ABSTRACT

The Capstone Team has worked in conjunction with Defense Group Inc. (DGI) to develop a software tool called Radiological Assessment Device (RAD). RAD includes seven tools that perform the following functions:

- Searches for information regarding sealed sources
- Searches for information about historical radiological accidents/incidences
- Provides relevant assessment recommendations
- Allows for conversion between mass and activity
- Provides a mechanism for equivalent radiological unit conversions
- Returns critical exposure output based on user inputs relative to the type of scenario encountered

Through the above functionality, the goal is to support first responder decision making during radiological emergency response.

1 INTRODUCTION

This paper discusses the design, implementation, and testing of a radiological response software tool by a Capstone Team working together with members of the Defense Group Inc. (DGI). DGI currently has a software package called Chemical Biological Response Aid (CoBRA™) that helps first responders (FBI, HAZMAT teams, etc.) understand biological and chemical incidents. However, because of the heightened awareness of radiological threats a need has been identified to help first responders handle radiological situations. For these reasons, DGI and the Capstone team collaborated to expand the CoBRA™ software by adding radiological response capability, supporting first responder decision making during radiological emergency response.

RAD had to function within the framework of several different software programs. First, RAD needed to function within CoBRA™. Additionally, RAD needed to combine most of the functionality of two software programs: Briefing Improvised Radiological Devices (BIRD) and HOTSPOT.

Table 1: BIRD Scenarios Adapted from BIRD User’s Manual (Cima)

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Assumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact</td>
<td>The attacker positions the radiological source so that individuals’ skin comes in contact with radioactive material</td>
</tr>
<tr>
<td>External Near</td>
<td>The attacker positions the radiological source near a line, exposing individuals who pass by the source.</td>
</tr>
<tr>
<td>External Far</td>
<td>The attacker positions the radiological source near a road, exposing individuals who pass by in vehicles.</td>
</tr>
<tr>
<td>Internal Food</td>
<td>The attacker laces food with radioactive material and one or more individuals consume the contaminated food.</td>
</tr>
<tr>
<td>Surface Dispersal</td>
<td>The attacker scatters radioactive material over the ground.</td>
</tr>
<tr>
<td>Internal Plume</td>
<td>The attacker disperses the radioactive material through an aerosol device, factory smoke stack, or chimney.</td>
</tr>
<tr>
<td>Dispersal Device</td>
<td>The attacker disperses the radioactive material through high explosives.</td>
</tr>
</tbody>
</table>

The U.S. Army developed BIRD to brief troops in Bosnia about the threat of radiological weapons. BIRD enumerates seven scenarios (described in Table 1 above) for a radiological weapon attack: contact, external near, external far, internal food, surface dispersal, internal plume, and dispersal device. Based on exposure level in millisieverts (mSv) calculated from user inputs specific to each scenario and the number of individuals near the attack scene, BIRD places exposed individuals into one of six different radiation exposure level (REL) categories shown in Table 2 below. External exposures occur in the external near, external far, and surface dispersal scenarios. In these scenarios, exposed individuals’ skin does not come in contact with the source or isotope involved, but individuals receive exposure to the high-energy gamma particles radiating from the source or isotope; gamma particles cause exposure by penetrating up to several centimeters of human tissue (“Characteristics”). Contact exposures occur in the contact scenario when exposed individuals’ skin touches radioactive material. Finally, internal exposures occur when individuals ingest or inhale radioactive material; such exposures are applicable in the internal food, internal plume, and dispersal device scenarios. For the internal plume and dispersal device scenarios, BIRD uses values from HOTSPOT to calculate the exposure received by individuals at seven different downwind distances from the release point.

Table 2: REL Categories and Corresponding Limits in mSv Adapted from BIRD User’s Manual (Cima)

<table>
<thead>
<tr>
<th>REL</th>
<th>Designation</th>
<th>Maximum Exposure (mSv per year)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>External</td>
</tr>
<tr>
<td>1a</td>
<td>Public Limits</td>
<td>1</td>
</tr>
<tr>
<td>1b</td>
<td>Normal Occupational</td>
<td>50</td>
</tr>
<tr>
<td>1c</td>
<td>Special Occupational</td>
<td>100</td>
</tr>
<tr>
<td>1d</td>
<td>No Acute Effects</td>
<td>700</td>
</tr>
<tr>
<td>2</td>
<td>Minor Acute Effects</td>
<td>1500</td>
</tr>
<tr>
<td>3</td>
<td>Injuries</td>
<td>&gt; 1500</td>
</tr>
</tbody>
</table>

Lawrence Livermore National Laboratory developed HOTSPOT to enable first responders to produce exposure level contour plots of radioactive plumes resulting from radiological dispersions and nuclear weapon explosions.
LITERATURE REVIEW

Testifying before the Senate Foreign Relations committee, Henry Kelly from the Federation of American Scientists (FAS) recommended that an “effective disaster response” system be used to access radiological emergencies. The system requirements that Kelly outlined, such as promptly calculating the extent of the damage, and developing a sound response plan, are some features of RAD.

