

Creating Realistic, Scenario-Based Synthetic Data for Test and Evaluation of Information Analytics Software

Mark A. Whiting
Pacific Northwest National Laboratory
PO Box 999
Richland, Washington 99352
509-375-2237
mark.a.whiting@pnl.gov

Jereme Haack
Pacific Northwest National Laboratory
PO Box 999
Richland, Washington 99352
509-375-6350
jereme.haack@pnl.gov

Carrie Varley
Pacific Northwest National Laboratory
PO Box 999
Richland, Washington 99352
509-375-2814
carrie.varley@pnl.gov

ABSTRACT

We describe the Threat Stream Generator, a method and a toolset for creating realistic, synthetic test data for information analytics applications. Finding or creating useful test data sets is difficult for a team focused on creating solutions to information analysis problems. First, real data that might be considered good for testing analytic applications may not be available or may be classified. In the latter case, tool builders will not have the clearances needed to use, or even see, the data. Second, analysts' time is scarce and obtaining the needed characteristics of real data from them to create a test data set is difficult. Finally, generating good test data is challenging. Commercial data generators are focused on large database testing, not information analytics tool testing. Our distinctive contribution is that we embed known ground truth in a test data set, so that tool developers and others will be able to determine the effectiveness of their software and how they are progressing in their support for information analysts. Our automated methods also significantly decrease data set development time. We review our approach to scenario development, threat insertion strategies, data set development, and data set evaluation. We also discuss our recent successes in using our data in open analytic competitions.

Categories and Subject Descriptors

H5.1. [Information interfaces and presentation]: Multimedia Information Systems – *evaluation /methodology*

General Terms

Verification.

Keywords

Evaluation, data generator, visual analytics, information visualization

(c) 2008 Association for Computing Machinery. ACM acknowledges that this contribution was authored or co-authored by a contractor affiliate or employee of the U.S. Government. As such, the Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only.

BELIV'08, April 5, 2008, Florence, Italy.
Copyright 2008 ACM 978-1-60558-016-6/08/0004...\$5.00.

1. INTRODUCTION

Can newly developed information analytics technology detect a terrorism threat buried in a multi-source, composite data stream? The National Visualization and Analytics Center (NVAC) Threat Stream Generator (TSG) project team has developed a new approach to creating realistic test data sets and has created a series of data sets that are being used by advanced commercial analytic tool developers, government information analysts and technology developers at universities for tool evaluation. The distinct advantage to TSG data sets is that known ground truth in the form of a pre-determined threat, such as terrorist activity or a law enforcement concern, is created and translated into data cues. The cues are inserted into the data set with a known expressivity—the number of cues and their subtlety of representation are controlled.

2. MOTIVATION

The goal of the NVAC is to advance the state of the art in Visual Analytics. To this end, NVAC has published an R&D agenda, called *Illuminating the Path: The Research and Development Agenda for Visual Analytics* [14]. In Chapter 6, the agenda presents the challenges faced in the evaluation of moving the R&D advances into practice, as well as recommendations for supporting the critical role evaluation takes in advancing visual analytics. A key recommendation is to create an infrastructure for evaluation that includes “sample data sets, tasks, scenarios, benchmarks, and metrics.” As part of the NVAC research portfolio, the Threat Stream Generator (TSG) project was formed. Initially, our scope was to find a way to create test data sets for visual analytics applications that contained known ground truth, with respect to a scenario and one or more threats. In other words, we were directed to embed data within a data set that resembled actual data being used by information analysts. The embedded data should be discoverable by an analyst looking at the data using a visual analytic tool. The embedded threat data should be expressed in a way that the analyst would identify it as a threat.

Data with known ground truth brings previously unexplored value to evaluation processes. Having known ground truth in a data set provides a baseline for evaluating how an analytic tool supports an information analyst in their job. Synthetic data has suffered from the criticism that it is unrealistic due to the typical approach of using random data that looks like real data, but lacks the characteristics and substance of real data. Using data that is real, but previously unanalyzed, requires analytic results to be

evaluated using qualitative rather than quantitative approaches. Using data with results that are known to information analysts *a priori* also may not adequately help evaluate usefulness of an analytical tool.

The literature on synthetic data generation includes approaches to speedily creating large data [5] and creating data with specific content and statistical characteristics [14]. Considering recent literature on evaluation of human-information interaction and metrics for assessing how well information analysts interact with new systems, we see work in domains such as intelligence analysis [11],[12] and technologies such as information visualization [1],[9]. Our work fits well in facilitating evaluation of tools with respect to both domain and technologies, particularly with respect to insight-based evaluation. We provide a new approach for the data generation methods and processes themselves.

