



ARL-TR-8216 • Nov 2017



Scalable Database Design of End-Game Model with Decoupled Countermeasure and Threat Information

by Decetria Akole and Michael Chen

Approved for public release; distribution is unlimited.

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.



Scalable Database Design of End-Game Model with Decoupled Countermeasure and Threat Information

by Decetria Akole

*The Thurgood Marshall College Fund Program Internship
Alabama A&M University, Huntsville, AL*

Michael Chen

Weapons and Materials Research Directorate, ARL

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY) November 2017		2. REPORT TYPE Technical Report		3. DATES COVERED (From - To) June 5, 2017 – August 11, 2017	
4. TITLE AND SUBTITLE Scalable Database Design of End-Game Model with Decoupled Countermeasure and Threat Information				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Decetria Akole and Michael Chen				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) US Army Research Laboratory ATTN: RDRL-WMP-A Aberdeen Proving Ground, MD 21005				8. PERFORMING ORGANIZATION REPORT NUMBER ARL-TR-8216	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT An input file to End-Game Model (EGM) in active protection system simulations contains coupled countermeasure (CM) and threat information, the data structure of which possesses significant shortcomings in scalability given rapidly growing threats. For instance, accounting for 3 CMs and 50 threats requires 150 separate input files, which is hardly manageable. This report describes a systematic study of a relational database development, an approach that can consolidate all threat information into one individual table. This investigation began with analyzing existing Excel files, organizing component structures, distinguishing multivalued attributes, and identifying data dependency. A number of individual entity-relationship diagrams were iteratively developed followed by implementation of the corresponding relational schemas and table designs. Microsoft Access was adopted to construct the EGM database, where referential relationships can be enforced to ensure data integrity. After migration of the EGM data into the Access tables, C++ code employing an object linking and embedding database connection along with a series of standardized query language statements was created for data retrieval. Scalability was demonstrated that the required number of tables is linearly proportional to the number of CMs. Best of all, information of an emerging threat for EGM analysis can be accommodated with only one additional record in the threat table.					
15. SUBJECT TERMS active protection system, SSES, MAPS, End-Game Model, relational database, vehicle survivability, countermeasure					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 64	19a. NAME OF RESPONSIBLE PERSON Michael Chen
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified			19b. TELEPHONE NUMBER (Include area code) (301) 394-5900

Contents

List of Figures	v
List of Tables	vi
Acknowledgments	vii
1. Introduction and Background	1
1.1 US Army Active Protection System	1
1.2 Overview of End-Game Model in APS	2
1.3 Introduction of Excel: Current State of the Art	5
1.4 Microsoft Excel vs. Access	10
2. Research Plan	11
2.1 Objective	11
2.2 Hypothesis	11
2.3 Experiment Plan	11
3. Database Design	12
3.1 EGM Data Structure: Current State of the Art	12
3.2 E-R Diagrams	17
3.3 Relational Schemas	21
4. Implementation	24
4.1 Creation of Relational Database in Access	24
4.2 Consolidation of Databases	32
4.3 Data Population	34
5. Information Retrieval, Analysis, and Results	38
5.1 Development of C++ Code for Information Retrieval	38
5.2 SQL Analysis and Execution	41
5.3 Code Modification/Replacement Attempt	42

6. Summary and Conclusion	42
7. References	44
Appendix. List of Structured Query Language (SQL) Queries	45
List of Symbols, Abbreviations, and Acronyms	51
Distribution List	53

List of Figures

Fig. 1	SSES simulation features of APS	2
Fig. 2	Scenario of an APS event	3
Fig. 3	Dynamics of fragment fly-out.....	4
Fig. 4	Main operational procedures of an RPG for EGM	5
Fig. 5	Data structure of EGM input Excel file, part 1 of 2.....	6
Fig. 6	Data structure of EGM input Excel file, part 2 of 2.....	7
Fig. 7	CIAPS CM section current state of the art.....	13
Fig. 8	CIAPS threat and critical components section current state.....	14
Fig. 9	SRCM CM current state of the art	15
Fig. 10	SRCM threat and critical components current state	15
Fig. 11	Iron Curtain APS current state of the art	16
Fig. 12	IC Critical Components and Threat current state of the art.....	17
Fig. 13	CIAPS E-R diagram.....	18
Fig. 14	SRCM E-R Diagram	20
Fig. 15	Iron Curtain E-R diagram	21
Fig. 16	CIAPS relational schema	22
Fig. 17	SRCM relational schema	23
Fig. 18	Iron Curtain relational schema.....	24
Fig. 19	Screenshot of creating a blank database in Microsoft Access	25
Fig. 20	Titling database in Access	25
Fig. 21	Changing view to Design View	26
Fig. 22	Titling CIAPS table.....	26
Fig. 23	View after renaming Table 1 to CIAPS	26
Fig. 24	Making CIAPS the primary key	27
Fig. 25	Adding all the field attributes to the CIAPS table	28
Fig. 26	All tables in CIAPS.....	28
Fig. 27	CIAPS and Associate_CIAPS tables relationship	29
Fig. 28	Critical_Components and Associate_CIAPS tables relationship	29
Fig. 29	Relational schema tables for CIAPS in Access	30
Fig. 30	Relational tables schema for SRCM in Access.....	31
Fig. 31	IC relational tables schema in Access.....	32

Fig. 32	Consolidated E-R diagram.....	33
Fig. 33	Consolidation table schema in Access	34
Fig. 34	Excerpts of CIAPS in Access	35
Fig. 35	Excerpts of SRCM in Access.....	36
Fig. 36	Excerpts of IC in Access.....	37
Fig. 37	Excerpt of the Critical_Components table in Access	38
Fig. 38	C++ code for informational retrieval of Access database.....	40
Fig. 39	Descriptive screenshots.....	41

List of Tables

Table 1	Excel and Access pros and cons	10
---------	--------------------------------------	----

Acknowledgments

The authors are grateful to the Thurgood Marshall College Fund for offering this wonderful opportunity.

The assistance with data population in the Access database by Mr Anthony Abrahamian and the sketch for the dynamics of fragment fly-out offered by Dr Mary Jane Graham are also appreciated.

INTENTIONALLY LEFT BLANK.

1. Introduction and Background

1.1 US Army Active Protection System

Due to the United States Army encountering numerous threats during war, the Army now uses an active protection system (APS), which seizes or diverts inbound threats. This has been possible with the use of a variety of countermeasures (CMs). Protection delivered by an APS secures vehicles in active fashion on top of traditional passive armors. An APS improves survivability by overcoming inbound threats, such as rocket-propelled grenades (RPGs), antitank guided missiles (ATGMs), tank-fired high-explosive antitank missiles, tank-fired kinetic energy (KE) rounds, and so on. Other threats, including indirect fire such as mortars and bomblets, and guided top-attack fallers, may be of concern as well. It is intended that a vehicle be equipped with layered protection technologies (i.e., an APS along with passive reactive armors).¹ A general APS consists of a sensor subsystem, such as a threat warner and a radar; a CM subsystem, like Iron Curtain, Iron Fist, or Trophy; and a data processor responsible for data filtering and fire control solution calculations.²

The US Army Research Laboratory (ARL) initiated the development of Survivability Suite Engineering Simulation (SSES) in support of the Army Modular Active Protection System (MAPS) program to provide end-to-end APS modeling and simulation capabilities. The SSES simulation features, as shown in Fig. 1, highlight a sequence of events, beginning with threat launch flash detection, threat trajectory tracking, gimballed actions for radar and launcher, data filtering, fire control solutions, CM launch, fuze operation, CM warhead detonation, fragment flight, and threat residuals.^{2,3}

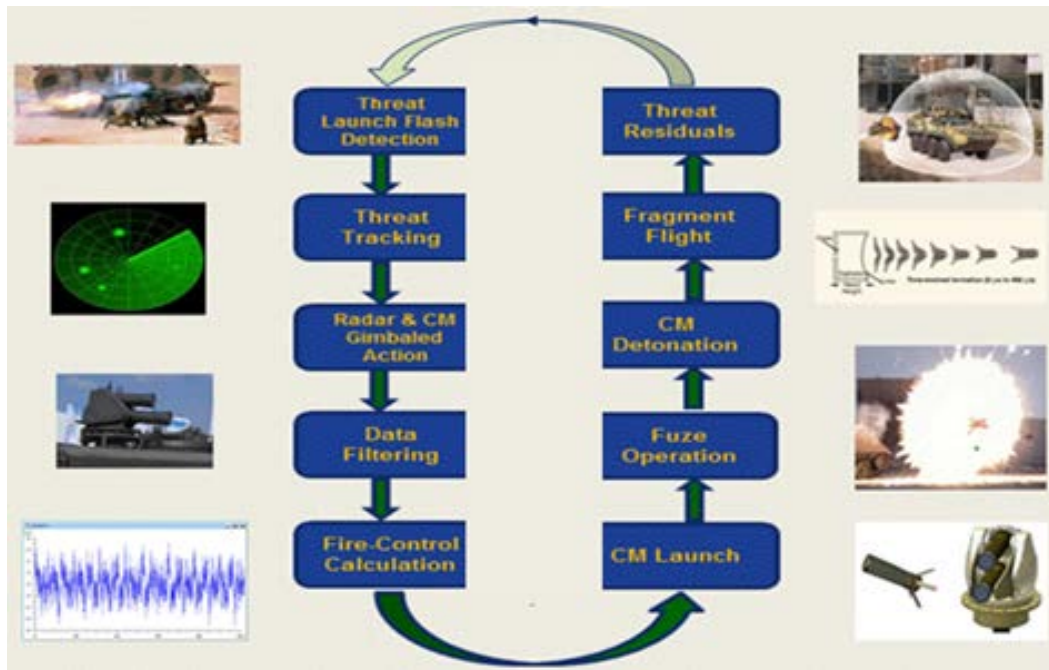


Fig. 1 SSSES simulation features of APS

1.2 Overview of End-Game Model in APS

An APS aims to destroy or disable an incoming threat before it hits the vehicle. An End-Game Model (EGM) in the APS refers to the engagement phase between a threat and a CM. The CM is typically on the vehicle that it intends to protect. Figure 2 illustrates a scenario of an APS event, where a threat is detected, and as the threat moves closer to the vehicle, the vehicle launches a CM and the moment it intercepts the threat is called the EGM.

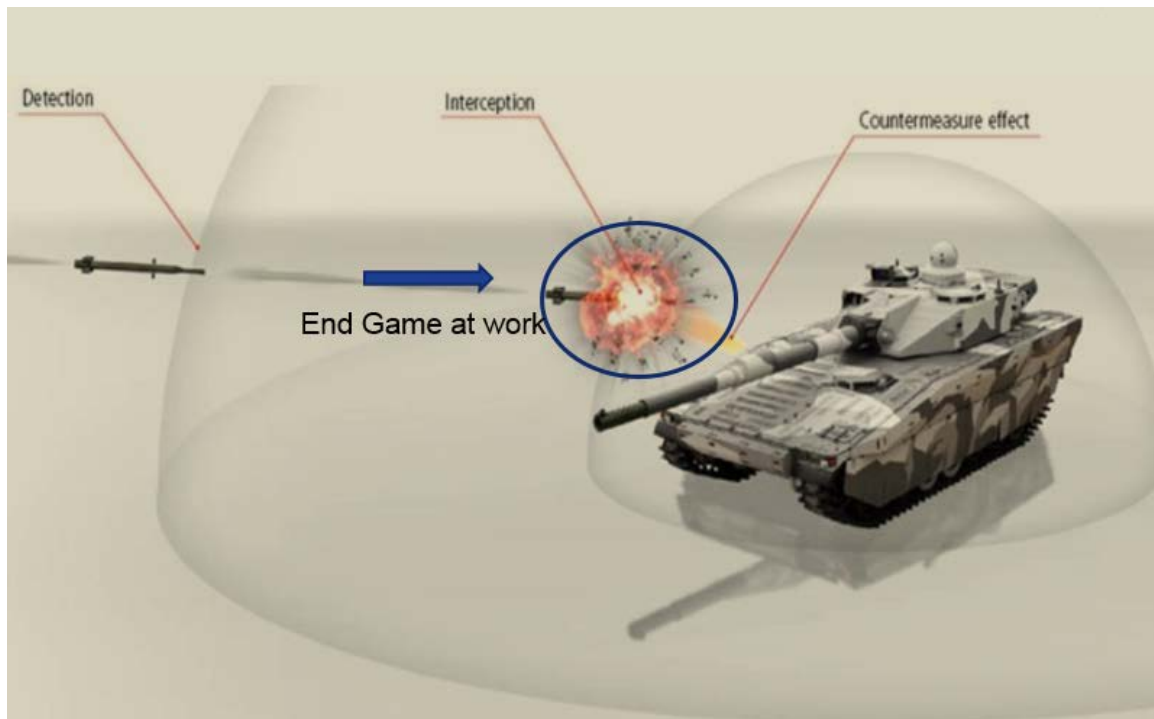


Fig. 2 Scenario of an APS event

The EGM consists of 2 major parts: 1) fragment fly-out and determination of hits on components, and 2) determination of an outcome for each warhead. The CM design contains the warhead data characteristics, such as number of fragments and each trajectory, and the threat contains detailed information for the possible CM hit locations, such as component number and component dimensions. The interaction between the CM and the threat determine the effect fragments impose on each component and store that data for engagement outcome analysis. The Survivability/Lethality Analysis Directorate (SLAD) at ARL developed State Machine methodology to identify possible outcomes of the engagement, which is used to determine the characteristics of the residual threat. A total of 8 possible outcomes are listed in the following:

- 1) Early Initiation with Normal Jet (EINJ)
- 2) Early Initiation with Damaged Jet (EIDJ)
- 3) Built-In Stand-off with Normal Jet (BISONJ)
- 4) Built-In Stand-off with Damaged Jet (BISODJ)
- 5) Fragment Induced Detonation (FID)
- 6) Fragment Induced Reaction (FIR)

7) Dismembered Warhead (DWH)

8) Dud (DUD)

One of the first EGM procedures when in the “run” mode is converting components from the threat coordinate system to the warhead detonation coordinate system, in the area where most of the EGM operations take place. With the suitable reference frames and before the establishment of the simulation loop to fly out fragments and find hits, the EGM decides if the CM is in a striking distance of the threat, calling CheckInPattern() function to assess whether a designated aim point on the threat is close enough to the threat and whether the threat is within the “spray” pattern. The beginning phase is shown in the first part of Fig. 3 where the CM is shaded gray. By iterating among all the fragments, the simulation flies out each one (see Fig. 3, middle schematic) and looks for hits on the threat by looping over the critical components. The sequence is stored in an array for the State Machine analysis, that is, storage of hits and timing, shown in the last part of Fig. 3.^{3,4}

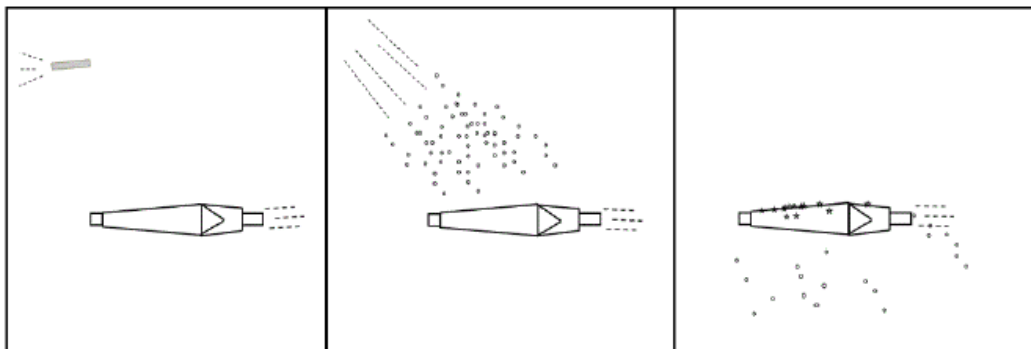


Fig. 3 Dynamics of fragment fly-out

Note that these outcomes are not the final stage or state of the threat. The final state is determined through State Machine methodology and is reliant on the sequence of events that would happen based on where and when the fragment hits the threat. For instance, when it comes to a unitary threat, there are 6 subcomponents to consider possibly causing detonation, in which it could likely be a dismembered warhead or partial detonation.^{3,4}

To provide illustration of the “run” procedure, Fig. 4 shows an example of an operational procedure for determining the outcomes of hits in EGM against an RPG. In this process, a run begins and determines if it should run a test mode. If it runs a test mode, then it sets up a hit list; if not, it runs FragFlyout() routine. From there, the program checks to see if it is a tandem warhead; if not, then it calls Find Outcomes- Generic RPG; however, if it is a tandem, then it calls StMachRun() routine and Return.⁴

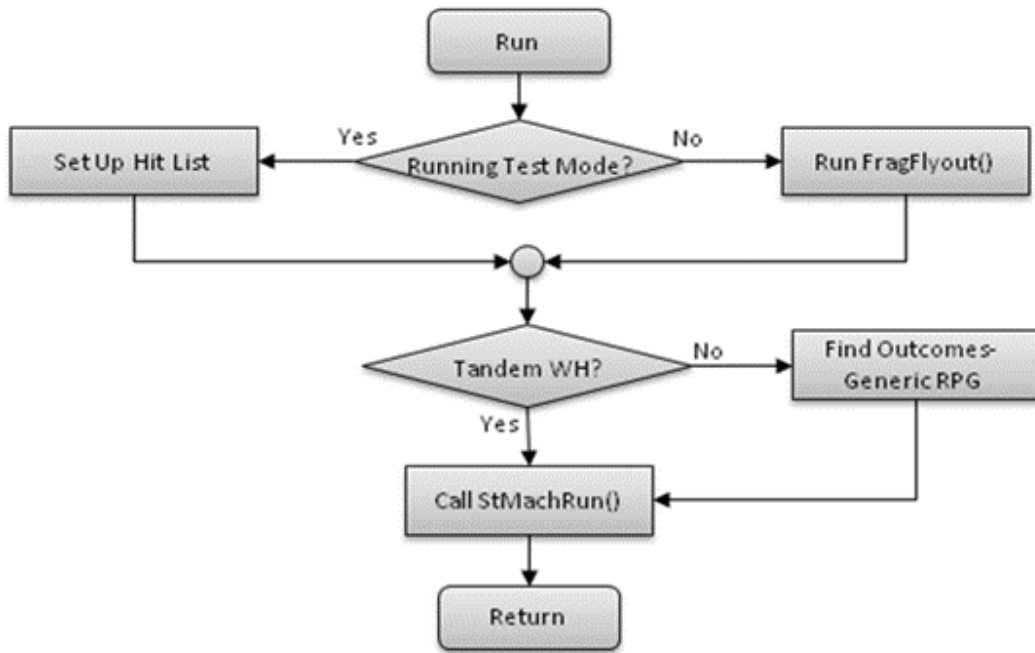


Fig. 4 Main operational procedures of an RPG for EGM

To enable the execution of the operation algorithms, data files based on Excel spreadsheets are being used in SSES. The following section introduces a general data structure of the Excel files for the EGM.

1.3 Introduction of Excel: Current State of the Art

The input to the EGM uses an Excel spreadsheet that contains coupled information related to the CM warhead and threat geometry configurations. Figure 5 demonstrates the top portion of a representative Excel file, where most of the APS CM information is shown. The spreadsheet serves as a template to guide users as to what data shall be provided and in what format. The numerical values have been removed because of data sensitivity in order to make the report publicly releasable. In practice, a utility function, embedded in the Excel file, is used to convert the data of the spreadsheet to a pure text file, which is then parsed and retrieved by the EGM software module.⁵ Some of the fields in the spreadsheet along with their definition descriptions are summarized in Fig 5.

Warhead Data					
WH configuration	flag				
Number of Shells	int				
Frag Mass	kg				
Frag Side	m				
Drag Coefficient	dimless				
WH Length	m				
WH Radius	m				
WH Cone Half Angle	deg				
Sigma V Axial	m/s				
Sigma V Radial	m/s				
Number of zones	int<=20				
	Zone Number	Ang Width (deg)	Num Frags (int)	Vel at Middle (m/s)	Outer Edg (m/s)
	0				
	1				
	2				

Fig. 5 Data structure of EGM input Excel file, part 1 of 2

- **Warhead Configuration:** an index flag to indicate the type of CM warhead. For instance, it could refer to Iron Curtain, Iron Fist, or Trophy APS.
- **Number of Fragments:** the total number of fragments associated with the warhead of the CM.
- **Frag Mass:** the mass of a single fragment assumed to be identical to all fragments resulting from the detonation of the CM warhead. In Iron Curtain, the mass is specified for each individual fragment.
- **Frag Side:** the length of a side of a cubic fragment or the diameter of a spherical fragment. In practice, all fragments are viewed as spheres in the EGM algorithms that determine hits on threat components, where the specified value is used as the diameter.
- **Drag Coefficient:** one of the factors to determine drag force in fragment fly-out model. The persistent drag coefficient, c_d' is defined as

$$\text{Drag force} = c_d' A_{\text{avg}} \rho_{\text{air}} v^2 \quad (1)$$

in which the A_{avg} is the existing area of the object using the uniform positioning theory, ρ_{air} is characterized as the density of the air, and v refers to its velocity relative to the air. For the case of Iron Curtain, the drag model and constant drag parameter, α , may be expressed as

$$V_z = V_o \exp(-\alpha z), \quad \alpha = \rho_{\text{air}} C_D A / 2m, \quad (2)$$

where A is the presented area of the object, and m is the mass of the particle.

Users may find the definitions of all the other fields, if of interest, in the reference.⁶

In the lower portion of the spreadsheet, a basic geometrical model of a threat along with its critical components, identified as Crit Comp 0, 1, 2, and so on, are included in Fig. 6.⁵ This section begins with a case matrix, primarily designed for a tandem warhead. The case matrix is composed of the main warhead, represented by the rows of the matrix, and the precursor, represented by the column. The matrix, populated with some index values, accounts for the combined effects of the main warhead and the precursor. These cases enable a qualitative evaluation of the overall outcome as desirable or undesirable from the standpoint of vehicle survivability. Users are referred to the report published by Bentley and Gleason for the definition of the index values in detail. Overall, the value of 99 populated in the cells stands for ineffectiveness, indicating that the effect of the outcome combination is not feasible. In other words, for a threat with a unitary warhead, all the entries in the table shall be 99s (i.e., not applicable).

Case Matrix		0	1	2	3	4	5	6	7	8	9	10	11
	0												
	1												
	2												
	3												
	4												
	5												
	6												
	7												
	8												
	9												
	10												
	11												
Threat Data													
Number of Critical Comps	int												
Nose to CG	m												
Length	m												
All positions in threat coord sys (origin at CG, x along axis, positive x towards front)													
		BEHIND nose.....											
Aim Point	m												
Crit Comp 0		Ogive/SW											
		BEHIND nose.....											
Top Pos (x, y, z)	m												
Bottom Pos (x, y, z)	m												
Top Radius	m												
Bottom Radius	m												
Base	m												
Apex	m												
Stopper Flag (1 = yes)	int												
NOT Flag (1 = yes)	int												
WH Flag (1 = yes)	int												
MF Flag (1 = yes)	int												
Activates Comps	int												
Mean Delays	s												
SD of Delays	s												
Det Chain	int												
Probabilities													
Min Vel	m/s												
Max Off-Axis Angle	deg												
DWH Code	int												

Fig. 6 Data structure of EGM input Excel file, part 2 of 2

The remainder of the spreadsheet delineates threat information followed by its corresponding critical components, which can apply to most threat types. A component of a threat is considered “critical” for EGM analysis if a fragment hit can influence the threat’s lethality against the vehicle. This section starts with geometric configuration and parameters that describe the effect of fragment hits on the critical components. Some field definitions are provided as follows:

- **Number of Critical Components:** the total number of critical components for the threat being analyzed.
- **Nose to CG:** the distances for the nose of the threat to the center of gravity (CG) of the threat, used to tie the simplified mechanical model to the threat positions provided in the system simulation.
- **Length:** distance from the Nose to CG (i.e., the overall length for the threat for graphics purposes).
- **Aim Point:** the anticipated intercept point of the center axis if the CM is on the center axis of the threat at the time of initial fragment impact. This is only used in the in-pattern assessment.

Please note Fig. 6 includes only 1 of the 6 critical components for the threat in this example. The fields of the parameters for Crit Comp 0 are identical to the remaining critical components, defined as the following:

- **Top Pos:** the position of the critical component closest to the threat’s nose.
- **Bottom Pos:** the position of the critical component closest to the threat’s tail.
- **Top Radius and Bottom Radius:** radii specified in the coordinate system holding origin at the ogive nose cone of the threat, where the planes of the circles are expected to be vertical to the line that connects with the midpoint of the top circle and the midpoint of the bottom circle.
- **Base and Apex:** the x coordinates of the front edge and the apex of the liner cone of a warhead. Both are used in evaluating hits on the warhead.
- **Stopper Flag:** one of the factors that is tied to the specified CM in the spreadsheet. If the flag is set to unity, a fragment striking that component will be stopped and will not be able to pass through to strike a neighboring component. If not, fragments can pass through components without the loss of energy and hit other components that lie along the fragment trail.
- **NOT Flag:** an indicator when it is placed or set as unity, the shape of the component is measured as a cavity and not a significant object. This permits

description of the volume in the liner cone of a shaped-charge warhead. A NOT component cannot be affected by a fragment since that component is actually a void.

- **WH Flag:** a value of zero represents non-warhead components, 1 stands for a precursor warhead, and 2 refers to a main warhead, applicable to both unitary and tandem threats.
- **MF (multi-function) Flag:** it specifies whether or not the component has multi-functional capability. Some components can only function once (MF = 0) and others can function multiple times (MF = 1). For instance, a booster can only explode once, but an ogive can function (by carrying a current) multiple times.
- **Activates Components:** list the component indices of up to 2 other components that are activated by the functioning of the component being specified.
- **Mean Delays and SD of Delays:** the mean and standard deviations (SDs) of 2 delays: Delay [0], Delay [1], DelaySD [0], and DelaySD[1].
- **Det Chain:** a list of components in operational order, which forms a chain that can produce detonation of the warhead in a “normal” routine.
- **Probabilities:** 2 fields referring to a 3-way draw, which must be numbers in the range from 0.0 to 1.0. For a warhead they are the single hit probability that a fragment will cause an FID and the single-hit probability that a fragment will cause an FIR. The single-hit probability that a fragment will cause only ordinary damage is the unity minus the sum of the 2 probabilities specified.
- **Min Vel:** the minimum fragment speed that will have any effect on the component. Hits by fragments with speeds below that value will not be scored as hits in the EGM. It is provided as a coarse filter to be used at the discretion of the user.
- **Max Off-Axis Angle:** the maximum striking angle relative to the axis of the threat at which the fragment should be considered to have any effect on the component. This is another coarse filter that can be used or set high at the discretion of the user.
- **DWH Code:** used in connection with the prediction of the dismembered warhead (DWH) outcome. It applies only to warheads and is the number of fragment hits on a warhead prior to the detonation of that warhead that will result in a DWH.⁵

1.4 Microsoft Excel vs. Access

Portability is one of the development goals in SSES. It is desired that the database used for EGM can be platform independent. However, given the limited time frame for the summer internship project and the lengthy internal approval process for non-standard database software, Microsoft Access Database, which is a part of the ARL standard computer image, was chosen for proof of concept. All the development in the database design and implementation can be carried over to a portable database in the near future. As previously mentioned, the EGM module for the SSES is currently taking input from Excel files that contained coupled threat and CM information. The Excel files are intended to be replaced with a relational database in Access. As a result, a comparison was made between Excel and Access as shown in Table 1.

