Prajna – Evacuation Traces Challenge

Edward Swing*

Vision Systems & Technology, Inc.

ABSTRACT

The Prajna Project is a Java toolkit designed to provide various capabilities for visualization, knowledge representation, geographic displays, semantic reasoning, and data fusion. Rather than attempt to recreate the significant capabilities provided in other tools, Prajna instead provides software bridges to incorporate other toolkits where appropriate.

This challenge required the development of a custom application for visual analysis. By applying the utilities within the Prajna project, I developed a robust and diverse set of capabilities to solve the analytical challenge.

KEYWORDS: Information Visualization, Software Toolkit, Knowledge Representation

INDEX TERMS: D.2.11 [Software Engineering]: Software Architectures - Domain-specific architectures; [Computer Graphics]: Methodology and Techniques - Interaction Techniques.

1 INTRODUCTION

This challenge involved tracking the movements of individuals within a building around the time of a terrorist event. Data provided included the floor plan of the building, the RFID identifiers for each employee, and their movements over a span of time.

The challenge included questions about the evacuation, identification of suspects, and identification of casualties.

VSTI develops software for diverse customers who face numerous analytical challenges. The solutions developed for this challenge should apply to other analytical challenges that our customers must face.

2 DEVELOPING THE SOLUTION

2.1 Analysis of the Problem

This challenge clearly required a display of the building with a temporal control. Such controls would allow us to display the movements dynamically, watching the activities of the various employees. In addition, we considered a display which charted movement over time in order to differentiate suspects, victims, and other employees.

The data files each had unique formats. The data files would require custom readers. In addition, neither the time index of the employee's movements nor the coordinates had units associate with them. This implied that the spatial and temporal values were relative. While this did not significantly influence the development of the analytical tool, it would mean that the tool would need to be adapted for other applications.

2.2 Building with Prajna

The Prajna Project includes significant capabilities in a variety of technology areas. It also provides various software bridges to enable application developers to use specialized toolkits. Therefore, part of the design of the solution included selecting the appropriate elements of Prajna to include.

The first task for any analytical challenge is parsing the data. Since the data files provided for this challenge followed no particular standard, we developed custom readers for each file. We did use the building data reader as the basis for enhancing Prajna's ability to handle Grid data.

Similarly, the traces suggested an enhancement to Prajna's tracking capabilities. We included the ability to toggle the employee's trace over time, showing the entire path of each individual through the building. We also enabled displaying all traces simultaneously to view trends.

We also created a new capability to display the building, using image double buffering for animation. We later adapted this technique into a Grid renderer for other applications. By providing a dynamic way to redisplay the data in an animated fashion, the Dynamic Trace Display provided a compelling and useful way to view the scenario.

We provided a Cumulative Distance Chart to display the cumulative distances travelled by using the JFreeChart bridge within Prajna. JFreeChart is an open source project that provides a Java toolkit for displaying graphical charts. We initially also created a chart of the movement rates of individuals over time, but discovered that this offered little benefit. We discovered that all employees evidently moved at the same speed when they are moving. This was a surprise, since we would assume the employees would be more hasty when trying to exit the building.

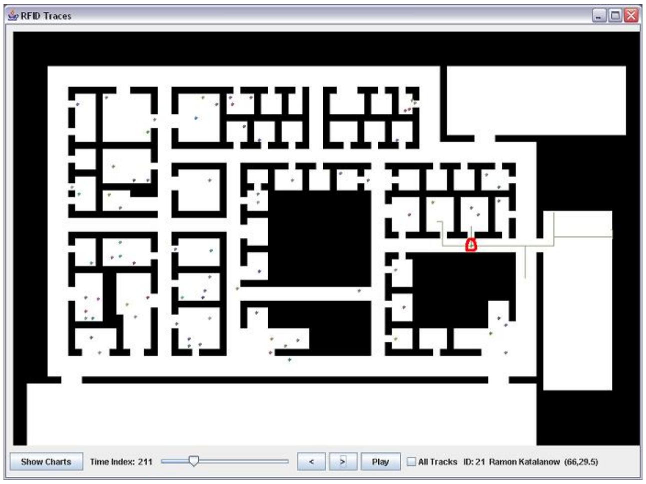
3 PERFORMING THE ANALYSIS

Once we completed the development of the application, we applied the capabilities of the tools to the challenge itself.