3. TSG PROCESS

3.1 Methodology

There are several phases involved in our method for creating TSG data sets. We begin with a determination of which aspects of the information analytics process are to be tested. We follow this with scenario development, which is the creation of a believable story from which data will be generated. Included in this task is threat creation, which follows particular guidelines provided to us by information analysts. We then decide the type of data set to be created, for example, all text or a heterogeneous mix of data types. We refer back to the scenario and threat specification to determine what cues are to be expressed in the data and how they will be realized. For example, a cue may end up as a suspicious set of numbers in a spreadsheet or as a statement made by a character reported in a newspaper article. Finally, we generate and review the data set through a combination of automated and manual processes.

Each step of the TSG data set development process is evaluated by information scientists, information analysts, and domain experts. Step-by-step testing is necessary to maintain the validity of the scenario expression we seek to achieve in the data. Our approach is similar to that of writing a mystery novel, but our whodunit is expressed in raw data as opposed to prose (although prose may constitute a considerable part of the overall data set we create).

3.2 Aligning the Analytic Challenge and TSG Data Set

TSG data sets are created to aid information analytics tool builders in assessing progress in developing software products. We quickly realized the relationship among several components of the information analysis process that needed to be considered in the development of a successful test data set for this purpose.

Figure 1 illustrates these components and their relationships. It might be expected that the TSG data generation would only be applicable to the “Data” component. However, when defining a scenario and tasks for the user to achieve with the data, obviously the “Tasks” and the “Goals” components are involved. We define the Tasks component to represent the job that the information analyst has to perform on the TSG data, for example, “Find the threat!” We define the Goals component as relating to the

product to be generated as a result of performing the Task. For example, a PowerPoint slide show describing the analyst’s approach and results may be the specific goal. Looking even deeper into the work to be performed with a TSG data set, the People to be involved must be considered. Analyses performed by university students should not be expected to demonstrate the same degree of analytical skill as those performed by professional analysts. Also, university students would typically not employ formal methods or Processes, unless they have had special training. Finally, when considering developing a data set to test Tools, the forms of data and the types of tasks to be performed must be considered. A tool that visualizes large corpora of text documents may not be able to also analyze large numeric data sets. It is not a large leap to understand how each of these components interacts with each other. The connectivity of the components may also be used to our advantage in creating data sets. For example, if we wanted to encourage collaboration among people working on our data sets, we may create a data set sufficiently diverse so that a team with one particular tool must collaborate with another team with a tool that can analyze data the original team could not.

3.3 Scenario Development and Threat Specification

For TSG purposes, we define a scenario as a story – one or more plot lines that take place over a finite time period. Characters, situations, and places are all important to the scenario. A threat is some menace to national security or a challenge for law enforcement. A scenario is often considered as a chain of events, and the scenarios for TSG data sets reflect this. A simplified example might be: “A terrorist group in Japan had been active for several years, but major leaders were captured and trials have been ongoing. Several of the leaders have been recently given harsh penalties from Japanese courts. The group has reformed and acquired additional resources and has been angered by the court decisions. They decide to strike back using terrorist methods. They perform trial runs of attacks at Disneyland Tokyo, as well as executing targeted assassinations of high level judges involved in the trials.”



Figure 1 TSG Analytical Components

One decision to make is whether to base an invented scenario on historical fact, that is, altering reality in some way, or to create the scenario from scratch. If we had put “Aum Shinriko” in place of “terrorist group” in the example above, we would be creating a scenario involving changed history, since this scenario never actually occurred. If we made up the terrorist group, judges, and other people and events, we could be creating a scenario from scratch. In both cases, care must be taken in making the scenario realistic, or establishing suspension of disbelief for the information analyst who will review the data. This is the same process mystery writers must attend to when developing the set, setting, and action of their stories. Once, when we had developed a “changed history” scenario, one information analyst reviewer told us that if their area of expertise had been in the domain of the scenario, identifying the threat would have been no challenge at all to them to discover and our data set would have failed.

Important information to be captured in a threat specification is who, what, when, where, why, and how. The aggregation of these factors in a threat specification must be sufficiently important to catch an information analyst’s attention. For example, a planned attack to topple Seattle’s Space Needle would be of more interest than a plot to dig up a rural community’s neighborhood garden. We try to informally assess these factors with respect to *consequences* of a successful action, although in-depth discussion of such assessment is beyond the scope of this paper. We have interacted with other groups performing threat risk assessment, and we plan to obtain formal assessment of TSG data sets in the future.

3.4 Data Set Composition

Once we understand the scenario, the threat to be embedded, and the goals of the challenge that the TSG data set will create, we can create a model of our synthetic data. We make an initial determination of the form of the data that we would like to express the threats within, the form of the data environment within which to couch the threat data, and some initial characteristics about the data set, such as size.

