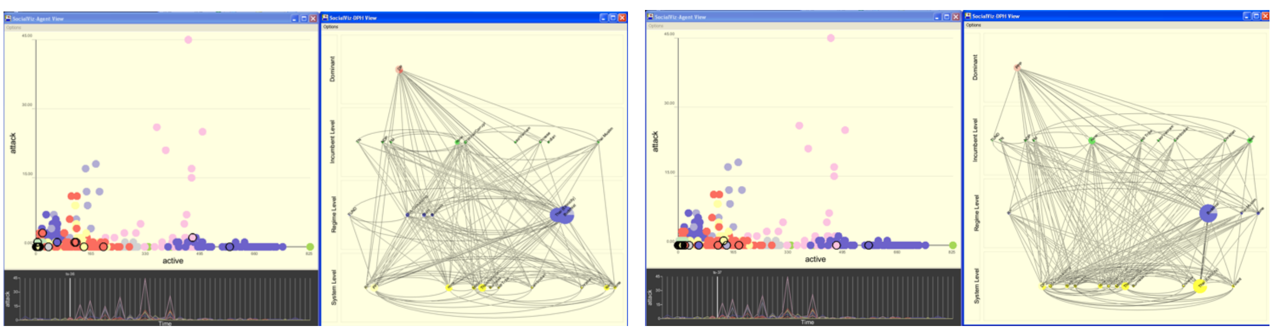
(a) (b) (c)

Fig. 5. The analysts selected one run from each of the Dominant Identities present in the two clusters to see how the following attributes differ in each run: (a) Lobby, (b) Protest, and (c) Attack. Note that there are significant differences between the two clusters for the Lobby and Protest attributes, but not the Attack attribute.



(a) (b) (c)

Fig. 6. Analysts hypothesized that one important distinction between the two clusters is that runs in the Buddhist/Thai Ethnic clustering exhibit high levels of legal lobbying and low levels of protest, whereas runs in the Red Shirts/Yellow Shirts clustering exhibit the opposite. To confirm their hypothesis, our collaborators selected ten runs from each cluster and observed similar patterns for each attribute: (a) Lobby, (b) Protest and (c) Attack.



(a) (b)

Fig. 8. To understand spikes in Attack occur in a pair of outlier runs, analysts examine the DPH View of the time steps immediately preceding the increase in attacks. In this view, they observe a pattern: in the two time steps immediately before the attacks, there is a movement in the DPH level of the Thai Ethnic identity from the Regime Level (a) to the System Level (b). Additionally, there is also a shift in the Isan group, bringing them from the System Level (a) up to the Incumbent Level (b). From this, the analysts conclude that, for this run, the very high levels of violent attacks are likely to be the result of the alienation of the Thai Ethnic group whenever the Red Shirts align themselves closely with the minority Isan ethnicity.

6 Expert Evaluation

In this section, we report the feedback from evaluations conducted with expert analysts in political science. We first introduced the analysts to the design and utilities of each system, and to the various visualizations supported in each tool. We then invited these analysts to use these systems in their analytical tasks over the course of a few days. As indicated by their detailed comments below, our analysts agreed that these systems were a highly useful tool that provided an analytical capacity that exceeds all their existing tools, and that they are robust and efficient enough to support a broader range of analyses than were previously possible.

6.1 MDSVis

Expert analysis revealed that MDSVis was overwhelmingly useful for comparing runs according to their similarity across multiple data dimensions. One analyst reported that "[t]his is the first time we've really been able to group runs according to multidimensional similarity. Until this point we didn't even really have a rudimentary strategy... and even univariate similarity comparisons relied on comparing [a] large number of time-series or comparing means." MDSVis has broadened the range of possibilities for analysis by providing a straightforward mechanism for performing multivariate clustering on complex data, as well as greatly reducing the computation time for performing traditional comparisons.

The analysts also reported that the barrier to entry to their analytical

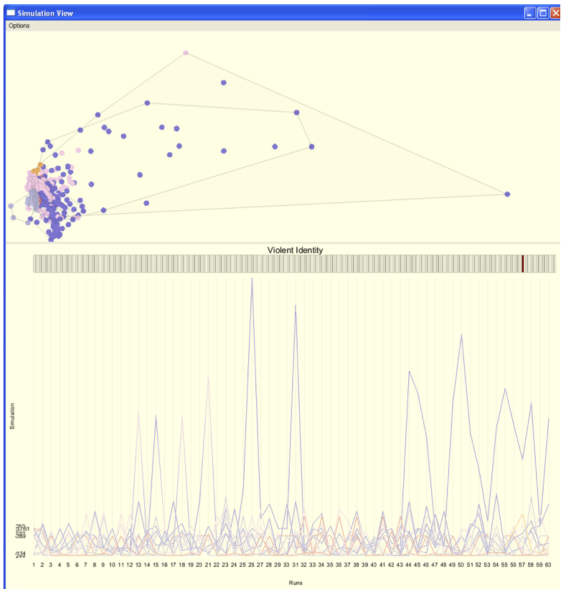


Fig. 7. To analyze unlikely, yet high impact outcomes, the analysts focused their attention on outliers. Comparing two of these outliers to more characteristic runs, analysts then used the Temporal View to explore the Attack attribute. In the characteristic runs, note that there is little difference between the level of the Attack attribute for the runs, but the outlier runs show several spikes of very high level of Attack