RAD is an example of a first response tool that was created to access radiological attacks. Many other companies and government agencies are realizing the importance and need for such systems. For instance, as a way to uncover and monitor radioactive substances, researchers at the Oak Ridge National Laboratory, in Tennessee, are currently developing a nation-wide system called SensorNet. It can detect radiological materials and transfer the information to a national operations center. Incorporated into SensorNet is a software called Hazard Prediction and Assessment Capability (HPAC). HPAC is currently used by all military command centers at nuclear facilities. HPAC is a service created by Defense Threat Reduction Agency (DTRA) and is a computer program similar to RAD. It helps first responders calculate the direction of a radioactive plume over different time intervals.

In addition, to estimate the health risks and contamination from radiological attacks, Levi and Kelly use the software program, Hotspot to model the dispersed radioactive materials and combine the results from Hotspot with theoretical and experimental data.

DESIGN PHASE

After the problem analysis, the team moved on to the next step in the Systems Methodology, design.

Design consisted of creating the user interface prototypes using Microsoft Visual Basic 6.0 and structuring the database design for the project using ER diagrams. Designing solid interfaces was an important component of RAD, since the tool was to be intuitive and easy for first responders to use. Even though first responders were expected to have the knowledge and skills to make decisions under radiological scenarios, the team assumed that first responders would not have expert computer skills.

Before design, a set of metrics was formed to evaluate the excellence of our interfaces compared to other alternatives of our own and previous CoBRA™ designs. Figure 5 gives the criteria the team came up with to decide on the best interface.

Cost = Number of Unique Mouse Clicks
Accuracy = Estimated Number of User Errors
Decision = Min \{Sum of Cost and Accuracy\}
also used as a metric, since it was important to have a system that provided correct results on which first responders could base their decisions. Accuracy was based on an estimated number of unique errors that a user might perform using the interface. The selected interface was chosen by taking the alternative with the smallest total between cost and accuracy.

After evaluating the alternatives and discussing them with DGI, the final interfaces were selected and are shown in Appendix A. For consistency, each screen had the same color scheme. Buttons were enlarged to allow for the system to shift into touch screen mode. After the design phase, the team embarked onto the implementation phase.

4 IMPLEMENTATION

The following describes in detail the process undergone to develop RAD’s functionality.

4.1 SSD Search Tool & Accident Search Tools

The SSD Search Tool helps users locate certain safety sheets and their accompanying information. Users searching for a safety sheet want to understand information about the source from which a radiological disaster might originate and can search for particular safety sheets by any of the below methods:

- Source Name
- Manufacturer Name
- Model Number
- Isotope Activity
- Isotope Name
- Facility Type

Here the manufacturer name refers to the person or company responsible for manufacturing a source such as fire detectors. Likewise, the model number gives a distinctive number for a distinctive type of source. “Facility Type” identifies a type of location, building, or general area where a source may be found or built. If an individual wanted to approach the search for a source based on what isotope types or levels of activity it accommodates they could also do so.

A first responder using the Accident Search Tool, may want to search for a particular radiological accident in history. RAD allows users to search the database of past radiological incidents by the following search methods:

- Accident Name
- Accident Date
- Isotope Activity
- Accident Location
- Isotope Name
- Facility Type

36 accidents have been sufficiently documented and recorded on professional document files which have been placed into a database by our team. Once a correct record is located by the system, similar to the SSD Search tool, a user has the ability to click on the file retrieved in the output screen and pull up a soft version of the information on the screen for immediate viewing. Though each tool requires different informational inputs and pulls information from different tables in the database, the two tools necessitate the same code.

Overall, the process entailed creating a single SQL statement from the user inputs, a combination of text and conditions, and then opening a connection string in Visual Basic 6.0 (VB), which linked the database, created by our team in Microsoft Access, to our program. Data from the database was retrieved using a Structured Query Language (SQL) statement and then used to populate a data grid control in VB. The data grid control simply displayed the determined results in a table like format where a user could then select and view a PDF file. A series of Case statements ultimately constructed the conditions of the SQL statement. Even though the program compiled several lines of code, all of the compilation occurred behind one button, the Execute button. The only other button available on the search screen is the Reset button, and here also a set of Case statements allowed the resetting control to function properly.