4. TSG DATA GENERATION

Our method of generating data set comprises a stepwise approach where domain specialists, information analysts, are involved in and evaluate the results of each step.

Figure 2 shows the process of creating a data set. Ovals represent

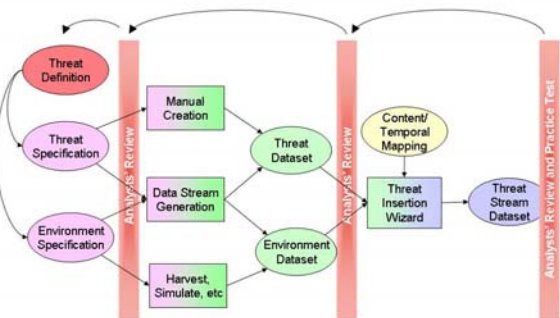


Figure 2 Data Set Generation Process

products of steps; rectangles are processes. As we view this diagram from left to right, we begin with the scenario and threat definition, as described above. The specification steps for the threat and the data environment describe the kind of data set we intend to generate. The scenario, threat, and data set concept are reviewed by information analysts, who have domain expertise in the area that the scenario describes. Reasons for rejecting a scenario may include insufficient believability, excessive complexity, or lack of scenario significance (i.e., why would anyone care about this?).

As we mentioned earlier, one of the ways our approach is different from writing a mystery novel is that we take elements of a scenario and express them in data. To support this process, we have created an automated tool called the Threat Definition Matrix (TDM). The TDM (Figure 3) is a wizard-like tool that allows us to fill in cells of a spreadsheet with a stepwise description of how we transform cues into data elements. We start by expressing the cue from the scenario that we wish to define in data. We can then map from the element into a data type (for example, a text passage, a picture, or a set of numeric values). We precisely specify the data element to be expressed and the environment data within which it is to reside. We provide an index to the data element so that we can find the cue in the finalized TSG data set. Finally, we assign a subtlety estimate to the expressed cue, to better understand how we are formulating our data set overall. At this point, our subtlety estimates are subjective, and reviewed by analysts prior to data set release.

Data to be included either in the threat component or the surrounding environment component may be created by hand, generated with our TSG toolset, or imported from another source, e.g., harvested from the web. Once the threat data and the environment data are obtained or created, we review them once again with information analysts. Generating or obtaining appropriate environment data is critical to the success of a TSG data set. Some people describe this data as “noise,” and indeed, if you were to consider the threat and environment data as “signal” and “noise,” this characterization may be appropriate. However, the environment data will host the threat data and must be carefully selected and engineered as part of the overall data set. Its characteristics must be well understood in order to properly achieve integration with threat data. One characteristic of

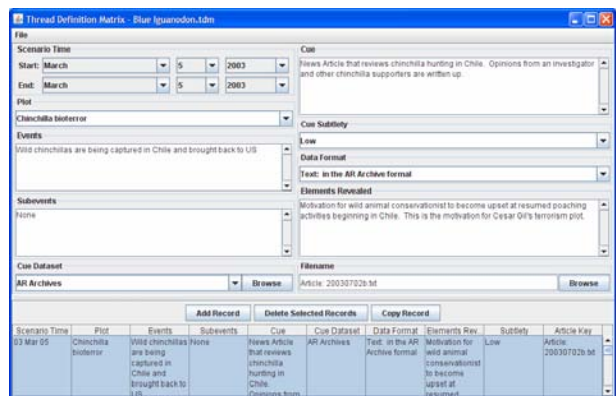


Figure 3 Threat Definition Matrix Wizard

importance to environment data is its ability to be mimicked. If we cannot adequately model threat data to resemble the surrounding data, we will not be believable. Another characteristic is the environment data set's malleability. It must be sufficiently flexible to accommodate insertion of the threat data. Images, for example, are difficult for non-professional image artists to manipulate – hiding a cue in an image can be difficult. Mimicking data has become a specialty for the TSG team. Imitating header and footer information, indexing codes, writing styles and numerical formats, and structures and formats are all aspects of this activity.

Information analysts are asked to pay special attention to the threat data at this point to decide if this data resembles information they would encounter in their everyday jobs. Evaluation criteria for them include data size, data format, presentation of content, and match of the cue information with the form of expression. This last criterion looks at whether a cue is being naturally expressed, from the point of view of their profession, in a data format. A major task for the information analysts at this point in the evaluation of the threat data is whether or not an analyst would be able to assemble a description of the threat from the threat data presented. In other words, did the TSG team do its job in expressing the threat in data, similarly to how information analysts would find it in their everyday jobs?