By using the dynamic trace display, we could easily discern the movements of each employee in the building over time. The first challenge was to determine when and where the bomb detonated. We easily identified when the bomb detonated by seeing when the employees began moving in reaction to the explosion. By observing the movements of the employees during the evacuation, we narrowed the possible areas where the bomb was set.

The next step was to identify likely suspects. The most obvious suspect, Ramon Katalanow, moved through the area of the detonation prior to the event. In addition, he paused in place for a period of time. We suspect he planted the bomb at this time. He then observed the detonation, and left the building. We also realized that Katalanow was a homonym for Catalano.

Email: deswing@vsticorp.com



Building Layout with occupants. Katalanow's movements are displayed; the location where he set the bomb is highlighted

In addition to Katalanow, we also wanted to identify if other employees were involved. Our immediate suspects were all of the other employees who were moving prior to the detonation. These employees would possibly be acting with foreknowledge of the event.

Of the four other individuals moving prior to the explosion, we noticed three of them all visited the same location. Two of them moved toward exits, while one of them actually was caught in the building collapse. We suspect one (Maxwell Lopez) left a message for the other suspects. The last person moving before the blast was likely moving to the entry lobby for other reasons, possibly meeting someone.

Identifying casualties was also simple. In the dynamic trace display, a number of employees failed to exit the building. Most of these were not moving for some time prior to the end of the scenario. The Cumulative Distance Chart corroborated this conclusion, showing horizontal lines for the victims after the blast. This would indicate these people are trapped, injured or dead.

We noticed several other unusual behaviors. One individual, Francisco Salter, did not move at all during the scenario. Questioning other employees about his role in the company and his activities may indicate whether his badge malfunctioned, or if he too were a victim.

Another odd incident happened when Katalanow saw Anton Knapp during the evacuation. Knapp turned toward Katalanow, then the two moved away from each other. This suggests something transpired between them

4 RESULTS

We were able to identify the prime suspect and other potential suspects easily, using the Dynamic Trace Display to recreate the movements of all employees during the scenario. Similarly, we identified the casualties as well. By corroborating our findings with the Cumulative Distance Chart, we feel that we have a good understanding of the situation before, during and after the event.

5 CONCLUSION

The tools developed for this challenge provided us the answers we sought. Furthermore, we were able to apply the principles and design of Prajna to this challenge, demonstrating its utility. The Prajna project attempts to provide a robust toolset, leaving the development of sophisticated visualization tools for other toolkits. In this fashion, Prajna may adopt the best visualization techniques by providing a software bridge to innovative toolkits.

Much of the components developed for this challenge were significant enhancements to the Prajna toolkit. While we developed a number of custom components for this challenge, we identified ways in which we could apply them to other problems. VSTI is evaluating some of the components for use in immersive simulations. VSTI has already applied the experience we gained in scenario representation

By providing an innovative architecture, which extends with software bridges to a variety of toolkits, Prajna avoids competing with the rapid pace of development across the spectrum of technology. Instead, Prajna offers developers the utilities to integrate new technology for knowledge representation in an intelligently designed architecture.

REFERENCES

[1] E. Gamma, et al (the Gang-Of-Four), "Design Patterns: Elements of Reusable Object-Oriented Software", Addison-Wesley Professional, January 1995.

[2] D. Gotz, M. Zhou and V. Aggarwal, "Interactive Visual Synthesis of Analytic Knowledge", Proceedings of the IEEE Symposium on Visual Analytics Science and Technology, 2006.

[3] J. Heer and M. Agrawala, "Software Design Patterns for Information Visualization", IEEE Transactions on Visualization and Computer Graphics (Proceedings), 2006.

[4] JFreeChart Project (website): http://www.jfree.org/jfreechart/