Acquiring Agent-Based Models of Conflict from Event Data

Glenn Taylor[1], Michael Quist, PhD[1], Allen Hicken, PhD[2]

[1]Soar Technology, Inc.
{glenn,quist}@soartech.com

[2]Department of Political Science
University of Michigan
ahicken@umich.edu

Building and using agent-based models is often impractical, in part due to the cost of including expensive subject matter experts (SMEs) in the development process. In this paper, we describe a method for "bootstrapping" model building to lower the cost of overall model development. The models we are interested in here capture dynamic phenomena related to international and sub-national conflict. The method of acquiring these models begins with event data drawn from news reports about a conflict region, and infers model characteristics particular to a conflict modeling framework called the Power Structure Toolkit (PSTK). We describe the toolkit and how it has been used prior to this work. We then describe the current problem of modeling conflict and the empirical data available to learn models, and extensions to the PSTK for model generation from this data. We also describe a formative evaluation of the system that compares the performance and costs of models built entirely by an SME against models built with an SME aided by the automated model generation process. Early results indicate at least equivalent prediction rates with significant savings in model generation costs.

1 Introduction

Building agent-based models is typically a costly manual process, involving subject matter experts (SMEs) as the primary source of data, and often requiring programming by software engineers or others. On the other hand, many automatically built models are based on incomplete data (e.g., only news reports), typically using statistical methods that are brittle to non-linear events. Neither approach can stand by itself as a robust solution to anticipating future conflicts. This paper describes extending an existing agent-based modeling tool called the Power Structure Toolkit (PSTK – [Taylor et al., 2008]) to automatically generate models from coded event data (from news reports), and then to supplement those auto-generated models with the ability for SMEs to tweak and tune the models based on their own expertise.

1.1 Background: Power Structure Toolkit

The Power Structure Toolkit (PSTK) is an agent-based framework for building models of power structures and their dynamics. It has been developed to represent and simulate the power dynamics among key actors in a network. At an architectural level, PSTK is grounded in theories about power and decision-making. Theories about social power structures [Bourdieu, 1986; Mann, 1986], including what constitutes sources of power and how they might be used to accomplish goals, guide the underlying representations in the PSTK. Social Exchange Theory [Foa and Foa, 1974] guides how power dynamics happen throughout a model. The decision making process of an actor – how it decides to use its resources to accomplish its goals – is an implementation of Rational Choice Theory [Allison and Zelikow, 1999] where an actor weighs a number of options and chooses the one with the highest utility.

PSTK was initially designed with the idea of allowing SMEs to build, execute, and analyze agent-based models directly, without requiring software engineers or scientists to write code. Its constructs are somewhat abstract, intend-

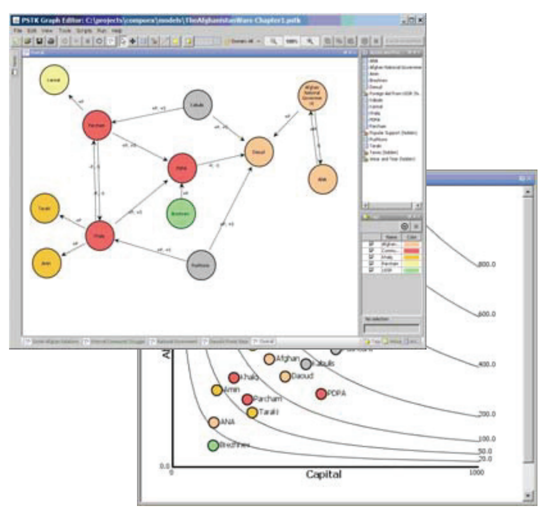

Figure 1: The Power Structure Toolkit (PSTK) allows for SMEs to build dynamic models of power struggles.