Following this review, we intersperse the threat data with the surrounding environment data. If the threat data is a coherent chunk, then this activity may be easy. If the data must be blended into some of the environment data, for example, news articles placed within a corpus of other news articles, then the task becomes more challenging. It becomes important to pay close attention to the grouping characteristics of the environment data, for example, temporal ordering, when inserting threat data. We have had some unfortunate, but sometimes amusing, situations when we have confused temporal ordering of data in a TSG data set. That form of construction error can confuse an entire plot line, and subsequently discredit a data set. So, threat data must be carefully evaluated with respect to the environment data at this point. In particular, we wish to avoid data that “sticks out like a sore thumb”, due to formatting errors, composition mismatches, or content mismatches. For example, if we insert newspaper articles we have written into a set of existing newspaper articles, our news articles must look like the others, read like the other and inconspicuously blend with the others.

At this point, our assembled data set is ready for review, and this represents the final bench test for assessing the data set before it is used externally. We provide the data set to analysts, who have access to tools we are interested in testing, and ask them to find the threat. Other instructions may be provided at this time, including time limits, guidelines on interacting with other analysts, and form of their results. Once a tester is finished reviewing the data, the TSG team will debrief them. We review their results from performing the specified tasks, and then ask them about the task experience. We are interested in quantitative results, such as the who, what, where, when, how, and why of the threat, but also insight-based, qualitative results. Some of our questions include:

- Was it easy or hard to find the threat?
- Was it easy or hard to use your tool(s) in helping you find the threat?

- Would you have liked other tools to help you in your task? What kind of tools?
- Do you think you found all of the embedded threats in this data?
- How confident are you in your solution?
- Were there any gaps in the data?
- If you were in a position to do so, are there any actions you would recommend as follow on activities to either support law enforcement activities or to prevent terrorism activities?
- Did you have enough time to find the threat? If not, how much more time do you think you would need?
- Which was the most difficult aspect of the task you encountered? Which was easiest?
- What other human or automated support would you have liked to help you find the threat?

Note that while these questions will help evaluate a task and the activities involved in performing a task, we ask these questions from the perspective of data developers. As we have shown above, there are several cooperating components of developing the data set, but for these evaluation exercises, we are particularly concerned with creating a high-quality test data set. So, adjustments that we make following these tests will look at the data set as a first priority.

5. TSG GENERATOR SOFTWARE

As mentioned earlier, it is essential to create tools to support the synthetic data generation process. We began our experiments in creating data sets by manually producing threat data and inserting this data within a data environment of similar form. This was the approach we used for Trilobite, our first TSG data set¹. For Trilobite, we wished to test visual analytic tools that could process and support analysis of large amounts of text data. Two of these tools developed at Pacific Northwest National Laboratory are IN-SPIRE [18] and Starlight [10]. We knew that these tools could support analyses of thousands of short documents, so we targeted a data set size of about 1200 documents. We harvested test data from a news service and limited results by restricting topics and the date range. The articles turned out to be about three-quarters of a page in length, with a particular header, footer, and writing styles associated with various sources. We hand wrote several text articles that revealed cues about our invented threat and embedded them in the data set. This turned out to be a laborious but very informative exercise. For example, we learned the importance of cue subtlety. When we generated our first version of this data set, the analysts were not able to find our threat within the data, because it was overly subtle (i.e., either an insufficient threat or insufficient information for detection). In this case, the text articles were insufficiently defined as a topic that the visual analytics tools would allow them to be found in a reasonable amount of time. After tuning the data set, our next iteration of evaluation was much more satisfying. Five of our six

¹ All TSG datasets are code-named after a prehistoric creature, and “Trilobite” seemed appropriate for a first effort.

test evaluators found the embedded threat. None of the evaluators tagged our threat as “the fake threat,” that is, it was not called out as obviously artificial. We believe we accomplished two important goals: first, we were able to make our threat discernable within a blended data environment. Second, we were able to adequately mimic the data environment with our threat data, in a subjective sense, so that our data did not appear synthetic.

As encouraging as our initial efforts were, we realized from early on that automated help would be needed to produce synthetic data sets more quickly and efficiently. We began by producing a modest software tool and have since grown a much more sophisticated system. We also realized from the beginning that some desirable automated tasks would not be achievable, such as free-text generation, so we have developed substitutes and workarounds, until such technology is more advanced.

5.1 Data Generation Background

Current state of the art in data generation focuses on creation of data sets to test applications and databases, typically in a business environment. We reviewed several of these tools before embarking on building our own. An example of an application-oriented test data generator is GS-Apps [8]. GS-Apps looks to be one of the leaders in enterprise-level data and application environments and provides a flexible tool for creating data that can be used for load testing on databases and for system integration testing among other uses. The tool does not include functions to define or embed data with context in what is created. Another toolset of great interest to our team is ToXGene [2] out of the University of Toronto, a template-based tool that uses a declarative approach to generate synthetic data in XML format. We used ToXGene for early experiments in development of the TSG software, as we agreed with the ToXGene thinking recognizing the utility of the declarative approach for generating several different data sets from a single template or simple modifications of that template.