process would be greatly reduced by using MDSViz. They report that while identifying and grouping similar runs and then drilling down into the data to determine what makes those runs unique was possible "based on a high level of familiarity with the model... the process was often opaque." By using MDSVis to identify groups of similar runs and then utilizing the Parallel Coordinates and Time Series views to examine the details of the simulation runs, "a new user is able to explore a data set and find interesting relationships or an experienced user can more quickly understand a new data set."

During the evaluation, the experts also identified a few shortcomings of the existing system. In particular, they noted that while MDSVis is a powerful tool for analysis, it is not particularly well-suited to presentation due to the challenges in comparing across multivariate space. They also noted that while they found it useful to be able to alter their distance function by using the control panel to modify the variable weights, computation speed can be problematic. One final drawback of the multidimensional component of MDSVis that the analysts identified is that patterns across many dimensions can tend to cancel each other out. They suggested that in some cases, patterns among fewer variables might be more intuitive and show stronger relationships whereas in a data set where relationships are generally weak, this technique might help illuminate less obvious patterns.

6.2 SocialVis

Analysts agreed that SocialVis provided them with a much more efficient framework for exploring individual trajectories and different variables. One expert stated that to accomplish this task without previously they would "have to open the model in PS-I and watch the particular trajectory run or use off-the-shelf software (e.g. Excel, STATA)." The SocialVis system enabled analysts to straightforwardly access and visualize many of the variables at work in their model.

Another analyst noted that "one of the great advantages of SocialVis is its speed, which allows a user to analyze the configuration of a landscape over an entire run very quickly without having to flip back and forth between a series of images. Some of the views, like the sequential DPH visualization were not available to us at all; [before SocialVis] we only had the ability to generate the visualization from individual time steps, which is a very time-intensive process." The only drawbacks to the SocialVis system that were noted by the analysts were that not all variables and attributes within the model were available to be viewed, such as the rules and functions operating within the model.

Overall, the analysts reported that our systems are invaluable tools that met all of the design considerations that we had collectively identified at the onset of our partnership. They indicated that in many cases, both MDSVis and SociaVis would significantly streamline their analytical processes and support them in identifying interesting patterns and exploring how different factors influence political systems.

7 Discussion and Future Work

In this section, we discuss the current limitations of our system and identify areas for future research.

7.1 Identifying Appropriate Distance Functions

In our current implementation, we use Euclidean distance as a proof-of-concept distance function. However, in this distance measure, the attributes are not normalized, and thus have uneven weighting depending on the range of values for each individual attribute. While it is possible to compensate for this by adjusting the contribution for an over- or under-represented attribute in the MDSView control panel, it would be much more intuitive for analysts if equal contribution values in the control panel equate to equal contribution of attributes in the distance function. Along with this normalization, we would like to explore the utility of offering the user several initial predefined options depending on the data being examined in order to minimize the amount of time and effort required to properly tune the distance function.

Another issue with using Euclidean distance for comparing time series is that it tends to perform poorly when similar features are shifted slightly in time. This weakness is exploited especially by the agent-based simulation data used in this paper, where attributes can vary greatly between consecutive time steps. Intuitively, the distance between two runs that are identical with the exception of a slight shift in time should be almost nonexistant. However, Euclidean distance has no mechanism to recognize this.

Because of this, we have considered several other distance measures. The first alternative is dynamic time warping (DTW) [4], which can be very effective at handling such temporal shifting, but is unfortunately computationally intensive. Another alternative is symbolic aggregate approximation (SAX) [7], which can be used to determine a lower bound of Euclidean distance between two time series in a fraction of the time, and so could be applied to subsequences of the time series to quickly find similar features that are shifted in time.

In our future work, we would like to continue to explore different distance measures to afford analysts better performance and increased control when using these tools. One area of particular interest is the automatic generation of distance functions. One method currently being evaluated in our lab is the effectiveness of using a computational "best guess" approach coupled with an iterative refinement process in partnership with the user to assist the analyst in externalizing their intuitions about the data and thereby computing an appropriate, custom distance function.