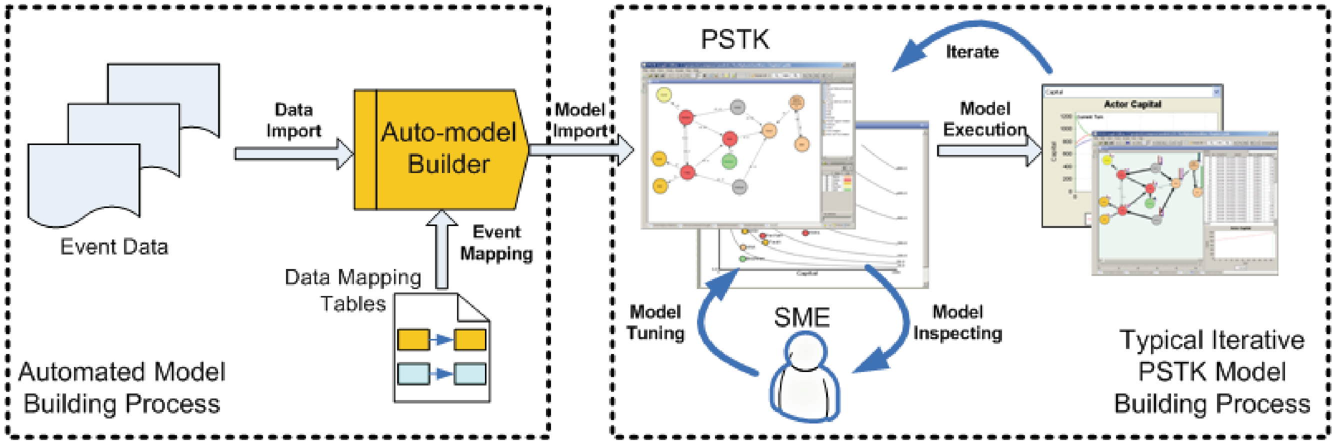

Figure 2: Bootstrapping model building to decrease costs

ing to be general, and yet at the same time simple enough such that models can be specified without traditional programming. As shown in Figure 1, the PSTK presents multiple views on a model to allow a user to specify the data needed to run a model, and to allow the user to understand how a model generated its results. PSTK is focused on actors, power and influence in different domains – how actors use their influence to help achieve their goals. PSTK actors exchange capital of a few types: political, military, economic, and social (these are configurable). Positive exchanges represent *support*; negative exchange represents *opposition*. These decisions to expend capital to support or oppose (or not to spend at all) are based on estimations of the impact of those actions with respect to an actor's goals. Where those projections indicate high payoff, the actions are taken. PSTK output includes the dynamic distribution of power across actors over time, as well as what actions were taken by the actors to drive the system at any given time, what goals were active in a given context such that those actions were taken, etc.

The PSTK has been used by SMEs to develop models of conflict areas in support of DoD experimentation. While the PSTK allows SMEs to build models, the process can still be expensive – good models require good inputs, which requires time and insight on the part of the SME. In this paper, we describe an extension to the PSTK to automatically build models based on event-coded data from news reports. In order to retain his or her expert knowledge, the SME is still involved in the process, but we hypothesize that bootstrapping the process of model building can greatly reduce the time to develop models. These additions are represented in Figure 2.

1.2 Background: Conflict Data and Estimation

There are a number of ongoing efforts in the social sciences to use data-driven methods to categorize and predict conflict around the world. One popular scale of conflict is the Heidelberg Scale [HeidelbergInstitute, 2007], which assigns a score of 0-5 for the level of conflict in a region during a given time period: 0 means no conflict, 5 means all-out war. Some methods use macro-level variables such as GNP or average life expectancy to predict conflict [O'Brien, 2002]. Often, statistical models built from historical data are used to assess the current state of conflict and make predictions about a future state of conflict; for example, [Schrodt, 1997] uses Markov models to do this kind of prediction.

One active area of research is automatically extracting and coding events from news reports, such as in the CAMEO data framework [Gerner et al., 2002] for use in conflict prediction. CAMEO data includes the date, actors, and activity/event information. For example, from a data set about conflict in the Philippines, we have the following entry:

```
991003 IGOUNO MIL 072  (Provide military aid)
```

This news event example includes the following information: on Oct 3, 1999 (991003), actor IGOUNO (the United Nations) provided military aid (event type 072) to actor MIL (the Philippines Military). These events are typically linked to the original news story, but in our case we did not have the originals, so we are limited to the extracted information. There are dozens of different conflict-related events, including all-out war to signing treaties.

The rest of this paper describes our process of acquiring agent-based models from this event data, and an evaluation of the generation process and the models generated from that process to predict levels of conflict in a country.