As attractive as the use of existing software was to us, we realized that we needed an extreme amount of flexibility in the tool we were to use to generate data, which eliminated commercial software we could not modify. ToXGene posed problems for us in the way in which it generated certain types of basic data types. In addition, our ultimate plans call for the development of a

Threat Stream Generator language, with which we can drive the automated creation of test data based on a scenario.

5.2 Core Software

Our core generation software is written in Java and runs on commodity computers. However, it is capable of generating large amounts of data. For example, in one version of our Spinosaurus data set, we have created test data sets of ten million multivariate data records. Specifications for data generation are encoded in templates, so that schemes may be created, saved and modified with ease. The data creator interacts with the Generator software through a user interface that facilitates definition and easy testing of their generation rules (Figure 4).

The Generator functions in three modes: generation according to statistics, according to rules, and according to semantic specification. Some of the Generator’s basic functions include ability to generate random numbers using various statistical distributions, including normal, chi-square, Poisson, student, and uniform. It creates dates in several different formats. Alphanumeric strings can be created with great flexibility using a regular expression syntax for specification.

Lists of values can be created using rule definitions. The user can ask for values to be drawn from a file filled with data of arbitrary type in a random selection or unique selection without replacement. Rules can also be specified so that generated dates may be dependent upon each other, and can be quite sophisticated in possibly unexpected ways. For example, rules may be created that ensures date of birth precedes date of voter registration that comes on or after the person’s eighteenth birthday, and this precedes date of death, if this exists.

Although the Generator cannot create free text, we have sophisticated templating and text replacement capabilities. Text replacement templates can be defined that allows a text structure to be populated with values from other fields in the overall template, or with information from other templates, or by invoking an information harvester to draw text from another source. This approach has been useful in creating incident reports concerning names and information drawn from other records.

Data entities may need to be dependent upon other entities and make semantic sense. For example, if we are creating fields representing an Event in a record, and the Event requires a Type, Subtype, and a Name, then we can easily create a Type such as “Public Event” and a “Subtype” such as “Sporting Event” with simple rule selections. However, we need to be a little more intelligent in matching the Name appropriately to that subtype of Sporting Event. We can create ontology-based attributes for information, so that we know that a file of names such as “Seattle Mariners baseball game” is a sporting event, and use this information to appropriately populate a field.

5.3 Utility Software

In addition to the core generator, we have created an increasing number of supporting utilities to help refine data sets. One utility is the Timeline Tool. As mentioned, maintaining integrity in the temporal aspects of the threat data is extremely important in bolstering analyst confidence in the data.

Figure 5 shows a screenshot from the Timeline Editor where documents are aligned on the timeline above. Documents may be selected and reviewed as seen in the bottom left panel. Searching

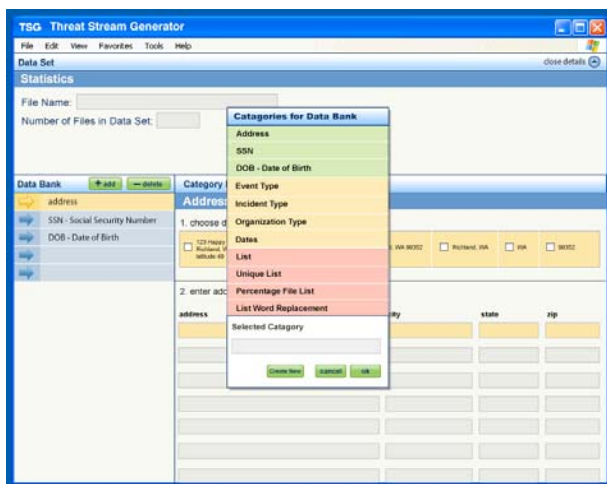


Figure 4 TSG Core Software

and editing may be performed in the other panels. Most importantly, the timeline gives a visual depiction of the arrangement of data in a data set, and allows the data set developer to easily move one or more data elements around on the timeline to best fit the scenario, and to search and find documents of temporal interest. Documents are tagged when they are moved so that all dates may be appropriately modified. There currently exists limited functionality to automatically update dates when a document is temporally re-arranged.

Another utility used extensively by the TSG team is the Universal Parsing Agent (UPA) [17]. Our key use for UPA is to anonymize data that we have harvested or imported from other sources. UPA uses a template-based approach that allows us to easily establish rules for selection and replacement of real people, places, and other proper names with invented names, and run the process on large amounts of data.