Table 1 Excel and Access pros and cons

	Excel	Access
Pros	Easy to learn	Data structure and normalization through multiple tables
	Easy to store or populate data	Scalability: adding more data is free
	Simple way to organize data	Data and referential integrity
		Queries and reports
Cons	Large data is difficult to manage	Difficult to learn and requires a lot of skills to use
	Become problematic as the data grows	Copying and pasting is more challenging

There are pros and cons associated with Excel and Access. In general, the use of Excel is fairly straightforward while Access requires some knowledge in database design. When dealing with a large amount of data, Excel tends to be too cumbersome to operate while it is more manageable with Access. In addition, data and referential integrity can be easily enforced in Access. Further, the tables created within Access can be scaled up in conjunction with the expansion of the data set without much difficulty.

2. Research Plan

2.1 Objective

The objective of the research project is to investigate whether the EGM of an APS is more adaptable to proliferation of threats when a relational database (RDB) is implemented. This may be accomplished by replacing the data read function from existing end-game Excel files with the development of C++ code and Structured Query Language (SQL) queries for data retrieval.

2.2 Hypothesis

Due to rapidly evolving threats in combat fields, the design of an RDB to separate information of CM warheads from threats will exhibit higher scalability than an Excel file that contains the coupled information. It was hypothesized that with RDB, if the number of threats is increased by 10, then the effort to implement will be in proportion to 10 only (i.e., only 10 more records will be added). However, with the existing data structure, the effort to incorporate additional threats is augmented by the number of CMs. In the situation that there are 3 countermeasures of concern, it would lead to 30 additional spreadsheets that will need to be added, as opposed to only 10 records for an RDB.

2.3 Experiment Plan

An experimental plan was established to investigate whether the EGM of an APS is more adaptable to proliferation of threats when an RDB is implemented. Literature research to gain fundamental understanding of general engagement scenarios was conducted to become familiar with the State Machine methodology. In addition, investigation of existing Excel files as to the characteristics of CM warheads and threat configurations in an EGM was warranted to assess the data structure for the construction of a scalable database. The detail of the step-by-step experimental plan is outlined as follows:

- 1) Create an entity relationship (E-R) diagram for each CM. The E-R diagram is a data model describing how entities or concepts relate to one another. A total of 3 CMs (i.e., 3 available Excel files) will be investigated. They are Close-In APS (CIAPS), Short Range CM (SRCM) and Iron Curtain (IC).
- 2) Develop relational schemas based on previously created E-R diagrams. The relational schemas refer to the organization of data as a blueprint of how the database is divided into tables along with associated attributes, where primary keys and foreign keys are imposed to ensure integrity constraints.

- 3) Implement a query-able Access database. Based on the developed relational schemas, a number of tables will be generated, where the names of the table and the associated fields must be specified along with an appropriate data format on each field and their pre-defined interrelationship.
- 4) Populate data per table design hierarchy. The bottom-up data population must be followed so that referential data integrity can be enforced. Some weak entities exist in the data structure, which must be addressed and defined with cascade rules.
- 5) Create a C++ console application to leverage an object linking and embedding database (OLE DB) connection in Visual Studio. An OLE DB connection manager enables the application to connect to a data source, such as Microsoft Access, where dynamic SQL queries will be used to communicate with the database for data retrieval.
- 6) Replace the Excel files in the SSES EGM module with the Access database where the threat and the CM information are decoupled. Substitute the corresponding data read function with the SQL application, which must be integrated with the SSES EGM module.

3. Database Design

3.1 EGM Data Structure: Current State of the Art

As previously mentioned, the threat and CM data for EGM analysis and prediction currently co-reside in Excel files. In the investigation, 3 Excel files are available: one for CIAPS, one for SRCM, and one for IC. Figure 7 highlights the current state of the CIAPS CM warhead data, which contains the following attributes: Number of Fragments, Frag Mass, Frag Side, and Drag Coefficient. Each of the fragments shall be specified with associated Fragment dynamics parameters: x, y, z, vx, vy, vz, SD_dir, SD_v, yaw, phid, thtd, and phsid, based on the warhead coordinate system. A total of 280 segments is specified in this case.

An excerpt of the CIAPS spreadsheet listing the components of CIAPS countermeasure and Fragment													
		USER INPUTS											
		CALCULATED											
Warhead Data													
WH Configuration Flag	-		0 SRCM, 1 CIAPS-2 (statistics), 2 Cyl MultiShell, 3 Trophy (statistics), 4 Rectangular (detail), 5 Iron Curtain										
Number of Fragments	-	280	*There are a total of 280 Fragments										
Frag Mass	kg												
Frag Side	m												
Drag Coefficient	-												
Fragment dynamics (WHF)	x (m)	y (m)	z (m)	vx (m/s)	vy (m/s)	vz (m/s)	SD dir	SD v (m/s)	yaw (deg)	phid (deg/s)	thtd (deg/s)	psid (deg/s)	

Fig. 7 CIAPS CM section current state of the art

The next section of the CIAPS spreadsheet holds the Case matrix, which indicates the Main WH (rows), Precursor (columns), and the combined outcomes in the cells. The Case matrix is applicable to tandem warheads only. Therefore, a table named “Tandem” was specified in database design. Following the Case matrix is Threat Data, which holds the following attributes: Number of Critical components, Nose to CG and Length, and Aim Point. A total of 6 critical components are shown in this example, implying that the threat is equipped with a unitary warhead. Subsequently, the attributes of each critical component are highlighted one after another. Some of them include Activates Components, Delays, Positions, and Detonation Chain. Some attributes are dependant on the CM, which shall be separated out in the database design. A table named Associates_CIAPS is created to capture the dependancy. The names of the fields are Max-off_Axis_Angle, Min_Vel, PbFIR, PbFID, DWH_Code, Stopper_Flag and MF_Flag. Figure 8 highlights the current state of the art for the CIAPS EGM information.

The last data set of CM to be reviewed is Iron Curtain (IC). The current state-of-the-art IC warhead information is given in Fig. 11. IC fragments are explosively formed projectiles (EFPs), of which there are 35, as seen in the field named Number of Fragments. In addition, other unique attributes associated with the CM include CG circle radius for quads, ctr-to-ctr x separation, ctr-to-ctr y separation, CMM-to-CMM EFP x period, CMM-to-CMM EFP y period, sql speed, quad speed, sql alpha, quad alpha, sql mass, quad mass, sql side, quad side, quad div angle, and dept interval.

Warhead Data												
WH Configuration Flag	-			sql speed	m/s			sql mass	kg			quad div ang
Number of Fragments	-	35		quad speed	m/s			quad mass	kg			dept interval
CG circle radius for quads	m			sql alpha	1/in			sql side	m			
ctr-to-ctr x separation	m			quad alpha	1/in			quad side	m			
ctr-to-ctr y separation	m											
CMM-to-CMM EFP x period	m			distance between center of an EFP on one unit and corresponding EFP on the x adjacent unit								
CMM-to-CMM EFP y period	m			distance between center of an EFP on one unit and corresponding EFP on the y adjacent unit								
Fragment dynamics (WHF)												
fragment designation	x (m)	y (m)	z (m)	vx (m/s)	vy (m/s)	vz (m/s)	SD dir	SD v (m/s)	yaw (deg)	phid (deg/s)	thtd (deg/s)	psid (deg/s)
P front sgl												
PP front sgl												
M front sgl												
MM front sgl												
C back sgl												
P back sgl												
M back sgl												
C front quad-1												
C front quad-2												
C front quad-3												
C front quad-4												
P front quad-1												
P front quad-2												
P front quad-3												
P front quad-4												
M front quad-1												
M front quad-2												
M front quad-3												
M front quad-4												

Fig. 11 Iron Curtain APS current state of the art

Several fields are uniquely specific to a certain warhead design. To create a table that is more robust for a general design pattern, the field name of pattern 1 speed is used for “sql speed”, pattern 2 speed for “quad speed”, pattern 1 alpha for “sql alpha”, pattern 2 alpha for “quad alpha”, pattern 1 mass for “sql mass”, pattern 2 mass for “quad mass”, pattern 1 side for “sql side”, pattern 2 side for “quad side”, and pattern 2 div angle for “quad div angle”.

The EFP in the field of Fragment dynamics is a multi-valued attribute, containing x, y, z, vx, vy, vz, SD vir, SD v, yaw, phid, thtd, and psid. A separate table shall be created to capture the information.

Similar to CIAPS and SRCM, the case matrix and threat data follow the CM section of the IC spreadsheet, as shown in Fig. 12. Since the threat is equipped with a unitary warhead, all cells in the case matrix are populated with 99. The threat data

has the same attributes as those in CIAPS and SRCM, which leads to no additional requirement in the table design.

Case Matrix	0	1	2	3	4	5	6	7	8	9	10	11
0	99	99	99	99	99	99	99	99	99	99	99	99
1	99	99	99	99	99	99	99	99	99	99	99	99
2	99	99	99	99	99	99	99	99	99	99	99	99
3	99	99	99	99	99	99	99	99	99	99	99	99
4	99	99	99	99	99	99	99	99	99	99	99	99
5	99	99	99	99	99	99	99	99	99	99	99	99
6	99	99	99	99	99	99	99	99	99	99	99	99
7	99	99	99	99	99	99	99	99	99	99	99	99
8	99	99	99	99	99	99	99	99	99	99	99	99

In this excerpt from Iron Curtain spreadsheet list the Threat and Critical Components as they relate to Iron Curtain

Threat Data												
Number of Critical Comps	int	6.0										
Nose to CG	m											
Length	m											
All positions in threat coord sys (origin at CG, x along axis, positive x towards front)												
Aim Point	m											
Crit Comp 0		SW										
Top Pos (x, y, z)	m											
Bottom Pos (x, y, z)	m											
Top Radius	m											
Bottom Radius	m											
Base	m											
Apex	m											
Stopper Flag (1 = yes)	int											
NOT Flag (1 = yes)	int											
WH Flag (1 = yes)	int											

This section of the spreadsheet holds the same attributes as that of CIAPS and SRCM

Fig. 12 IC Critical Components and Threat current state of the art

3.2 E-R Diagrams

An E-R diagram demonstrates the relations of entity groups or sets placed in a database. In general, an E-R diagram may consist of the following 5 main components:

- 1) **Entity (represented by a rectangle)**. An entity stands for a concept, an object, or a place to store information. Entities are characterized in 3 categories: strong, weak, or associative. A strong entity is identified by its own attributes. A weak entity relies on the foreign key of another entity and an associative entity connects entities.
- 2) **Action (represented by a diamond shape)**. An action demonstrates how the entities in the design share information in the database.
- 3) **Attribute (represented by an oval shape)**. An attribute stands for a unique characteristic of an entity.
- 4) **Connecting line**. A connecting line connects the attributes, showing the relationships of the entities in the E-R diagram.
- 5) **Cardinality**. A cardinality states how many instances are associated to 1 entity and another. For example, there might be a mother (strong entity),

who has children (weak entity). The cardinality will be 1 mother to many children (i.e., 1:n or 1-to-many relationship).⁷

In consideration of the data structure in the CIAPS spreadsheet, an E-R diagram was developed and drawn in PowerPoint (shown in Fig. 13). An entity named CIAPS was created to capture the attributes of the CM, such as fragment mass, fragment size, drag coefficient, and so on. A weak entity named Fragment (double-lined rectangles) tied to the CIAPS table was created, which contains several attributes, such as positions, velocities, angles, and so on. A 1:n relationship between the CIAPS and the Fragment was specified since one CM may have one or more fragments. Similarly, a table named THREAT was created to capture the characteristics of a threat including geometric and aim point information. As mentioned previously, a threat may possess up to 13 critical components. As a result, another weak entity named Critical Components tied to the THREAT table was created. This entity consists of a good number of attributes including single-value and multi-value fields. For single-value ones, such as length and nose-to-CG, they are directly associated with the table of Critical Components. For multi-value attributes, such as Activate_Components, Delays, Positions, and Detonation_Chain, 4 additional tables were created (one for each). Such a design aims for better data management and scalability as the database grows over time. Among them, for instance, the table of Positions contains the attributes of Top and Bottom Positions in the x, y, and z directions.