2 Agent Model Generation

Agent model generation includes both automatic processes and SME involvement. The SME is involved at the beginning of the process to help determine which actors (as identified in the event data) are of interest, and again at the end to supplement the generated model with information not available in the source event data. The automatic processes use the events to construct a model. As seen above, an event includes an actor pair, which implies a small social network. All such actor pairs taken in sum help form a fuller social network representing the actors in the area of interest. The event types (e.g., "Provide military aid") helps define the types of relationships between those actors. The events also give a time history, which helps provide the historical context for a model up to a given point in time. Thus, we can characterize the relationship between two actors over time.

To build an agent model from this data, we need to create a mapping from the constructs in the raw event data to constructs in the PSTK framework. For example, a key representation in PSTK is a social network comprised of actors and links between them. Actors identified in the event data map directly to actors in a PSTK model. The events also indicate a relationship between them. In PSTK, relationships are represented as Lines of Influence, which indicates generally how one actor might behave toward the other: a positive relationship implies support; a negative relationship implies opposition.

Each event type is weighted using the Goldstein scale for military conflict [Goldstein, 1992], and that weight is used to calculate the strength of that relationship and a valence (positive or negative). Given a set of many events over time, we use a weighted sum of events to capture the relationship of those actors over time, favoring those events that occur more recently (using a half-life on the events). In order to reduce the impact of noise and to generally simplify the model, we then discard the relationships whose weights are below a threshold. (We use 2 standard deviations from the mean, though this is arbitrary and can be adjusted as part of model building.) Finding optimal thresholds or ways to maintain some of these important nuances in relationships are areas for further experimentation.

The result of this process is a social network of actors and relationships that characterize the information extracted from the event data. However, this is not enough by itself to execute as a PSTK model; actors in PSTK also require goals and the resources to achieve those goals. Unfortunately, the event data does not explicitly state either of these modeling elements, so they must be either inferred indirectly from data or set to defaults. We do both in this work. Resources are specified in two ways in PSTK models: initial conditions and replenishing sources. We define defaults for the initial conditions and the rate at which actors get replenished by the sources. At the end of generation, all actors are set equivalently in this regard. (Since resource values are unavailable in the event data, this is a key area where SME insight plays a role to help tweak the generated models.)

A few kinds of goals are added in model construction. Where the derived relationship between two actors is positive in valence, we add a "supporting goal" that would allow one actor to give resources to another. Where the derived relationships is negative, we add an "attacking goal" that would allow one actor to use its resources to drive down another's. Each agent is also given a "survival goal" to try to maintain some level of influence. Finally, goals are given priorities based on the relative weights of all of their outgoing lines of influence, which impacts the proportional amount of resources dedicated to achieving each goal. Those with higher priorities get more attention in the form

Given:
1. CAMEO event file
2. Hand-generated fused actor table from full actor set
3. Selected time window of events to analyze

For each actor pair:
1. collect all events between both actors
2. assign weight/valence of event (Goldstein scale)
3. decay event value based on distance in time
4. if total weight of edge between actors is above threshold, keep the edge
5. if the link from actorA to actorB is positive, create a "support" goal; otherwise, create "attack" and "tit-for-tat" goals
6. assign default "survival" goals (keep minimum resources)
7. assign goal priorities based on the valence of the edge relative to all other edges
8. assign default initial resources to actors
9. assign support to each actor via process nodes

Figure 3: Model Generation Algorithm

of resources. Table 1 summarizes the PSTK model structs and how we fill in these constructs from the event data and other sources including defaults and SME input.

3 Experiment

In this effort, we began with a major hypothesis: An agent-based modeling approach that leverages both historical data

Table 1: PSTK Modeling constructs & Source data

Model Construct	Source Data / Method
Actors	• Event Data • Actor fusing (SME) • SME modifications
Lines of Influence (LOI)	• Event type score (Goldstein) • Time decay (parameter) • Weighted thresholds on summary score (parameter) • SME modifications
Actor Support	• Derived Network (peer support) • Default support / baseline parameter (scaling) • SME modifications
Goals / Contexts	• Deduction from Event Data: support/attack • Survival goals (default) • TFT behaviors • SME modifications
Initial resources & replenish	• Baseline parameters (equivalence/scaling) • SME modifications

(such as represented in coded news events) and the area expertise of SMEs will be superior in quality (higher prediction rates) and cost (lower SME time) compared to purely hand-built models. We will examine the two components of this hypothesis separately.