6. DATA SET DESCRIPTIONS

We have a diverse set of TSG data sets that have each contributed to our knowledge of synthetic data set generation. Trilobite, as described above, has about 1200 text articles containing an invented terrorism threat. Triceratops and Stegosaurus are two heterogeneous data sets used in the 2006 Visual Analytics Science and Technology (VAST) symposium contest, containing plot lines concerned with political intrigue and chemical weapon production. Blue Iguanodon and White Smilodon are two heterogeneous data sets created for the VAST 2007 contest, featuring exotic animal smuggling and bioterrorism subplots. Pterodactyl holds about 100 pages of chat room-style data, developed for analytical tools that can analyze deception in text data. Allosaurus consists of 90,000 web documents with Hurricane Katrina and energy issues as its primary topics. Spinosaurus contains multi-variate data describing fictitious U.S. border crossings with possible suspicious activities.

7. EVALUATING SYNTHETIC DATA SETS

One effective approach to evaluate a tool's effectiveness, as well as stress test a synthetic data set is to hold a contest, open to international participation. We put two TSG data sets to the test at the 2006 VAST symposium [6] and our complete data set may be accessed at the contest website [16]. The challenge is issued to visual analytic tool builders (not professional analysts) to test

their software against the TSG data set to find the threats in the data. We presented potential contestants with a task, a data set, and instructions on how to create an analytical product. The effort of creating this contest involved developing the scenario and data set, developing the task, advertising the contest, creating initial metrics for the results, assembling a judging panel, judging entry and selecting winners, and finally, putting on an exhibition live event for the winners and invited information analysts during the symposium.

7.1 Contest Background

As we initiated the 2006 contest, we realized that even with the experiences of preceding, related contests, we would have to create evaluation measures for the visual analytics tools based on their contest entries. We drew from work in information visualization [1],[9] and began drawing from experiences in intelligence analysis [4]. For the contest, we were interested in contestants' use of their visual tools to help them attain insight and analyze the threat embedded in the data. Contests have been successfully used in other venues to achieve similar goals [3],[7],[11].

7.2 Contest Data Set

We used a previously developed data set, Triceratops, for the VAST contest. The scenario involved a political scandal, and possible bioterrorism issues, set in the imaginary city of Alderwood, Washington. The situation presented was that a previous investigator had gotten to a certain point in their work but had been called off on a priority assignment. The job of concluding the investigation and determining if something was up in Alderwood was assigned to the contestant. The threat was distributed across several different kinds of data. The data sets included about 1200 articles from the fictitious Alderwood Daily News, a 50,000 record Excel database, some photos, some maps, and Word documents containing background information (Figure 6).

The data set was released in February 2006 and submissions were due in July 2006. When we initially developed Triceratops, we had not planned on a 5 month examination period. However, since we knew no one would be spending full time on a contest, the timing seemed to work out.

We advertised the contest on numerous IEEE and ACM sites as well as KDD and European visualization and graphics

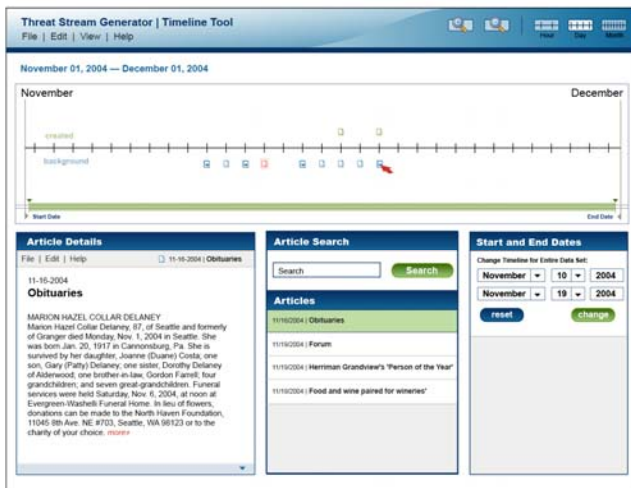


Figure 5 Timeline Editor

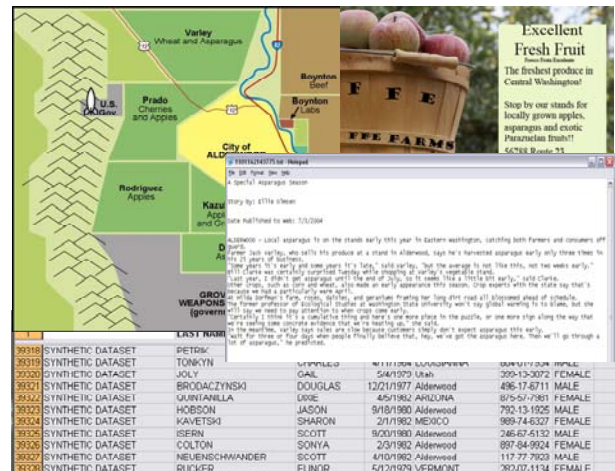


Figure 6 Data Montage from VAST 2006 Contest

newsgroups. The contest team also made extensive use of personal emails, announcements at other conferences and associated meetings, and related mailing lists.