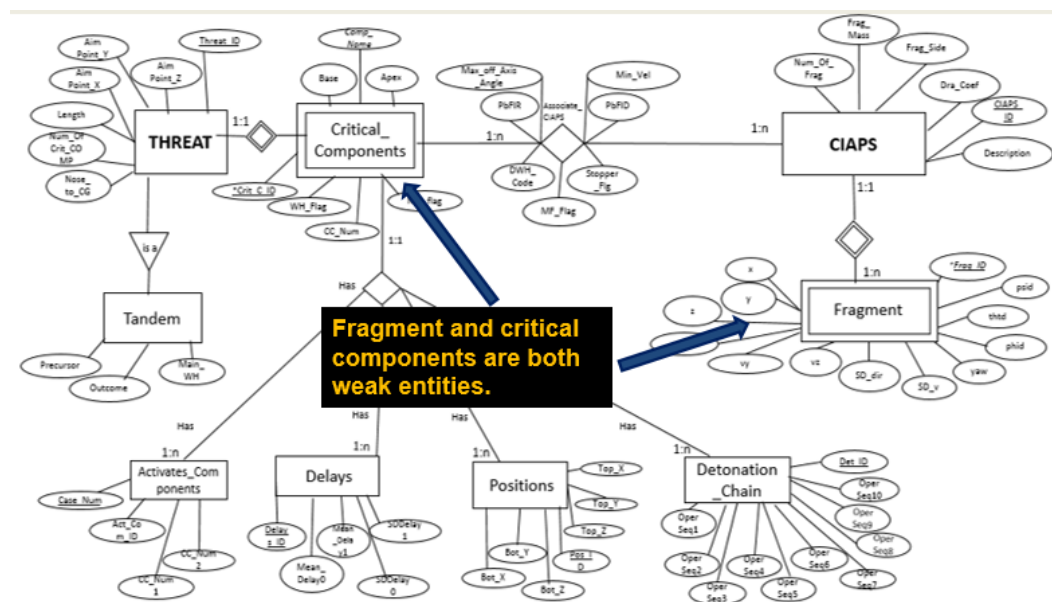


Fig. 13 CIAPS E-R diagram

In addition, in the threat category, tandem warhead is a special case that contains some unique features. As a result, a table named Tandem was specified as a

subcomponent of the THREAT table, which possesses the attributes of main-WH, precursor, and outcome in order to constitute the case matrix, shown in the spreadsheets. Some of the fields under the section of Critical Components exhibit dependency with the CM. Consequently, a relationship set named “Associate_CIAPS” was created to define the connection between the CIAPS and the Critical Components. It contains the attributes, such as Min_Velocities, Probabilities, Stopper_Flag, MF_Flag, and so on. Since the Associate_CIAPS specifies an n:n relationship, an additional table was required.

Finally, a total of 10 entities were developed in the diagram. All of them must come with a primary key (e.g., the CIAPS_ID in the CIAPS table), to distinguish one record from another. Note that the graphical presentation of an E-R diagram may vary slightly as opposed to that in the literature. However, the concept to define the basic entity components and the relationship shall be quite similar.

Moving forward is the assessment of the SRCM spreadsheet for its database design. An E-R diagram was developed to capture the overall SRCM information, shown in Fig. 14. It contains an SRCM table with SRCM_ID and Num_Of_Shells fields. The Num_Of_Shells leads to a weak entity of Shell with numerous attributes, such as Frag Mass, Frag Size, WH Length, WH Radius, Sigma V Axial, or Sigma V Radial. A total of 3 shells were specified in the spreadsheet. Each of the shells may contain up to 20 zones. As a result, another weak entity of Zone_SRCM tied to the Shell entity is required. The attributes of the Zone_SRCM include Ang Width, Num Frag, Vel at Middle, and Vel at Out Edge, which shall be populated in the table.

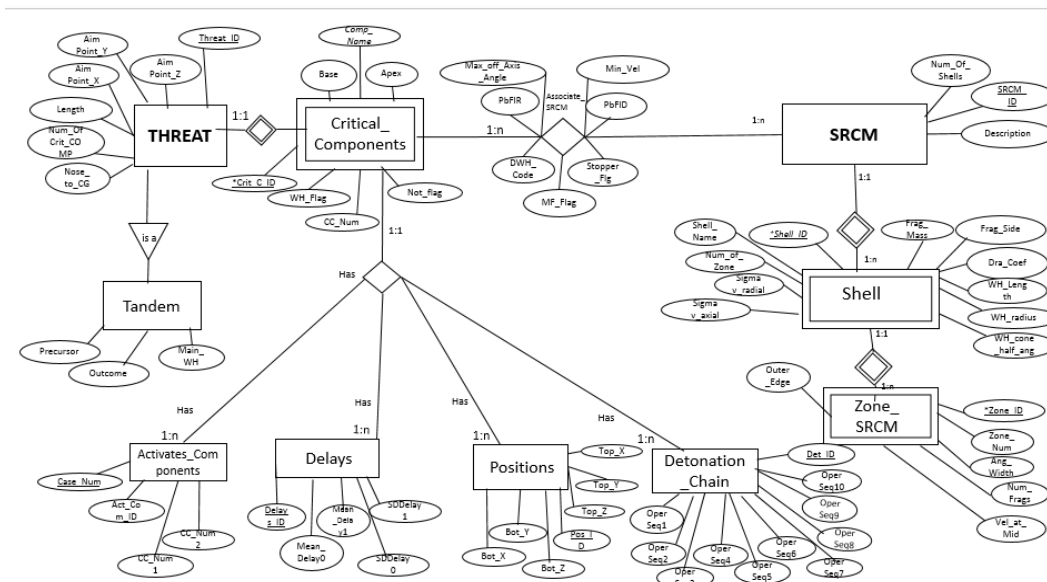


Fig. 14 SRCM E-R Diagram

It is similar to CIAPS in that the entities of THREAT, Tandem, Critical_Components, Activate_Components, Delays, Positions, and Detonation_Chain shall be created in the SRCM E-R diagram. The cardinality among the entities states a 1-to-many relationship, implying that no additional table is needed. Like CIAPS, to capture the dependency between the SRCM and the threat critical components, a many-to-many relationship set named Associate_SRCM shall be created. It warrants a separate table that shall cover the fields of Min_Velocities, Prob_FID, Prob_FIR, Stopper_Flag, and MF_Flag. Overall, the SRCM requires a total of 11 tables to be implemented according to the database design.

The last spreadsheet in the investigation was the Iron Curtain (IC) information. Based on the given data fields and data structure, an E-R diagram was developed (see Fig. 15). It is very similar to CIAPS and SRCM in that the entities of THREAT, Tandem, Critical_Components, Activate_Components, Delays, Positions, and Detonation_Chain are all included in the diagram along with the same cardinality. The areas in red circles highlight the commonality across all 3 CMs of study. On the CM side, an entity named IRON_CURTAIN was created along with numerous attributes, such as Ctr_to_Ctr x and y separations, CMM_to_CMM_EFP x and y periods, Sgl and Quad speeds, masses and sides, and so on. Because of the unique warhead design, a weak entity named EFP, tied to the IRON_CURTAIN entity, was created to capture the outcome of the warhead detonation. The attributes of the

EFP include the positions, the velocities, the standard deviations of the velocities, the angular coordinates, the angular velocities, and so on. Likewise, the interconnection between the IRON_CURTAIN and the Critical_Components entities shall be represented by a many-to-many relationship set named Associate_Iron_Curtain, which will be handled with a separate table that carries the primary keys of the IRON_CURTAIN and the Critical_Components. As shown in Fig. 15, a total of 10 tables are required in the implementation of the database for Iron Curtain.

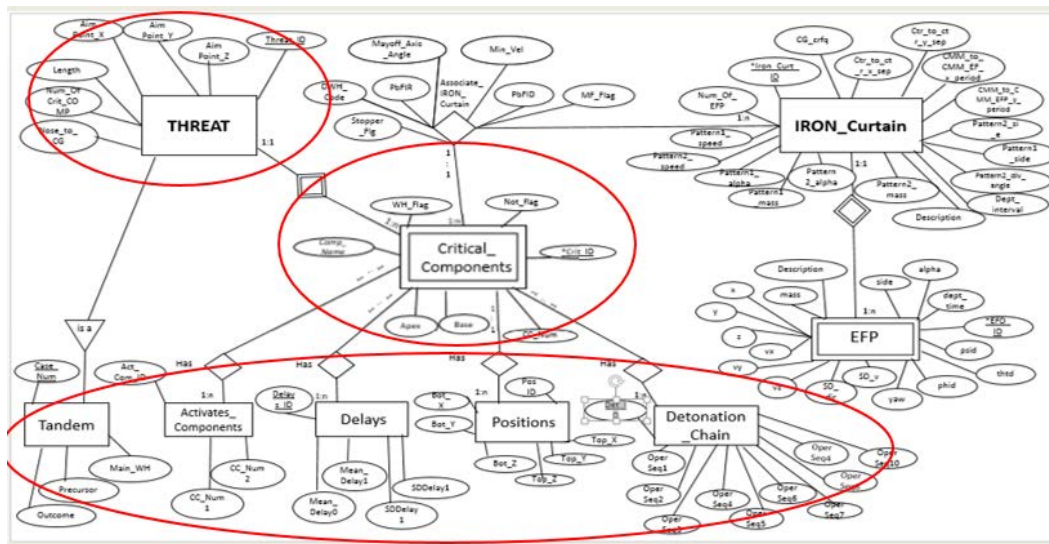


Fig. 15 Iron Curtain E-R diagram

3.3 Relational Schemas

A relational schema dictates how data should be organized in tables and shows how tables are related to each other. It serves as a blueprint of how the database is constructed. In the CIAPS database design, CIAPS_ID is a primary key added to the CIAPS table. Since Fragment is a weak entity, it shall carry the CIAPS_ID as a foreign key along with a unique identifier Frag_ID in the Fragment table. Similarly, the THREAT table contains a primary key of Threat_ID. The Critical_Components is a weak entity to the THREAT, which must carry both Threat_ID and Crit_C_ID as the primary key. The Tandem is a subclass of the THREAT, where the Case_Num and the Threat_ID constitute a primary key. In addition, the Associate_CIAPS defines a many-to-many relationship between the CIAPS and the Critical_Components; its primary key shall be defined by the following 3 fields: CIAPS_ID, Crit_C_ID, and Threat_ID. As for the Activate_Components, Delays, Positions, and Detonation_Chain tables, a primary key of Act_Com_ID, Delay_ID, Pos_ID, and Det_ID is specified, respectively. Figure 16 demonstrates the CIAPS

relational schema. As mentioned previously, a total of 10 tables shall be established to cover the information in the spreadsheet. In each table, the primary key is highlighted by the fields that are underscored.

Relational Data Schema for CIAPS

1. CIAPS (*CIAPS_ID, Num_of_Frag, Frag_Mass, Frag_Side, Dra_Coef, Description)
2. Fragment (*Frag_ID, CIAPS_ID, x, y, z, vx, vy, vz, SD_dir, SD_v, yaw, phid, thtd, phsid)
3. Associate_CIAPS (CIAPS_ID, Crit_C_ID, Threat_ID, PbFIR, PbFID, Stopper_flag, Min_flag, Min_vel, Maxoff_Axis_Angle, DWH_Code)
4. THREAT (*Threat_ID, Aim Point_X, Aim Point_Y, Aim Point_Z, Length, Num_of_Crit_Comp, Nose_to_CG)
5. Critical_Components (*Crit_C_ID, Threat_ID, CC_Num, Comp_Name, Apex, Base, Not_flag, WH_flag)
6. Tandem (Case_Num, Threat_ID, Main_WH, Precursor, Outcome)
7. Activates_Components (Act_Comp_ID, Crit_C_ID, Threat_ID, CC_Num1, CC_Num2)
8. Delays (Delays_ID, Crit_C_ID, Threat_ID, Means_Delay0, Means_Delay1, SDDelay0, SDDelay1)
9. Positions (Pos_ID, Crit_C_ID, Threat_ID, Top_X, Top_Y, Top_Z, Bot_X, Bot_Y, Bot_Z)
10. Detonation_Chain (Det_ID, Crit_C_ID, Threat_ID, Oper_Seq1, Oper_Seq2, Oper_Seq3, Oper_Seq4, Oper_Seq5, Oper_Seq6, Oper_Seq7, Oper_Seq8, Oper_Seq9, Oper_Seq10)

Fig. 16 CIAPS relational schema

With a similar approach to the transformation of E-R diagrams, the SRCM relational schema and the Iron Curtain relational schema were developed (see Figs. 17 and 18, respectively). The SRCM_ID is the primary key in table SRCM, which also appears in the table Shell as a foreign key. Subsequently, both the Shell_ID and the SRCM_ID serve as a foreign key in the table Zone_SRCM. Along with the Zone_ID, all 3 fields constitute the primary key of the table Zone_SRCM. In Fig. 18, the IRON_Curt_ID is the primary key in table Iron Curtain. The IRON_Curt_ID must be passed to the table EFP as a foreign key, which establishes the referential relationship. Likewise, the Associate_SRCM and the Associate_Iron_Curtain tables, which specify the relationship between the CM and

the threat critical component information, shall carry the primary key of the connecting entities (i.e., the SRCM_ID and the Crit_C_ID shall exist in table Associate_SRCM, and the Iron_Curt_ID and the Crit_C_ID must go to the Associate_Iron_Curtain table). All the other entities exhibit common attributes among all 3 cases. In the RDB design, the SRCM shall require 11 tables while a total of 10 tables are needed for Iron Curtain.