3.1 Methodology

To test these hypotheses, we conducted a formative experiment that included two SMEs: one that constructed a model by hand (SME-only case), the other which used a learned model from data (Learned+SME case). Both SMEs were political scientists with experience studying Southeast Asia. Neither SME had hands-on experience with building or editing models in PSTK, though one had prior exposure to PSTK and its concepts. To minimize this difference in exposure, we included a simple PSTK training process to try to bring them to a similar level of proficiency (though we did not have the resources to test relative proficiency going into the experiment proper). We included the tutorial time, and other times during which the SME sought help with PSTK, as part of the time components for comparison.

The test country was the Philippines, for which we had available data for the period 2001-2004, in both coded events (CAMEO format) and assessed ground truth conflict levels. The PSTK models were constructed in both cases to coincide with the end of 2003, and the data for 2004 was reserved to compare against the outputs of the model. The event data used included 41,258 events, all drawn from news sources over the course of 2001-2003. The ground truth conflict level data was given quarterly for a number of conflict types (e.g., those involving rebellion, insurgency, government repression). We concentrated on rebellion, since that was the dominant cause of conflict in the Philippines for the timeframe of interest. The ground truth data came in two forms: a maximum conflict level for the quarter (given as a discrete value from 0-5, per the Heidelberg Scale) and a binary conflict assessment (1 if any conflict occurred, 0 if not).

One issue we had to address in this work was that of data comparison. There is no established method or framework in computational social science for converting the outputs of a particular model to a form comparable to another model's outputs. Unless each model was built with comparison in mind, each model will likely produce different kinds of data, and mapping will have to be done post hoc with as much objectivity as possible. PSTK model outputs are not in the form of the 0-5 Heidelberg Index of conflict, nor are they given in a quarter-year grain size like the data we had for comparison. Instead, PSTK outputs are typically the turn-by-turn list of actions each actor takes against another (a turn is treated as a week of calendar time), in terms of the kind of action (supporting or attacking) and the size of the action (amount of resources expended). To compute the amount of conflict for a quarter, we summed the amount of conflict-related activity (total resources used in attacks) across three months of model activity. This sum becomes the quarterly level of conflict in *model* terms. We then scaled this model-conflict data to a 0-5 scale using a log-scale binning approach to better match the Heidelberg Scale. Prior to running the model, we computed the theoretical maximum amount of conflict possible in the model (based on the amount of resources available to the actors over the course of the run duration), and constructed log-scale bins to map the output of the model per time period. For example, if the max model violence capacity is 100, the bins would be (1-3), (4-10), etc. This method has the advantages of being objective and taking into account the size of the model, which can vary on a per model basis.

We established a number of metrics to compare the results of the models. First are standard machine learning metrics to measure the ability of the models to predict conflict against ground truth:

% Accuracy = # correct classifications / total # observations
% Recall = # correctly predicted conflicts / # conflicts that occurred
% Precision = # correctly predicted conflicts / # conflicts predicted to occur

One complication here is that the standard accuracy metric breaks down slightly when the comparison between classifications is not binary. That is, for a point of data that can range from 0-5, there is no partial credit given for being close but wrong, such as predicting a 1 when the actual value is 2, versus predicting a 0 – both get equal weight. The comparisons given later reflect this choice of not assigning partial credit to incorrect classifications – the strictest interpretation of the results.

To test the hypothesis regarding model building cost, we also measured the amount of time to build or tune models in each case, broken down into a few categories:

- PSTK Learning Time – time the SME spent learning PSTK (including experimenters' time to help)
- Research Time – time the SME spent researching the Philippines for 2001-2003
- Modeling Time – time the SME spend doing modeling proper (including automation configuration in Learned+SME case)

3.2 Results and Analysis

To generate model predictions, we took each of the models generated by the SMEs for the end of 2003 and ran them for a year's worth of model time (52 turns) to generate predictions of the levels of conflict for the four quarters of 2004 (so a total of 4 predictions total, each with a possible value of 0-5). In order to assess the contribution of the learned model, we also used the purely learned model (prior to the SME changing the generated model) to predict 2004. The predicted levels of conflict for 2004 were assessed for accuracy, precision, and recall for each of the three models, and those results were compared. We also compared the modeling times for the SME-only and Learned+SME cases to determine the relative time-cost of building models. The results are shown in Table 2.