7.2 Improve MDS Performance

Another area for future improvement is modifying the multidimensional scaling component to allow real-time user interaction with attribute weighting. In particular, we are interested in leveraging the work of Ingram et al. [6] on utilizing the GPU for MDS computation. In this work, Ingram et al. reported speed up factors of 10 to 15 times when using the GPU for their MDS algorithm. As noted in the expert analysis of these systems, the ability to perform multidimensional comparisons between simulations runs provides a previously unexplored opportunity for examining agent-based simulation data in close detail. However, due to the lengthy computation time, analysts are unable to iteratively refine their comparison by modifying the weight distribution across several variables and recomputing the distances between simulations. By refining and speeding up these calculations, we would enable analysts to better explore a range of hypotheses about the factors influencing sociopolitical interactions observed in their simulations.

7.3 Integrating MDSVis and SocialVis

Although both systems were very well-received, there has been some discussion about whether the functionality of both MDSVis and SocialVis should be combined into a single system. Because of the memory management issues that arise when working with such large data sets, having both systems combined into a single tool would require the system to dynamically load data, potentially resulting in diminished performance. However, the benefits to a combined system that does not required context-switching on the part of the analyst warrant further investigation into its development. This is especially true when combined with the potential for a dramatic performance increase that could be gained by leveraging the GPU for MDS computation, which would offset some of the dynamic loading bottleneck.

8 Conclusion

Analyzing and interpreting the results of agent-based models is a critical component of current research in social and political science. These simulations can help scientists to better understand the forces at work in social and political systems, which can in turn enable them to better inform decision-makers and international policy. Although there exist robust systems for developing and running these simulations, appropriate tools and support for analysis are not available, and as such it has become difficult for social scientists to interpret the results of their increasingly complex simulations. In this paper, we present two systems specifically designed to support inquiry and inference by social scientists using agent-based simulations to model political phenomena. We designed these systems in collaboration with domain experts to provide interactive exploration and domain-specific data analysis tools. Through analysis by domain experts, we validated that these systems provide an efficient framework to explore simulation data and and confirmed its novelty and utility. Based on this encouraging feedback, we plan to refine our system and make it available for use by other social and political scientists.

Acknowledgments

The results and interpretations are solely attributed to the authors. This material is based in part upon work supported by the National Science Foundation under Grant Number BCS-0904646. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. The authors also wish to thank Ian Lustick, Miguel Garces and Brandon Alcorn at Lustick Consulting. Data used in the research reported in this paper was provided with support received from the Defense Advanced Research Projects Agency (DARPA) through the Advanced Technology Laboratories wing of Lockheed-Martin in the Integrated Crisis Early Warning System (ICEWS) project (Prime Contract #FA8650-07-C-7749). The results and findings in no way represent the views of the Department of Defense or DARPA, and all results and interpretations are solely attributed to the authors. Approved for Public Release, Distribution Unlimited.

References

[1] B. Alcorn, A. Hicken, and M. Garces. VirThai: A PS-I Implemented Agent-Based Model of Thailand in 2010 as a Predictive and Analytic Tool. 2011.

[2] R. Axelrod. *The Complexity of Cooperation*, volume 159. Princeton University Press, 1997.

[3] R. Axtell, J. Epstein, J. Dean, G. Gumerman, A. Swedlund, J. Harburger, S. Chakravarty, R. Hammond, J. Parker, and M. Parker. Population growth and collapse in a multiagent model of the Kayenta Anasazi in Long House Valley. *Proceedings of the National Academy of Sciences of the United States of America*, 99(Suppl 3):7275, 2002.

[4] D. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *AAAI-94 workshop on knowledge discovery in databases*, pages 229–248, 1994.

[5] R. Bhavnani and D. Backer. Localized Ethnic Conflict and Genocide. *Journal of Conflict Resolution*, 44(3):283, 2000.

[6] S. Ingram, T. Munzner, and M. Olano. Glimmer: Multilevel MDS on the GPU. *IEEE Transactions on Visualization and Computer Graphics*, pages 249–261, 2008.

[7] J. Lin, E. Keogh, L. Wei, and S. Lonardi. Experiencing SAX: a novel symbolic representation of time series. *Data Mining and Knowledge Discovery*, 15(2):107–144, 2007.

[8] I. Lustick. PS-I: A user-friendly agent-based modeling platform for testing theories of political identity and political stability. *Journal of Artificial Societies and Social Simulation*, 5(3), 2002.

[9] I. Lustick, B. Alcorn, M. Garces, and A. Ruvinsky. From Theory to Simulation: The Dynamic Political Hierarchy in Country Virtualization Models. In *American Political Science Association (APSA) 2010 Annual Meeting*, 2010. Available at SSRN: http://ssrn.com/abstract=1642003.

[10] A. Srbljinovic, D. Penzar, P. Rodik, and K. Kardov. An agent-based model of ethnic mobilisation. *Journal of Artificial Societies and Social Simulation*, 6(1):1, 2003.