4.2 Conversion Tools

The Unit Conversion tool converts between units of radioactive dose (Sv to Rems) and activity (Ci to Bq). The conversion tool has a simple design and programming was fairly straightforward. The main code involved the "Convert" button, which executed the appropriate conversion equation and displayed the output in the appropriate space, and the "Reset" button, clears the entries. Error checking was an important part of the coding process. Some large values produced output outside the size range of the output area; thus, a rounding function was used to round these numbers to 10 decimal places.

The Source Term Conversion tool presented several unique challenges. Given a specific isotope the calculation tool will convert activity strength to approximate mass and volume or convert source mass impurity data to approximate activity value. Filling the combo boxes meant that the database had to be queried. The database needed to be queried again when the user selected an isotope in the combo box list in order to get values from the database for use in the calculations. Error checking throughout the coding process helped to find and eliminate errors.

4.3 Exposure Assessment Tool (EAT)

Critical steps for implementing the EAT involved accessing data from the database in Visual Basic, accessing exposure factors used in the BIRD algorithms, finding and integrating the relevant HOTSPOT algorithms into the EAT, and using the data from the database along with user
input to produce accurate exposure assessment calculations and plume models.

ActiveX Data Objects (ADO) allowed for easy access to the database within the software tool. The key to obtaining information from the database using ADO is creating a SQL query. SQL queries typically follow the form “SELECT Column 1, Column 2, Column 3, ... FROM Table 1, Table 2, Table 3, ... WHERE Criterion 1 AND / OR Criterion 2 AND / ORCriterion 3 AND / OR ...”. For example, the Isotope table in the database shown in Figure 6 on the next page contains the columns Isotope Name, Isotope NameFull, Half-Life (yr), Half-life (sec), Atomic Mass (amu), and Density (g/cm^3).

Figure 6: Isotope Table in Microsoft Access

The query “SELECT Isotope Name FROM Isotope” displays a list of all of the rows in the Isotope Name column from the Isotope table. To effectively use SQL within Visual Basic, one must create a query that returns the desired information from the database and then store the query to a variable of string type, a variable consisting of zero or more characters of text.

In order to display the results of the query, one can create an ADO Recordset, a table, and store the results of the query to the ADO Recordset using the Open(Query, Connection, ...) function. The Open(Query, Connection, ...) function requires a minimum of two parameters. The Query parameter, in this case, is the string containing the SQL query; the Connection parameter is an ADO variable of connection type. An ADO connection variable consists of the connection string containing the connection provider, the name and location of the database, and security information. An example connection string is “Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:\Database1.mdb;Persist Security Info=False.”

After querying the database and storing the results in an ADO Recordset, one can load the results into a table displayed onscreen using a DataGrid control in Visual Basic. An example of the code necessary to display a list of sources in a DataGrid control (called “SourceGrid” in this example) appears in Figure 7.

```
<Create ADO Connection>
Dim cnSource As ADODB.Connection
Set cnSource = New ADODB.Connection
With cnSource
    .ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source= & App.Path & ;Persist Security Info=False"
End With

<Create ADO Recordset>
Dim RADRS As ADODB.Recordset
Set RADRS = New ADODB.Recordset

<Open ADO Connection>
cnSource.Open
RADRS.CursorLocation = adUseClient

<Create SQL String>
Dim SQLQuery As String
SQLQuery = "SELECT [Source].[Source Name], [Source].[Manufacturer Name], [Source].[Model Number], [Source].[Isotope Name], [Source].Isotope Activity" & 
" FROM [Source] ORDER BY [Source].[Source Name];"

<Open ADO Recordset and Run SQL Query>
RADRS.Open SQLQuery, cnSource
Set SourceGrid.DataSource = RADRS

<Load ADO Recordset into DataGrid Control>
Set SourceGrid.DataSource = RADRS
```

Figure 7: Sample Code for Displaying Information from the Database in Visual Basic

After figuring out how to access data from the database within Visual Basic, it was necessary to code the BIRD algorithms into the EAT. As previously mentioned, BIRD contains varying algorithms for each attack scenario based on the specific scenario assumptions. The team found all of the necessary algorithms for each scenario in BIRD’s help manual. Implementing the calculations for non-plume scenarios (all scenarios excluding Dispersal Device and Internal Plume) was quite simple. Since non-plume scenarios do not calculate the number of victims at downwind distances from a radiological source, each non-plume scenario only required one value for the expected number of casualties. Thus, only one dosage calculation was required to determine casualties’ radiation exposure. By passing the two values, the exposure factor and isotope activity, into the dosage function the radiation dosage could be calculated. Thus, after a user submitted his or her information related to the scenario, the radiation dosages were immediately calculated.