For Accuracy, Precision, and Recall, the number listed is the measurement for the maximum conflict index (0-5) pre-

Table 2: Summary of modeling time and prediction comparisons for the three modeling cases.

Model	Total Modeling Time (learning+ research+modeling) (hours)	Accuracy	Precision	Recall
SME Only	56	75%	75%	75%
Learned Model	2.5 (SME config. time)	50%	67%	50%
Learned + SME	12 (including SME config. time)	75%	75%	75%
Difference (learned vs learned + SME)	N/A	+25%	+8%	+25%
Difference (SME vs learned + SME)	78.6% savings	0%	0%	0%

diction across all quarters in 2004. This data also shows that the bootstrapping process contributed an initial 50% prediction rate (accuracy) for the max intensity predictions, and adding SME knowledge to that model improved the model rates an additional 25%. The chance of randomly guessing the output correctly for each quarter was 16.6% (1 out of 6). While the rates are equivalent between the SME-only and Learned+SME cases, the actual predictions per quarter varied slightly; each case had a different error in 1 of the 4 predicted values. This result does not so far support our hypothesis that the Learned+SME model would generate better prediction rates.

However, the results show a significant savings in the cost of building a model when using the model bootstrapping process described earlier. The SME-Only time, including learning PSTK, research, and modeling time, totaled 56 hours to develop the 2003 model. The Learned+SME case required only 12 hours of the same kinds of time to achieve similar prediction rates – a time savings of >75%. Since the SMEs did not know quantitatively when their model-building efforts were "done," they relied on intuition about when to stop tuning their models. From our experience in model development, this can be highly variable between SMEs and models.

In debriefing the SMEs after the experiment, we obtained some anecdotal reasons to help explain the time difference. One effect here seems to be something like "writer's block": a learned model gives the SME something more than a blank page to start with. As one SME pointed out, the first step of identifying key actors was critical in coming quickly to a reasonable model. The SME in the Learned+SME case estimated that he would have normally taken a few days to identify the key actors in the conflict area before committing to a model; the automated model building (and the data available to us) gave that process a head start by providing a set of players to start with. Furthermore, the generated model served as an example of how to build PSTK models (in defining goals and contexts), which decreased the learning time and probably some of the modeling time in the Learned+SME case.

While we did not formally evaluate the user interface aspects of the PSTK, the comments from the SME in the Learned+SME case suggested that because the PSTK allowed the SME to inspect and modify anything about the learned model, he was able to see how the model was built and what needed to be changed to improve the model. Focused user interface evaluations will need to be done to test this hypothesis more directly.

4 Conclusions and Lessons Learned

We have developed a method for learning agent-based models from coded event data in service of improving the predictive power and the cost of using agent-based models. To our knowledge, the use of coded event data to build agent-based models is unique in the field. Others have generated agent-based models from social network data (e.g., [Carley, 2003] or have related demographic and GIS data to agent based models for conflict prediction (e.g., [Girardin and Cederman, 2007], but using coded event data as a source for generating agent-based models is new.

As part of this work, we have conducted a laboratory experiment to evaluate the efficacy of the learning process, as compared to using an SME to construct models entirely by hand. The experiment shows that equivalent levels of prediction can be achieved between an SME that starts with a learned model and an SME building a model entirely by hand, but with a >75% reduction in the time required to build a model when the SME is "bootstrapped" with a learned model. While this was a very limited experiment, the results are encouraging from the perspective of practical uses of agent-based models. Given the amount of money spent in experiments to build and maintain validated models, even achieving equivalent accuracy rates at 50% cost would be worthwhile.

From this exercise, it is clear that the content of event data (at least that which we had access to) was quite impoverished regarding the kinds of constructs typical in agent-based models. Most agent-based models deal in terms of actors, goals, beliefs, and actions – only part of which was directly coded in this data. It may be possible to extend the data coding process to include more detail, thus enriching the data available to the agent-based models. We expect that this could improve the performance of agent-based models generally, although there would also be new types of noise that would have to be dealt with.