Relational Data Schema for SRCM

1. SRCM (SRCM_ID, Num_of_Shells, Description)
2. Shell (SRCM_ID, *Shell_ID, Shell_Name, Frag_Mass, Frag_Side, Dra_Coeff, WH_Length, WH_radius, WH_cone_half_ang, Sigma v_axial, Sigma v_radial, Num_of_Zones)
3. Zone_SRCM (*Zone_ID, Shell_ID SRCM_ID, Zone_Num, Ang_Width, Num_Frags, Vel_at_Mid, Outer_Edge)
4. Associate_SRCM (SRCM_ID, Crit_C_ID, Threat_ID, PbFIR, PbFID, Stopper_flag, Min_flag, Min_vel, Maxoff_Axis_Angle, DWH_Code)
5. THREAT (Threat_ID, Aim Point_X, Aim Point_Y, Aim Point_Z, Length, Num_of_Crit_Comp, Nose_to_CG)
6. Critical_Components (*Crit_C_ID, Threat_ID, Comp_Name, CC_Num, Apex, Base, Not_flag, WH_flag)
7. Tandem (Case_Num, Threat_ID, Main_WH, Precursor, Outcome)
8. Activates_Components (Act_Com_ID, Crit_C_ID, Threat_ID, CC_Num1, CC_Num2)
9. Delays (Delays_ID, Crit_C_ID, Threat_ID, Means_Delay0, Means_Delay1, SDDelay0, SDDelay1)
10. Positions (Pos_ID, Crit_C_ID, Threat_ID, Top_X, Top_Y, Top_Z, Bot_X, Bot_Y, Bot_Z)
11. Detonation_Chain (Det_ID, Crit_C_ID, Threat_ID, Oper_Seq1, Oper_Seq2, Oper_Seq3, Oper_Seq4, Oper_Seq5, Oper_Seq6, Oper_Seq7, Oper_Seq8, Oper_Seq9, Oper_Seq10)

Fig. 17 SRCM relational schema

1. IRON_Curtain (Iron_Curt_ID, Num_of_EFP, Description, CG_crfq, Ctr_to_ctr
x_sep, Ctr_to_ctr_y_sep, CMM_to_CMM_EFP_x_period,
CMM_to_CMM_EFP_y_period, Pattern1_speed, Pattern2_speed,
Pattern1_alpha, Pattern2_alpha, Pattern1_mass, Pattern2_mass,
Pattern1_side, Pattern2_side, Pattern2_div_ang, dept_interval)
2. EFP (*EFP_ID, Iron_Curt_ID, Description, x, y, z, vx, vy, vz, SD_dir, SD_v, yaw,
phid, thtd, psid, mass, side, alpha, dept_time)
3. Associate_Iron_Curtain (Iron_Curt_ID, Crit_C_ID, Threat_ID, PbFIR, PbFID,
Stopper_flag, Min_flag, Min_vel, Maxoff_Axis_Angle, DWH_Code)
4. THREAT (Threat_ID, Aim_Point_X, Aim_Point_Y, Aim_Point_Z, Length,
Num_of_Crit_Comp, Nose_to_CG)
5. Critical_Components (*Crit_C_ID, Threat_ID, Comp_Name, CC_Num,
Apex, Base, Not_flag, WH_flag)
6. Tandem (Case_Num, Threat_ID, Main_WH, Precursor, Outcome)
7. Activates_Components (Act_Com_ID, Crit_C_ID, Threat_ID, CC_Num1,
CC_Num2)
8. Delays (Delays_ID, Crit_C_ID, Threat_ID, Means_Delay0, Means_Delay1,
SDDelay0, SDDelay1)
9. Positions (Pos_ID, Crit_C_ID, Threat_ID, Top_X, Top_Y, Top_Z, Bot_X, Bot_Y,
Bot_Z)
10. Detonation_Chain (Det_ID, Crit_C_ID, Threat_ID, Oper_Seq1, Oper_Seq2,
Oper_Seq3, Oper_Seq4, Oper_Seq5, Oper_Seq6, Oper_Seq7, Oper_Seq8,
Oper_Seq9, Oper_Seq10)

Fig. 18 Iron Curtain relational schema

4. Implementation

4.1 Creation of Relational Database in Access

In accordance with the 3 relational schemas, an RDB was created in Access 2010. The steps for the creation of an RDB are described as follows. First, a blank database was produced in Access by choosing File>New and selecting “Blank database,” shown in Fig. 19. Subsequently, one shall define the directory path where the database will be saved. A name of “SSES_EndGame Model_DB1” was specified in the prompt, exhibited in Fig. 20. Once the “Create” button was clicked, a blank database with the specified filename was generated.

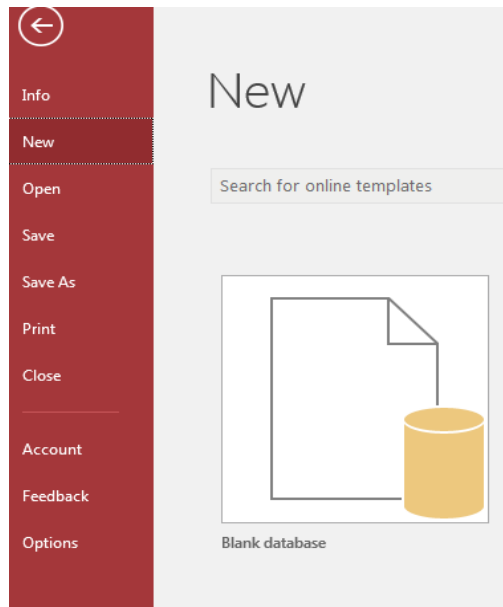


Fig. 19 Screenshot of creating a blank database in Microsoft Access

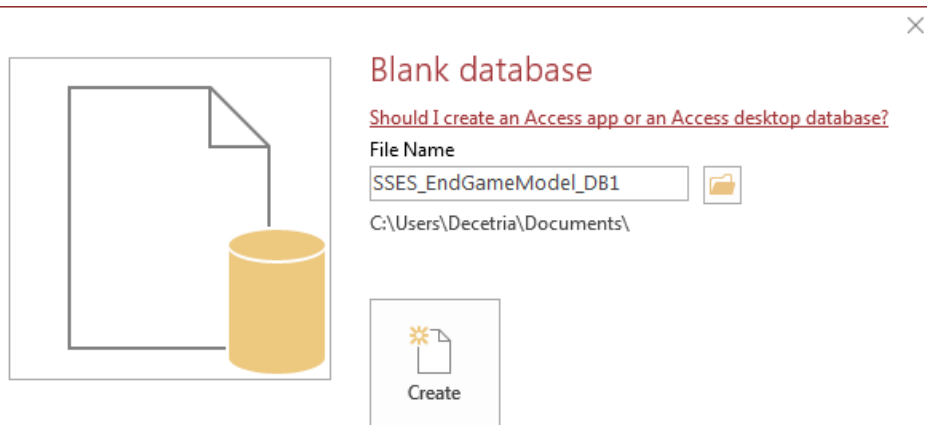


Fig. 20 Titling database in Access

The effort to construct tables of a database began with CIAPS. On the left-hand side of Fig. 21, the arrow under “View” was clicked, which triggered a dropdown menu. The option of “Design View” was chosen, prompting for the name of a table to be created, shown in Fig. 22. After replacing a default name of “Table 1” with “CIAPS” in the prompt, one would be given a template to define the Field Name and the Data Type for the table. A screenshot of the design template is provided in Fig. 23.

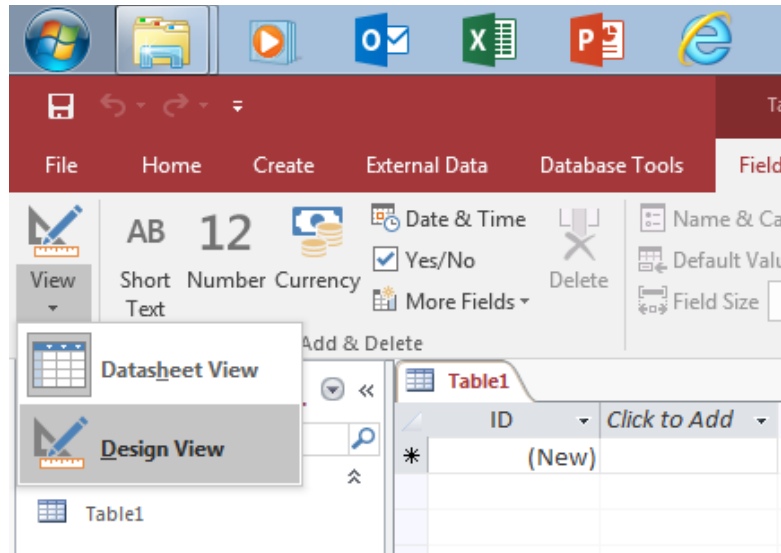


Fig. 21 Changing view to Design View

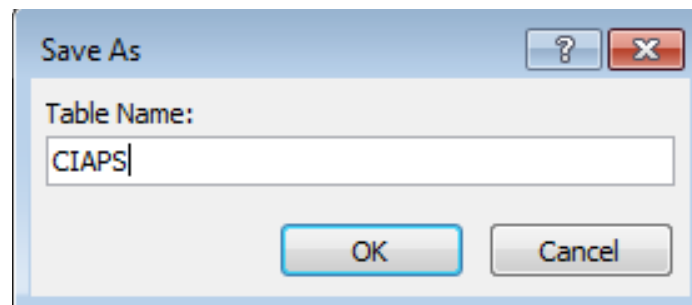


Fig. 22 Titling CIAPS table

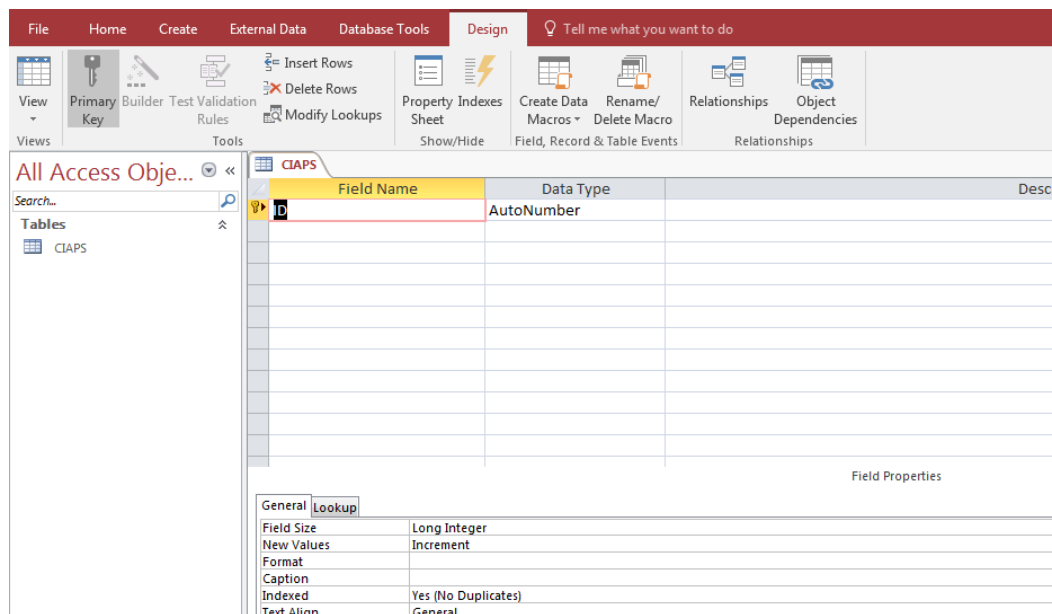


Fig. 23 View after renaming Table 1 to CIAPS

On the table design sheet, one can generate an identity of a column that typically categorizes the records in a table; that is, the columns represent the associated attributes of the table. In this case, a field name of “CIAPS_ID” was defined as a primary key, indicated by a key symbol next to the Field Name, highlighted in Fig. 24. Microsoft Access offers a list of available data types for users. They include “Short Text”, “Long Text”, “Number”, “Date/Time”, “Currency”, “AutoNumber”, “Yes/No”, “OLE Object”, “Hyperlink”, and “Attachment”. The option of “AutoNumber” was chosen because the primary key CIAPS_ID must be a unique identifier to distinguish one record from another. Thus, the automatically generated incremental numbers are sufficient.