Implementing the dosage calculations for the plume scenarios followed the same general concept as the non-plume scenarios. However; the dosage calculations for the plume scenarios required Chi-values, the time integrated activity in air of the dispersed material, from Hotspot’s algorithms. These Chi-values were stored in an array, and
were later retrieved when they were needed to calculate the dosages. Since the Chi-values were different for each of the seven distances (i.e. users need to enter expected number of casualties at each of the seven different distances), seven different radiation dosages had to be calculated. In relation to the BIRD algorithms, the team wrote the necessary code to find, load, and calculate the appropriate exposure factor from the database based on the isotope or isotope(s) involved and the given scenario. For example, the internal plume scenario requires an isotope specific inhalation exposure factor, since victims inhale the radioactive material. The contact scenario, in contrast, requires an isotope specific contact exposure factor, since the radioactive material contacts a victim’s skin.

The BIRD dispersal device and internal plume scenarios required obtaining information from HOTSPOT. The dispersal device and internal plume calculations in BIRD require Chi values for seven different downwind distances. The Chi value equation, taken from the HOTSPOT help manual, appears in Figure 8.

![Figure 8: Chi Value Equation Diagram Adapted from HOTSPOT Help Manual](image)

HOTSPOT calculates Chi values for individuals at a height of \( z \) (receptor height) an \( x \) distance downwind from the release along the centerline of the plume (this implies that \( y \) equals zero in the Chi value equation). \( \sigma_y \) and \( \sigma_z \) (shown in Table 3) vary based on the \( x \) distance, the atmospheric stability, and the terrain type (standard or city); city terrain decreases the standard deviation of the plume in the \( y \) and \( z \) directions, because the material has to disperse through buildings. \( Q \) is the activity of the release in curies (Ci) and \( \lambda \) is the inverse of the half-life for the particular isotope released (sec\(^{-1}\)). In order to determine the exposure received by an individual, one must calculate a Chi value in terms of \((Ci \times \text{sec}) / m^3 \) and then convert the result into becquerels (Bq) (specifically, \((Bq \times \text{sec}) / m^3 \)) by multiplying the Chi value by \(3.7 \times 10^{10} \text{ Bq} / \text{Ci}. \) Multiplying the result obtained from the previous conversion by the breathing rate (0.000333 m\(^3\)/sec) (how much of the material actually enters an individual’s lungs) times an inhalation dose factor specific to the isotope or isotopes involved (how much of an internal dose the inhaled material produces) times one thousand (1000 mSv per sievert (Sv)) determines an individual’s exposure level in terms of mSv.

<table>
<thead>
<tr>
<th>Table 3: Equations for ( \sigma_y(x) ) and ( \sigma_z(x) ) [Gifford]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HOTSPOT Equations for ( \sigma_y(x) ) and ( \sigma_z(x) ) [Gifford]</strong></td>
</tr>
<tr>
<td><strong>Terrain Type</strong></td>
</tr>
<tr>
<td>Standard</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>City</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

The previous paragraph describes the procedure for calculating Chi values and exposures in the internal plume scenario; the calculations differ slightly for the dispersal device scenario in that Chi values must be calculated using upwind virtual source terms, \( dy \) and \( dz \). In order to calculate \( dy \) and \( dz \), one must take the equations for \( \sigma_y(x) \) and \( \sigma_z(x) \) and set them equal to 0.5 \( x \) cloud radius and 0.2 \( x \) cloud top respectively. In each case, one must then solve for \( x \); the \( x \) solution is \( dy \) or \( dz \). After solving for \( dy \) and \( dz \), one can then calculate Chi values by replacing \( \sigma_y(x) \) and \( \sigma_z(x) \) in the equation shown in Table 3 with \( \sigma_y(x + dy) \) and \( \sigma_z(x + dz) \) respectively. Using the information provided in the previous two paragraphs, I coded the HOTSPOT Chi value algorithms into the EAT for both the internal plume and dispersal device scenarios.
The HOTSPOT help manual lacked instruction explaining HOTSPOT’s methods for graphing a plume contour plot. By carefully examining the Chi value equation, the appropriate method of producing a plume contour plot was inferred. In order to produce a plume contour plot, one must solve the Chi value equation for y and graph y values for varying values of x. The equation in terms of y appears in Equation 1.