Noise in the data was certainly a factor in the quality of the learned model. The raw data resulted in a model that was an order of magnitude larger in terms of the number of actors involved than either of the final models used, and larger still in the number of lines of influence. (Note that this raw model is not evaluated in the comparisons above.) Putting thresholds on when to construct a link between actors and when to include actors in a model during the automated process was also critical – without it, the learning process would have resulted in something close to a complete graph among close to a hundred actors. Such a large model would be too unwieldy for an SME to work with.

In some ways, the fact that PSTK is a conceptually simple modeling framework seems to play in its favor when learning models from noisy data. Perhaps as an analogue to many statistical modeling approaches that look only at a few parameters, a PSTK model has only a few types of parameters to set (though models themselves can be quite complex). Condensing the data to fit into a fairly simple modeling framework requires glossing many of the details, which reduces the potential impact of noisy data. Furthermore, that there are only a handful of parameter types in PSTK means that there are fewer gaps to fill between what is available in data and what is required for a running model.

In this work, we have also remained cognizant of the role of the human SME. The results of the experiment give credence to the idea that an SME brings new knowledge to the model that not always available in raw data. To keep the demonstrated cost benefits on the side of using SMEs, we need to ensure that the model building process generates understandable, easily modified models. If the resultant models are complex to the point of being black boxes, the SMEs will be reluctant to use them and may throw them out even if they perform well. This holds as well for the underlying modeling framework and any extensions made to it: frameworks that are too abstract or opaque, that do not fit the way SMEs think about the problem, or require so many parameters that model building is unwieldy, will be frustrating to SMEs. The complexity/usability relationship for both modeling frameworks and models is worth further research. Our approach to learning models from data was not to try to learn perfect models from the data, but to get enough out of the data to produce a model that the SME could take the last mile.

Acknowledgments

This work was funded under US Govt contract FA8650-07-C-7749. Cleared for public release; distribution is unlimited.

References

[Allison and Zelikow, 1999] Graham Allison and Philip Zelikow. Essence of Decision: Explaining the Cuban Missile Crisis, Longman.

[Bourdieu, 1986] Pierre Bourdieu. The (three) Forms of Capital. Handbook of Theory and Research in the Sociology of Education. J. G. Richardson. New York, Greenwood Press.

[Carley, 2003] Kathleen Carley. Destabilizing Terrorist Networks. 8th International Command and Control Research and Technology Symposium. Conference, National Defense War College, Washington DC.

[Foa and Foa, 1974] E.B. Foa and U.G. Foa. Resource Theory of Social Exchange. Contemporary Topics in Social Psychology. J. W. Thibaut, J. T. Spence and R. C. Carson. Morristown, N.J., General Learning Press.

[Gerner, Abu-Jabr, et al., 2002] D.J. Gerner, R. Abu-Jabr, P. Schrodt, and O. Yilmaz. Conflict and Mediation Event Observations (CAMEO). Annual Meeting of the International Studies Association, New Orleans.

[Girardin and Cederman, 2007] L Girardin and L.-E. Cederman. A Roadmap to Realistic Computational Models of Civil Wars. Advancing Social Simulation: The First World Congress. S. Takahashi, D. Sallach and J. Rouchier, Springer Japan.

[Goldstein, 1992] J Goldstein. A Conflict-Cooperation Scale for WEIS Events Data. Journal of Conflict Resolution **36**(2): 369-385.

[HeidelbergInstitute, 2007] Heidelberg Institute. Conflict Barometer 2007, Heidelberg Institute. http://www.hiik.de/en/konfliktbarometer/pdf/ConflictBarometer_2007.pdf

[Mann, 1986] Michael Mann. The Sources of Social Power, Cambridge University Press.

[O'Brien, 2002] Sean O'Brien. Anticipating the Good, the Bad, and the Ugly: An Early Warning Approach to Conflict and Instability Analysis. Journal of Conflict Resolution **46**(6): 791-811.

[Schrodt, 1997] Philip Schrodt. Early Warning of Conflict in Southern Lebanon using Hidden Markov Models. Annual Meeting of the American Political Science Association, Washington, DC, American Political Science Association.

[Taylor, Bechtel, et al., 2008] G. Taylor, R. Bechtel, K. Knudsen, E. Waltz, and J. White. PSTK: A Toolkit for Modeling Dynamic Power Structures. BRIMS, Providence, RI, SISO.