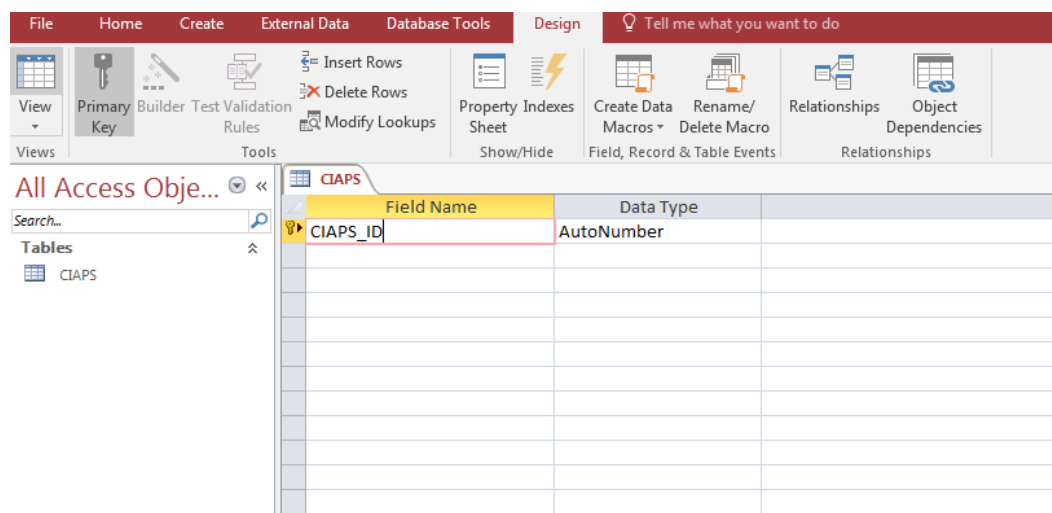


Fig. 24 Making CIAPS the primary key

Based on the CIAPS relational schema in Fig. 16, the table CIAPS shall include several other fields, such as Description, Number of Frag, Frag_Mass, Frag_Side, and Drag_Coef. The data types for the fields were specified as Long Text, Number, or Short Text (shown in Fig. 25). The Description field accommodates details about CIAPS. Defining the type for the Number of Fragment field as a Number is quite straightforward since it is an integer. It should be noted that the data types for the Frag_Mass, Frag_Side, and Drag_Coef fields were all specified as Short Text. They are supposed to be floating point numbers. However, due to the data compatibility between the communication of the Access database and the C++ code (introduced in Chapter 5), a short text shall be used. A conversion to “double” (floating-point values) type takes place in the C++ code after data retrieval. Additionally, in the lower portion of Fig. 25, users may specify some data properties or constraints for a certain field. For instance, the CIAPS_ID exhibits a field size of Long Integer and an indexed value of increment without duplicates.

The final step was to specify the relationships, which is critical to maintain the integrity of the database. Figures 27 and 28 demonstrate screens that show editing relationships between 2 tables. For example, the CIAPS and the Associate_CIAPS tables exhibit a 1-to-many relationship type. The common CIAPS_ID field from these 2 tables must be consistent in the data population. As a result, the option of “Enforced Referential Integrity” must be checked. Similarly, the primary key of the Critical_Components table (i.e., Crit_C_ID and Threat_ID), has to be referenced to the same field names of the Associate_CIAPS table so that data integrity can be maintained.

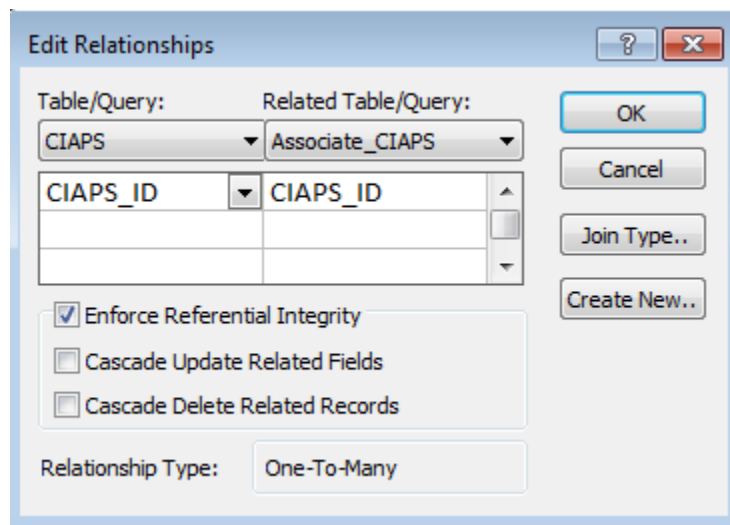


Fig. 27 CIAPS and Associate_CIAPS tables relationship

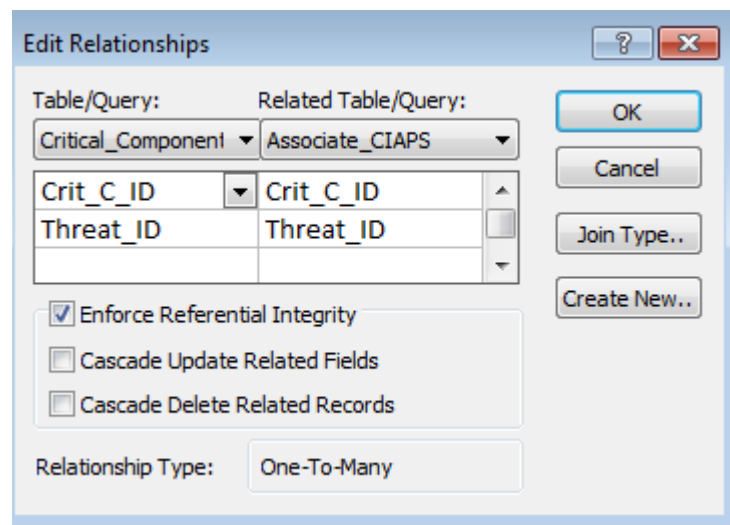


Fig. 28 Critical_Components and Associate_CIAPS tables relationship

Moving forward, the referential relationships shall be specified among all tables. The layout of the final table schema for CIAPS is shown in Fig. 29, where readers may find the names of the tables, the associated attributes and the primary key of each table, the referential relationships, and the cardinality among the tables.

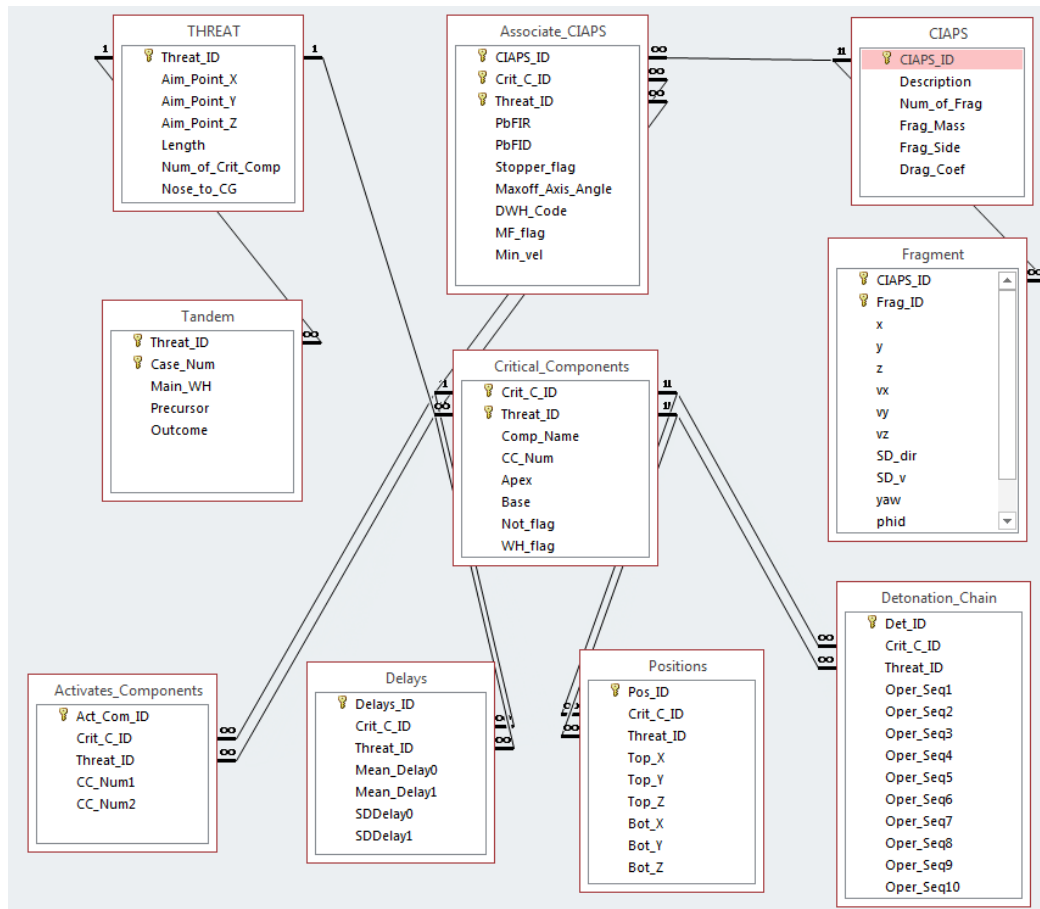


Fig. 29 Relational schema tables for CIAPS in Access

The same effort was made for the creation of the SRCM and the IC databases in Access. Figure 30 demonstrates the table schema of the SRCM, where a total of 11 tables were constructed, including SRCM, Zone, Shell, THREAT, Associate_SRCM, Critical_Components, Tandem, Activates_Components, Delays, Positions, and Detonation_Chain. For the IC database, another separate 10 tables were constructed, as shown in Fig. 31, which include Iron_Curtain, EFP, THREAT, Associate_Iron_Curtain, Critical_Components, Tandem, Activates_Components, Delays, Positions, and Detonation_Chain. Like CIAPS, the referential integrity was enforced in the table schemas of the SRCM and the IC.