$$y = \frac{2 \times \ln(C(x,y,z,H)) - \ln \left( \frac{Q}{(\sigma_{yx}(y))^2 \sigma_{zx}(y) \sigma_{x}(y)} \right) - \ln \left( \frac{(x)^2}{2 \times \sigma_{x}(y)} \right)}{2 \times \sigma_{x}(y)}$$  (1)

Plugging different Chi values (different values for C(x,y,z,H)) into the equation produces different contours. The team intended the plume contour plot to display six different contours, one contour for each REL category. BIRD provides lower and upper mSv limits for defining each REL category (see Table 2 for a table of REL categories). In order to obtain a Chi value, one must divide the lower mSv limit by \((3.7 x 10^{10} \times \text{breathing rate} \times \text{inhalation dose factor} \times 1000)\). One can then place this value in the y equation in place of Chi and produce a plume contour for the particular exposure level. Performing the same steps for all of the REL levels produces a contour plot of the plume. The team obtained ActiveImage, a graphics component that can be inserted into any Visual Basic project. Using ActiveImage, the team produced the plume contour plot by coloring pixels based on x and y-coordinates (y comes from solving the equation in Equation 1 based on a given value of x and Chi) different colors for each REL category. One advantage of ActiveImage is that it allows the coder to save the plume contour plot to a JPEG or GIF file for later use.

Table 4: Test Pass Summary for Scenarios

<table>
<thead>
<tr>
<th>Scenario</th>
<th># Failure/Total Test Cases</th>
<th>% First Test Failure</th>
<th>Final % of Failure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact</td>
<td>7/22</td>
<td>31.8</td>
<td>0</td>
</tr>
<tr>
<td>Disp. Device</td>
<td>2/17</td>
<td>11.7</td>
<td>0</td>
</tr>
<tr>
<td>External Far</td>
<td>4/23</td>
<td>17.4</td>
<td>0</td>
</tr>
<tr>
<td>Ext Near</td>
<td>1/21</td>
<td>4.8</td>
<td>0</td>
</tr>
<tr>
<td>Internal Food</td>
<td>5/27</td>
<td>18.5</td>
<td>0</td>
</tr>
<tr>
<td>Internal Plume</td>
<td>1/20</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>Surf Disp</td>
<td>4/26</td>
<td>15.4</td>
<td>0</td>
</tr>
</tbody>
</table>

After figuring out how to calculate Chi values and produce a plume contour plot, the team began to create test plans outlining different scenarios for testing the completed software tool. At this point (early March), the testing phase of the project began. The next chapter thoroughly discusses testing.

5 TESTING

To date, the team has completed the majority of implementation testing, which consists of comparing output from the software tool with output from BIRD and HOTSPOT, checking for coding errors, and looking for instances where user input produces system crashes. The team still needs to complete two other types of testing: usability testing, testing of the system according to usability heuristics; and integration testing, checking to ensure that the completed radiological response software tool functions correctly within CoBRA™.

The indices of performance in the goals tree served as metrics that the team followed in the testing phase. During the testing phase, the radiation dosage calculations and REL categories were measured against those of BIRD’s, and all the values tested for the seven scenarios correctly matched. Any errors that were encountered during the tests were corrected then re-tested. Table 4 below shows a summary of the test results, outlining the percentage of successes during the first test pass. The goal was to bring the percentage of successful passes to 100% and was achieved according to the tests conducted.

Comparing the final results to those in the beginning, the accuracy of the calculations were improved, and according to the test plan, the test pass was 100% successful (0% failure). Thus, according to the goals tree, this component was successfully implemented.
6 CONCLUSION
Incorporating RAD into the existing CoBRA™ framework is a first step in developing a radiological response system. What the Capstone team developed thus far is an excellent starting ground to build a better radiological assessment device for first responders. Having a system, such as RAD, to assist first responders under stress and pressure, is an invaluable asset for them. Therefore, RAD has the potential of helping first responders save more lives.

APPENDIX A: FINAL INTERFACE DESIGNS

Figure A-1: SSD Search Tool

Figure A-2: Accident Search Tool

Figure A-3: Source Conversion Tool

Figure A-4: Unit Conversion Tool

Figure A-5: EAT Select Method (2nd input) Screen

Figure A-6: EAT Spectrometer Input Screen (3.A)
REFERENCES


AUTHOR INFORMATION

All team members on the DGI Capstone team are 4th year Systems and Information Engineers at the University of Virginia.