7.3 Judging

Our judging teams consisted of information scientists, senior information analysts, TSG team members, and visualization experts. Submissions were scored on the basis of the clarity and reasonableness of the explanations given, an illustrative video, the correctness of responses to specific questions we provided, and the specific evidence provided. The judging measures were a combination of quantitative and qualitative scores. Identification of who, what, where, and when earned points for an entry. False positives resulted in points deducted. Judges provided a qualitative score for the written debrief provided by each team, with respect to how well it captured ground truth. Visualization features of the analysis environment were rated. Components evaluated include overall utility, interaction quality, layout, use of color, and clarity of symbols and labels. Other features reviewed included toolset scalability, versatility, handling of missing data, uncertainty, and collaboration support.

Since we had contest team members on both coasts, we held two judging sessions, and then reconciled results by telephone conference call. Each team of judges summarized their judgments and suggested awards.

The three teams were pronounced winners and invited to participate in a special interactive contest before the conference during which senior information analysts were paired with the teams to review their tools and provide advice from a user's point of view (Figure 7). The TSG team developed a completely new data set, similar in form and format to the original (this data set was based in Alderwood, Washington, also, to help provide a small amount of continuity) that was used for this session, called Stegosaurus. Stegosaurus was designed to be analyzed in two to three hours, using visual analytics tools. Our lab testing sessions were quite valuable in helping us tune the data for this phase of the competition. Small teams of tool builders and information analysts dissected the new problem after a training session for the analyst. The correctness of their answers and the number of



Figure 7 Analysts and Researchers Examining a TSG Dataset at the 2006 VAST Symposium

subplots they were able to locate were reviewed at the end of the session, although this session was held as a learning activity for both tool developers and analysts. The tool developers drove the application to eliminate the analysts' having to struggle with details of how to operate the software.

During the live session, contest team members observed the interactions, and our contest team members from NIST provided support to videotape the session. These three winners were recognized later in the symposium at a special awards presentation, and they also presented their work at the poster session. Their materials are posted on the contest website [16].

7.4 Assessment of the Contest and Data Sets

The response to the contest at the symposium was very positive. Analysts, researchers and tool developers were strongly supportive of this approach to assessing progress in tool development.

At the time of this writing, the 2008 contest is underway. We plan on continuing this contest and hopefully holding other contests in different venues to examine this approach to evaluation. We believe it is valuable to have both the long term contest, as well as a live one. The 2007 contest data set and task were similar to the 2006 work, so to provide some familiarity and encouragement to hesitant potential contestants. The 2008 contest has been slightly modified, so that contestants can compete in either mini-challenges, smaller tasks focusing on a particular analytic problem, such as social networking, or contestants can compete in a comprehensive, multi-task problem, requiring sensemaking across datasets. The contest team will continue their work on metrics, which will hopefully be published in the future. A key area of interest is the merging of qualitative and quantitative assessments, as we have realized the value in both. Another interest area is the degree to which these contests can encourage collaboration, which is also of interest to organizations supporting information analysis.

7.5 Lessons Learned

There are several lessons learned that have been mentioned in this paper that may help others create realistic, synthetic data for test and evaluation. To review some of these:

- Develop your first data sets by hand. You will obtain an appreciation of both the process of creation and what tools you will need to augment the process.
- Involve information analysts at the start of your processes. This will provide you with invaluable help in creating believable data sets and also provide credibility when other information analysts review the data.
- Decide which information analysis components you will focus on in your evaluation: tools, people, tasks, processes, or results.
- Consider that design phases of data set creation may take longer than the generation phases. Complex scenarios often result in very complex data sets.
- Do not dismiss data within which your threat data is to be embedded as simply "noise". It provides the basis

for shaping the look, feel, and context for what you place in it.

- Attend to syntax and semantics details of your data. Check and recheck them. You can be sure that if you inadvertently name someone “Mercutio Navarro” in a text document, and “Navarro Mercutio” in a related spreadsheet, your testers will uncover this.
- Evaluate your data and processes, in addition to evaluating analysis tools. Plan for time to incorporate lessons learned into your generated data set to improve it for the next evaluation.

Other lessons may depend on the specifics of the data set being created, for example, primarily numeric data expresses a story differently than primarily textual data.