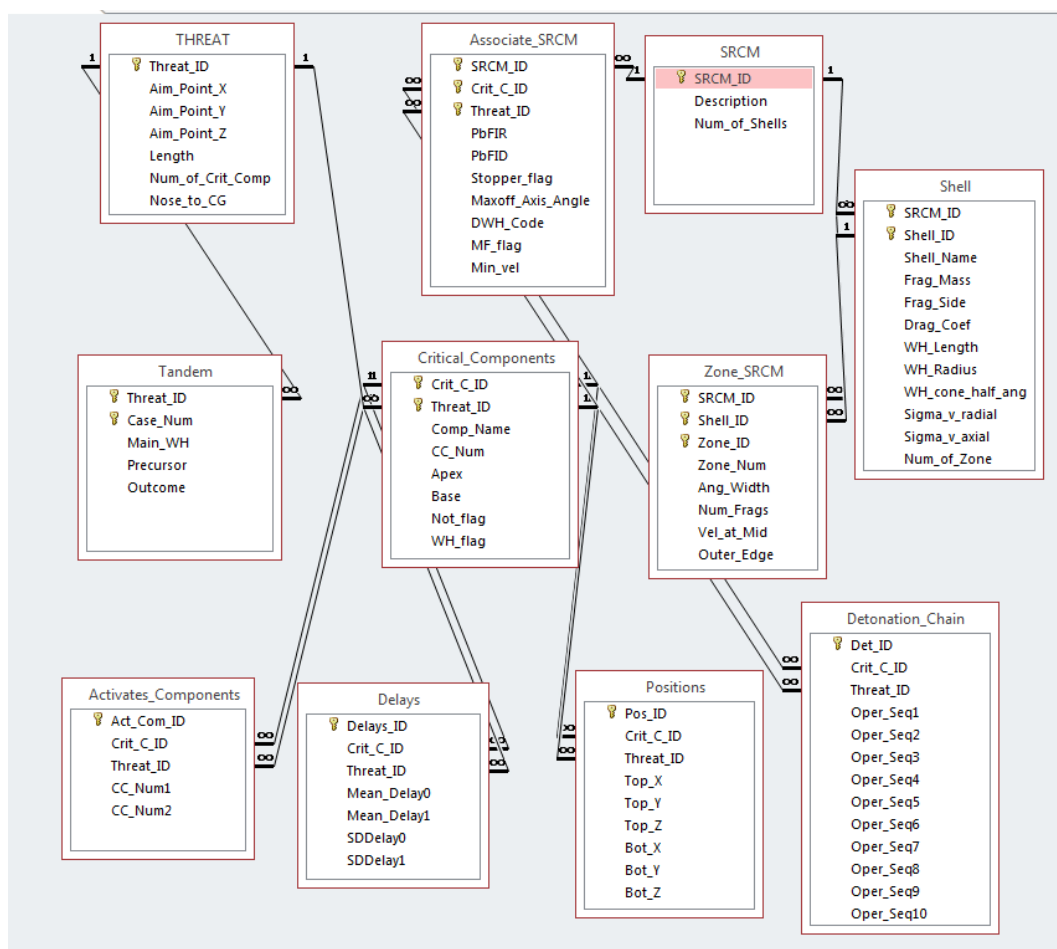


Fig. 30 Relational tables schema for SRCM in Access

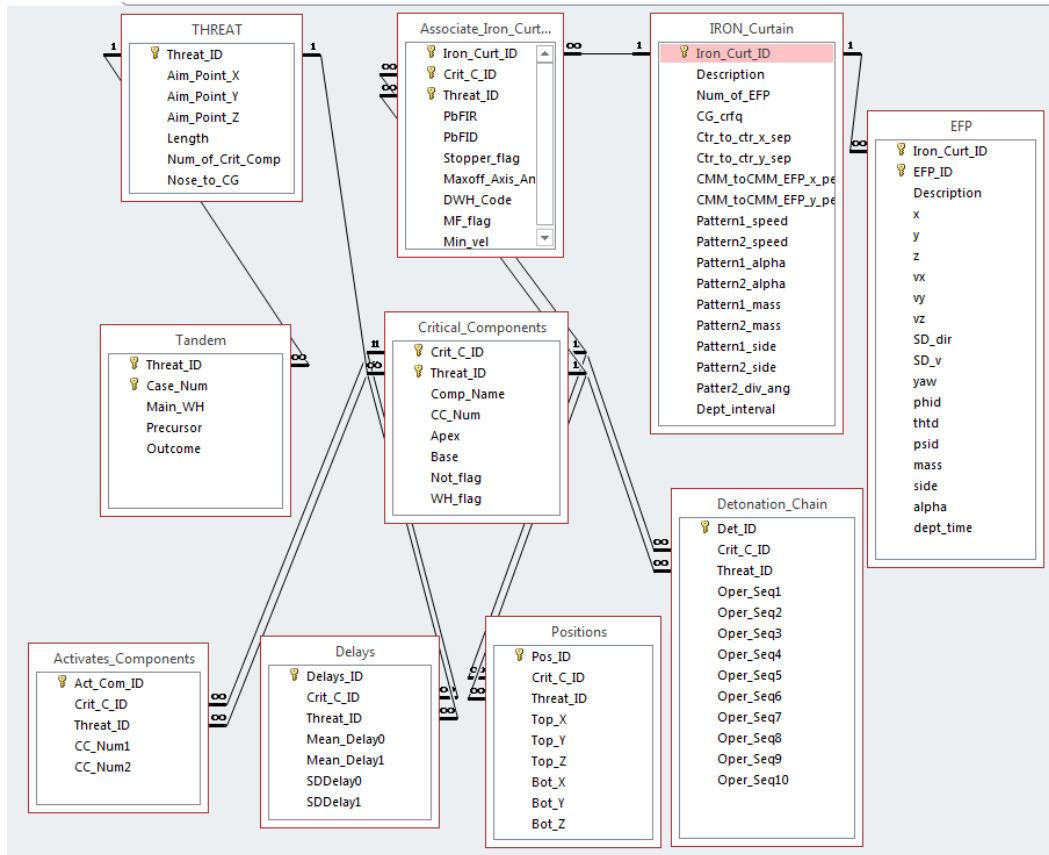


Fig. 31 IC relational tables schema in Access

4.2 Consolidation of Databases

In the creation of the CIAPS, SRCM, and IC RDBs, the CM tables are notably different. Specifically, the CIAPS and the Fragments tables are shown in the CIAPS RDB; the SRCM, the Shell, and the Zone_SRCM tables exist in the SRCM DB; and IC DB contains the Iron_Curtain and the EFP tables. These unique features shall be respectively handled. Interestingly, on the threat side, one may notice from a previous section that all 3 RDBs appear to possess common threat data structure and data format. The tables of THREAT, Tandem, Critical_Components, Activates_Components, Delay, Positions, and Detonation_Chain repeatedly occur in all 3 RDBs. As a result, the information could be consolidated and an integrated E-R diagram was developed, which is shown in Fig. 32. The tables accounting for the dependency between the CM and the threat, such as Associate_CIAPS, Associate_SRCM, and Associates_Iron_Curtain, shall be retained and included in the consolidated diagram.

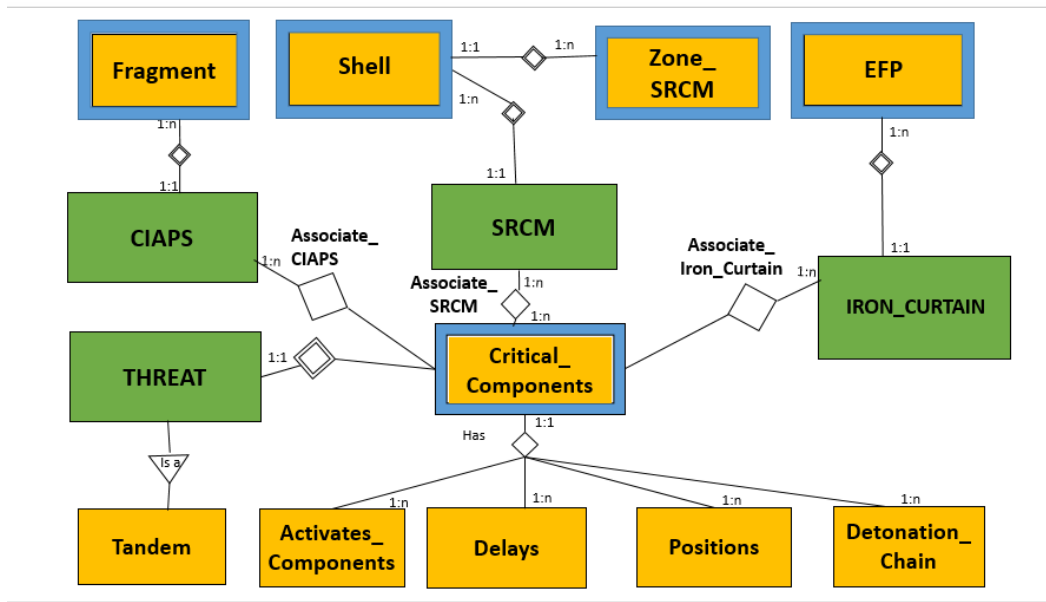


Fig. 32 Consolidated E-R diagram

Based on the consolidated diagram, the design of the tables was conducted in Access, demonstrated in Fig. 33. Readers may find substantial resemblance to previous schemas. However, the total number of tables required for the overall EGM database is significantly reduced to 17 from 31 (10 from CIAPS, 11 from SRCM, and 10 from IC). It implies that plenty of data redundancy can be eliminated with the adoption of the RDB compared to Excel spreadsheets. The decoupling of the CM from the threat information contributes to the outcome.

Another noteworthy feature of the RDB design is scalability. With the current state of the art, given the Army's interest in the implementation of nondevelopmental items, such as Trophy, Iron Curtain, and Iron Fist APSs, for every single threat, 3 additional separate Excel spreadsheets shall be required for the 3 CMs in the EGM analysis. On the battlefield, the Army has encountered numerous and various threats, which are ever changing. The fast expansion of the threat profiles over time will dampen the utilization of Excel data format. With the leverage of the RDB design, an emerging threat may imply merely one additional record in the threat tables. In short, the scalable RDB will greatly facilitate future EGM simulation and analysis, and will be highly adaptive to dynamic mission requirements.

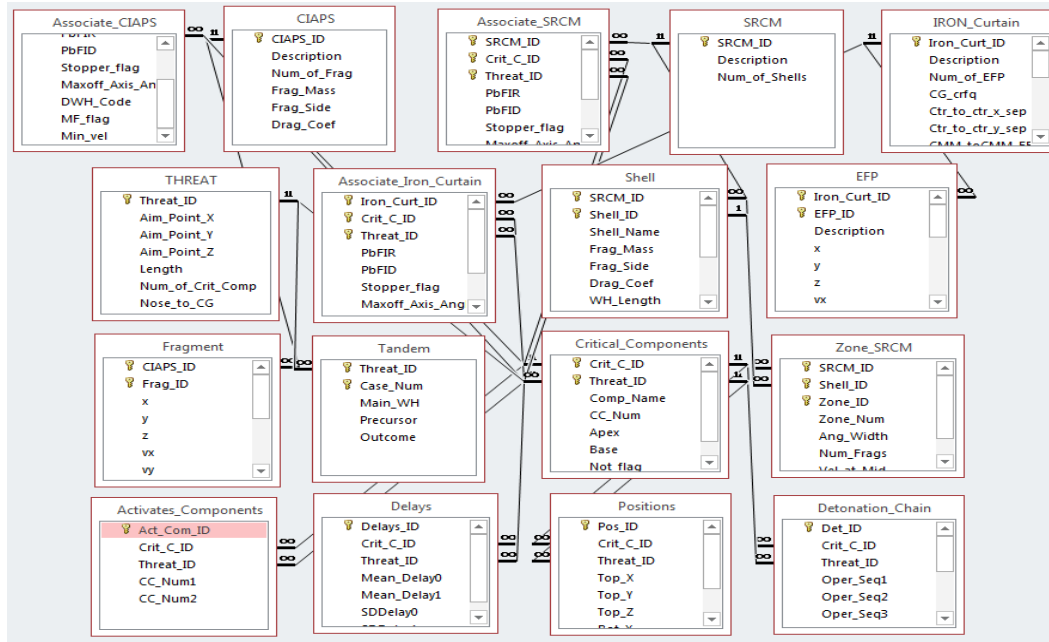


Fig. 33 Consolidation table schema in Access

4.3 Data Population

Due to the enforcement of referential integrity in the database, data population must follow a certain order. At the beginning of the attempt, a problem occurred when entering data. The data in the Associate_SRCM table was populated prior to the SRCM table. Once the input of a record was attempted, an error occurred because the Associate_SRCM table had no SRCM_ID existing in the SRCM table. Afterwards, caution was taken in the data population with respect to the table hierarchy. The input process on the databases was conducted in the following order:

- **CIAPS:** CIAPS, Fragment, THREAT, Associate CIAPS, Critical Components, Tandem, Activates Components, Delays, Positions, Detonation Chain
- **SRCM:** SRCM, Shell, Zone_SRCM, THREAT, Associate SRCM, Critical Components, Tandem, Activates Components, Delays, Positions, Detonation Chain
- **Iron Curtain:** IRON Curtain, EFP, THREAT, Associate Iron Curtain, Critical Components, Tandem, Activates Components, Delays, Positions, Detonation Chain

The process is to transform information from Excel spreadsheets to Access databases. Since both tools were developed by the same vendor, they are fairly compatible. For instance, to populate the 280 records in the Fragments table of the

CIAPS DB, one is not required to manually enter the data one record at a time. Instead, an Excel file extracting the section of the information can be directly imported into the Access table as long as the number of columns in the spreadsheet matches the number of fields in the table and their data types are compatible.

However, most table population is not very straightforward when the fields retrieve information from multiple places in Excel or the information does not exist, such as a primary key or a foreign key. For instance, the Associate_CIAPS table contains fields, such as Crit_C_ID, Threat_ID, PbFIR (Probability Fragment Induced Reaction), PbFID (Probability Induced Detonation), Stopper_flag and Max_of_Axis Angle. Some of the field data originated from the CIAPS table, the Threat table, and the Critical_Components table. Others came from various places in the spreadsheet. As a result, manually entering data would be required in that circumstance. The situation particularly holds for the population of the case matrix, where a 2-D matrix needs to be converted into a 1-D array. The index in the rows is handled with the WH field, the index in the columns is taken care of by the Precursor field, and the case numbers are entered into the Outcome field. Figure 34 shows excerpts of a few CIAPS tables after the data was populated, which include the Associate CIAPS table, the Fragment table, and the Tandem table.

CIAPS_ID	Crit_C_ID	Threat_ID	PbFIR	PbFID	Stopper_flag	Maxoff_Axis_Angle
1	1	1	0	3	0	120
1	2	1	0.5	0	0	120
1	3	1	0	0	0	0
1	5	1	0.3	0.2	0	123
1	6	1	0.5	0.2	0	124
1	6	1	1	0	0	128

Associate_CIAPS Table Excerpt

CIAPS_ID	Frag_ID	x	y	z
1	1	0.01112	0.002222	0.0334555
1	2	0.22222	0.333333	0.444444
1	3	0.555555	0.66666	0.333333
1	4	-0.555555	0.015031	0.0393994
1	5	0.444444	0.4444444	0.3333333
1	6	0.555555	0.33333	0.888888
1	7	0.22222	0.333333	0.0111111
1	8	0.333333	0.111111	0.2222222
1	9	0.3333333	0.444444	0.8888888
1	10	0.6666666	0.0000000	12.000000

Fragment Table Excerpt

Threat_ID	Case_Num	Main_WH	Precursor	Outcome
1	1	0	0	99
1	2	0	1	99
1	3	0	2	99
1	4	0	3	99
1	5	0	4	99
1	6	0	5	99
1	7	0	6	99

Tandem CIAPS Excerpt

Fig. 34 Excerpts of CIAPS in Access

Figures 35 and 36 highlight the excerpts from some of the SRCM and IC tables, respectively. In consideration of the design hierarchy, the SRCM table must be populated first, followed by the Shell table, and then the Zone_SRCM table. The data population of the Iron_Curtain table must be completed prior to that of the EFP table. Similarly, the data must be entered into the Critical_Components table and the SCRM table before the Associate_SRCM table is processed.

The data population of the Associate_Iron_Curtain cannot proceed without the completion of the Iron_Curtain table and the Critical_Components table to meet the requirement of referential integrity.