8. CONCLUSION

We have presented the NVAC Threat Stream Generator project’s method and toolset for creating realistic, synthetic data sets for the testing and evaluation of information analysis software. We believe we are helping to evolve evaluation processes by creating data with known ground truth, where we create a scenario and map elements of a threat from the scenario to the data. We are testing our results in contests where tool builders and information analysts try to “find the threat” in the data. Feedback from the contests has been very valuable to help us evolve our processes for data creation. Tool builders report that the contest is helping them to develop better software, and information analysts’ feedback has also been encouraging.

Our data generator toolset is critical to our processes of data generation. The toolset has strengths in generating multivariate data using statistical, rule-based, and semantic approaches, and provides significant speed-up in iterations of data set creation. It is also handy for generating different versions of the same data set that incorporate data set designer changes.

We have several challenges in continuing this work. Certain types of data generation are active research areas, for example, free text and images. We plan to blend advances in these areas into our tool when possible. We also plan to continue automation of the TSG processes. We are currently working on feeding TDM specifications into the core data generator. Also, we would like to examine how TSG data sets fare when analyzed by automated threat risk software.

One main criticism of synthetic data for testing has been the lack of realism, and subsequently, believability by users. Although some of this criticism may never be overcome, we attempt to address these feelings by using believable scenarios and threats and paying particular attention to those features of data that users need to “buy into” a data set. Known ground truth is a particularly compelling feature when running tests.

9. ACKNOWLEDGMENTS

We thank the Department of Homeland Security for supporting this research through the National Visualization and Analytics Center located at Pacific Northwest National Laboratory.

10. REFERENCES

- [1] Amar, R. and Stasko, J. A Knowledge Task-Based Framework for Design and Evaluation of Information Visualizations. In *Infovis'04*, IEEE Computer Society Press, (2004), 143-150.
- [2] Barbosa, D., Mendelzon, A., Keenleyside, J., and Lyons, K. ToXgene: a template-based data generator for XML. In *Fifth International Workshop on the Web and Databases*, (2002).
- [3] Chinchor, N., and Hirschman, L. Evaluating message understanding systems: an analysis of the third message understanding conference (MUC-3), *Computational Linguistics* (1993), 409 – 449.
- [4] Cooper, J. *Curing Analytic Pathologies*, Center for the Study of Intelligence, (2005).
- [5] Gray, J., Sundaresan, P., Englert, S., Baclawski, K., and Weinberger, P. J. Quickly generating billion-record synthetic databases. In *Proc SIGMOD*, ACM Press, (1994), 243-252.
- [6] Grinstein, G., O’Connell, T., Laskowski, S., Plaisant, C., Scholtz, J., and Whiting, M. VAST 2006 Contest: A Tale of Alderwood, In *Proc. VAST 2006*, IEEE Computer Society Press (2006), 215-216.
- [7] Grinstein, G., Cvek, U., Derthick, M., Trutschl, M. IEEE InfoVis 2005 Contest, Technology Data in the US, <http://ivpr.cs.uml.edu/infovis05>.
- [8] GS Data Generator. <http://www.GSApps.com>.
- [9] Plaisant, C. The challenge of information visualization evaluation. In *AVI '04*, ACM Press, (2004), 109-116.
- [10] Risch, J., Rex, D., Dowson, S., Walters, T., May, R., and Moon, B. The STARLIGHT information visualization system, in *IV'97*, IEEE Press (1997), 42.
- [11] Scholtz, J., Morse, E., and Hewett, T. In depth observational studies of professional intelligence analysts. *International Conference on Intelligence Analysis*, Proceedings available at <https://analysis.mitre.org/> (2005) {accessed 9/19/2007}.
- [12] Scholtz, J. Metrics for evaluating human information interaction systems. *Interact. Comput.* 18, 4 (2006). 507-527.
- [13] Text REtrieval Conference (TREC), <http://trec.nist.gov/>.
- [14] Theodoridis, Y., Silva, J. and Nascimento, M. On the Generation of Spatiotemporal Datasets, *Proc. Symp. Large Spatial Databases (SSD)*, (1999), 147-164.
- [15] Thomas, J. and Cook, K., eds. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*, IEEE CS Press, 2005; <http://nvac.pnl.gov/agenda.stm>.
- [16] VAST 2006 Contest Results. <http://www.cs.umd.edu/hcil/VASTcontest06/results.html>.
- [17] Whiting, M. A., Cowley, W., Cramer, N., Gibson, A., Hohimer, R., Scott, R., and Tratz, S. Enabling massive scale document transformation for the semantic web: the Universal Parsing Agent™. In *Proc. Doc Eng 2005*, ACM Press (2005), 23-25.

- [18] Wong, P., Hetzler, B., Posse, C., Whiting, M., Havre, S., Cramer, N., Shah, A., Singhal, M., Turner, A., and Thomas, J. IN-SPIRE InfoVis 2004 Contest Entry, In *Proc INFOVIS'04*, IEEE Computer Society Press (2004), 216.