SRCM_ID	Crit_C_ID	Threat_ID	PbFIR	PbFID	Stopper_flag
1	7	2	0	2	0
1	8	2	4	0	0
1	9	2	0	0	0
1	10	2	4	0	0
1	11	2	0	2	0
1	12	2	3	0	0
1	13	2	3	0	0
1	14	2	0	4	0
1	15	2	0	0	0
1	16	2	3	0	0
1	17	2	0	0	0
1	18	2	0	2	0

Associate_SRCM Table Excerpt

SRCM_ID	Shell_ID	Shell_Name	Frag_Mass	Frag_Side	Drag_Coef	WH_Length
1	1	Shell_0	0.77777	0.555555	0.8888	0.16000
1	2	Shell_1	0.33333	0.666666	0.8888	0.17000
1	3	Shell_2	0.55555	0.444444	0.8888	0.18000

Shell Table Excerpt

SRCM_ID	Shell_ID	Zone_ID	Zone_Num	Ang_Width	Num_Frags
1	1	1	0	1.0	12
1	1	2	1	1.0	34
1	1	3	2	1.0	32
1	1	4	3	1.0	36
1	1	5	4	1.0	36
1	1	6	5	1.0	36
1	1	7	6	1.0	40
1	1	8	7	1.0	44

Zone_SRCM Table Excerpt

Fig. 35 Excerpts of SRCM in Access

Iron_Curt_ID	Crit_C_ID	Threat_ID	PbFIR	PbFID	Stopper_flag
1	20	3	0	1	0
1	21	3	0.8	0	0
1	22	3	0	0	0
1	23	3	0.8	0.8	0
1	24	3	0.8	0.8	0
1	25	3	1	0	0

Associate Iron Curtain Table Excerpt

Iron_Curt_ID	EFP_ID	Description	x	y	z
1	1	self-forging warhead	0.000002	0.0000012	0
1	2	self-forging warhead	0.00004	0.0000010	0.00001
1	3	self-forging warhead	0.0000003	0.0000000	0.00002
1	4	self-forging warhead	0.00003	0.3333	0.00003
1	5	self-forging warhead	0.00001	0.03333	0.00004
1	6	self-forging warhead	0.000001	0.04444	0.00005
1	7	self-forging warhead	0.000010	0.04446	0.00002
1	8	self-forging warhead	0.000020	0.00444	0.00004
1	9	self-forging warhead	0.000060	0.00444	0.00001
1	10	self-forging warhead	0.00008	0.00444	0.00001

EFP Table Excerpt

Threat_ID	Case_Num	Main_WH	Precursor	Outcome
3	289	0	0	99
3	290	0	1	99
3	291	0	2	99
3	292	0	3	99
3	293	0	4	99

Tandem Table Excerpt

Fig. 36 Excerpts of IC in Access

Furthermore, the data populated in the Critical_Components table, shown in Fig. 37, is used to demonstrate the consolidation of tables through the sharing of table fields among all the CMs. In the figure, the Threat_ID of “1” represents the threat in the CIAPS spreadsheet, “2” stands for the threat in the SRCMs, and “3” refers to the threat in the ICs. The field of CC_Number accounts for the critical component number of the threat. The CIAPS has the numbers ranging from 0 to 5 (a total of 6 critical components), the SCRM ranges from 0 to 12 (a total of 13 critical components), and the IC’s threat consists of 6 critical components (ranging from 0 to 5). The corresponding names of the critical components were also populated in the field of Comp_Name where readers may find SW, Ogive, Cone, WH, Booster, S&A, and so on. Other data, such as Apex, Base, Not_Flag, and WH Flag, can be entered associated with the respective critical components.

In summary, all the information in the 3 Excel files (CIAPS, SCRM, and IC) shall be duplicated to the 17 tables as previously outlined. For future experimental data to be populated for EGM analysis, it is apparent that the RDB in Access can do a better job in maintaining data integrity when compared with the Excel files. In general, such enforcement will prevent human errors when entering data.

Crit_C_ID	Threat_ID	Comp_Name	CC_Num	Apex	Base	Not_flag	WH_flag
1	1	SW	0	0	0	0	0
2	1	Ogive	1	0	0	0	0
3	1	Cone	2	0	0	1	0
4	1	WH	3	0.276	0.233	0	2
5	1	Booster	4	0	0	0	0
6	1	S&A	5	0	0	0	0
7	2	SW1	0	0	0	0	0
8	2	Ogive1	1	0	0	0	0
9	2	Cone1	2	0	0	1	0
10	2	WH11	3	0.145	0.12	0	1
11	2	Bstr1	4	0	0	0	0
12	2	S&A1	5	0	0	0	0
13	2	Relay	6	0	0	0	0
14	2	SW2	7	0	0	0	0
15	2	Ogive2	8	0	0	0	0
16	2	Cone2	9	0	0	1	0
17	2	WH2	10	0.48	0.44	0	2
18	2	Bstr2	11	0	0	0	0
19	2	S&A2	12	0	0	0	0
20	3	SW	0	0	0	0	0
21	3	Ogive	1	0	0	0	0
22	3	Cone	2	0	0	1	0
23	3	WH	3	0.27600	0.23300	0	2
24	3	Booster	4	0	0	0	0
25	3	S&A	5	0	0	0	0

Fig. 37 Excerpt of the Critical_Components table in Access

5. Information Retrieval, Analysis, and Results

5.1 Development of C++ Code for Information Retrieval

Upon completion of data population of the query-able Access database, information retrieval from the database to be used for EGM analysis became the next step in the experimental plan of the research. The EGM software module of the SSES survivability suite was developed and written in C++ code using an integrated development environment tool (i.e., Microsoft Visual Studio). The Visual Studio supports various platforms and compilers of numerous programming languages, such as C++. In addition, it is equipped with rich development features and graphical user interfaces that can be used to streamline the processes of coding, compiling, linking, and execution. Therefore, a prototype of a C++ console application was proposed and developed in Visual Studio to enable the communication of the EGM module with the Access database.

Specifically, the console application uses the build-in object linking and embedding (OLE) database library in the namespace of System::Data::OleDb such that several underlying classes can be leveraged. For example, an object of the OleDbConnection class was constructed to establish a connection to the Access database. An instance of the OleDbCommand class was initialized with the connection object and an SQL text string. Subsequently, the console application invoked the ExecuteReader() function of the OleDbCommand class, which executed the SQL statement and returned the query result to a reader object that was constructed from the OleDbDataReader class. The reader object provides a

way of reading a forward-only stream of data rows from a data source, such as the Access database. A copy of the C++ source code is shown in Table 2.

To make the Access database file visible to the code, one must select the option of “Access Data File” in the “Choose Data Source” dialog box in Visual Studio, and then click “OK”. Subsequently, in the “Add Connection” dialog box, change the data source by clicking on “Change”, point to the folder where the “SSES_EndGameModelDB1” Access file is located, select the file and then click “OK”. The absolute file path of the Access file shall also be specified in the parameter of the `OleDbConnection()` when it is constructed, along with the connection provider information (i.e., `Microsoft.ACE.OLEDB.12.0`, as shown in Fig. 38).¹⁰

Generally speaking, SQL has been widely used to communicate with a database. According to the American National Standards Institute, it is the standard language for relational database management systems. SQL statements can be adopted to perform tasks such as update data on a database, or retrieve data from a database, which is applicable to the task of concern. It can be observed that an SQL query was passed through the argument of the `main()` routine from the Console command prompt. To enable continuously multiple entries of the SQL queries and their executions, the `Console::ReadLine()` and the `OleDbCommand::ExecuteReader()` functions were invoked inside a “Do” loop. This design greatly streamlined the process of code verification and validation of the data retrieval.

```

using namespace System;
using namespace System::Data::OleDb;

int main(array<System::String ^> ^args)
{
    String^ sqlstr;
    String^ temp;
    OleDbConnection^ conn = nullptr;
    OleDbCommand^ cmd = nullptr;
    OleDbDataReader^ reader;

    conn = gcnew OleDbConnection ("PROVIDER=Microsoft.ACE.OLEDB.12.0;Data
Source=L:\\2017 Summer Students\\Akole\\SSES_EndGameModel_DB1.accdb");
    conn->Open ();

    int counter = 0;
    try
    {
        Console::WriteLine ("Welcome to the SSES End-Game Relational
Database");
        Console::WriteLine ("Please enter a query:");
        do{
            if(counter == 0) sqlstr = Console::ReadLine();

            cmd = gcnew OleDbCommand (sqlstr, conn);

            reader = cmd->ExecuteReader
(System::Data::CommandBehavior::CloseConnection);
            String^ Sep = gcnew String ('*', 60);

            while (reader->Read ())
            {
                temp = "CompName " + "CritCNum " + "Apex " + "
Base " + "NotFlag" + " WHFlag";

                Console::WriteLine (temp);
                temp = "";
                for(int i=0; i<reader->FieldCount; i++)
                    temp += reader[i] + "\\t ";
                Console::WriteLine (temp);

                Console::WriteLine (Sep);
                temp = "";
            }

            counter++;
            Console::WriteLine ("Please enter another SQL query: ");
            sqlstr = Console::ReadLine();

        } while (sqlstr);
    }
    catch (Exception^ ex)
    {
    }
    return 0;
}

```

Fig. 38 C++ code for informational retrieval of Access database

5.2 SQL Analysis and Execution

To make the C++ Console application more user friendly, a heading statement, “Welcome to the SSES End-Game Relational Database”, was written out, followed by another statement, “Please enter a query:”, in a new line to prompt for user’s input. As shown in Fig. 39, a simple SQL query “Select Comp_Name, CC_Num, Apex, Base, Not_flag, WH_flag from Critical_Components” was issued in the command prompt. Once entered, the query results were returned and shown in the output window. For brevity, the figure shows only a portion of the results.

```
Welcome to the End Game Relational Database
Please enter a query:
Select Comp_Name, CC_Num, Apex, Base, Not_flag, WH_flag from Critical_Components

CompName CritCNum Apex Base NotFlag WHFlag
SW 0 0 0 0 0
*****
CompName CritCNum Apex Base NotFlag WHFlag
Ogive 1 0 0 0 0
*****
CompName CritCNum Apex Base NotFlag WHFlag
Cone 2 0 0 1 0
*****
CompName CritCNum Apex Base NotFlag WHFlag
MH 3 0.276 0.233 0 2
*****
CompName CritCNum Apex Base NotFlag WHFlag
Booster 4 0 0 0 0
*****
CompName CritCNum Apex Base NotFlag WHFlag
S&A 5 0 0 0 0
*****
CompName CritCNum Apex Base NotFlag WHFlag
SW1 0 0 0 0 0
*****
```

Fig. 38 Descriptive screenshots

In general, given a Table T, a typical SQL statement “Select * from T” will result in all the elements of all the rows of the table being shown. With the same table, the query “Select C0, C1 from T” will result in the elements from the columns C0 and C1 of all the rows of the table being shown. Therefore, the issued SQL query selected the fields or attributes of “Comp Name”, “CC Num”, “Apex”, “Base”, “Not Flag” and “WH Flag” from the Critical_Components table.¹¹ It was verified that the C++ code performed in the manner in which it was designed. The communications between the Access database and the code took place. In addition, the returned texts and numbers were compared against the records in the Critical_Components table, and the results were validated.

As mentioned previously, after the result of a query was returned to the output window, the C++ code would prompt for another query infinitely until the user hit the “Return” key twice. By leveraging the feature, a large number of SQL queries were thoroughly tried out in an attempt to ensure that all of the information in the

EGM Excel files could be retrieved properly with the SQL statements. In the Appendix, a series of SQL queries are provided for readers' reference.

5.3 Code Modification/Replacement Attempt

The last step in the experimental plan was to substitute the Excel files used for the SSES EGM analysis with the Access DB where the threat and the CM information are decoupled. To achieve this goal, the data read function to get data from the Excel files in the EGM software module must be replaced with the C++/SQL code. In addition, the code must be integrated into the SSES solution framework for end-to-end simulation runs in batch mode.

Significant efforts were made in the enhancement and modification of the C++ code for the integration. After time-consuming debugging, a major issue was identified. The C++ console application that runs under the control of common language runtime (CLR) is known as managed code. The use of OleDbConnection class depends on the CLR environment. The code that does not run under the CLR is known as native code. SSES software suite is one example of the native code. Some features are available only to the managed programming model or to the native programming model. In addition, the representations of primitive data type and data structures differ substantially between the managed code and the native code. As a result, once the C++ code was merged with the SSES, the execution of the SQL queries through the OleDbCommand could not be accomplished. Due to the complexity of the interoperations between the native and the managed C++ code, the replacement of the Excel data read function could not be completely implemented given a very limited time frame for the research. Future investigation shall be warranted to seek a solution.

6. Summary and Conclusion

The research project of scalable database design was initiated in support of SSES modularization efforts with respect to 4 major software components: Threat, Countermeasure, Sensor, and Vehicle. The current state-of-the-art data management for SSES EGM relies on Excel data structure, where both the threat and the CM information resides in a single file. Due to rapidly changing threats and evolutionary CM technologies, the coexistence of the information along with a few mutually dependent fields in a single location hinders the expansion of the EGM data files. Consequently, one of the research objectives was to decouple the information so that the Threat and the CM modules can be independently developed.

In the investigation, a hypothesis was formulated that a RDB is more adaptable to proliferation of threats than Excel files for EGM analysis. The data structure and the data format of 3 existing Excel data files, one for each of the APS cases—CIAPS, SRCM, and Iron Curtain—were evaluated. Following a typical RDB design process, the authors conducted requirement analysis, gathered and organized data, constructed tables, specified primary keys, identified cardinality, created relationships among tables, and refined and normalized the design. Specifically, entity relationship diagrams, relational schemas, and table structures were developed for all 3 cases.

Subsequently, the research hypothesis was validated through a series of experimental plans. With the use of Excel files, the effort to incorporate additional threats is augmented by the number of CMs, a major drawback in terms of data scalability. The RDB resolved the issue, yielded a solution, and met dynamic mission requirements. In addition to the scalable features, the total number of tables in the Access database was reduced to 17 from 31 as a result of data consolidation. Further, the data population of the tables demonstrated the decoupling of threats from CM information. A prototype of C++ code with embedded SQL queries was developed for information retrieval, which confirmed the feasibility of the RDB for EGM analysis.

Finally, it should be noted that one of the SSES design features is portability. Microsoft Access is Windows platform-specific, which was chosen simply because it was a part of ARL's standard computer image. No additional approval was required. Future focus will be on open source database tools with no proprietary technology. However, the RDB concept, the relational schemas, and the table design that have been accomplished in this study shall be applicable to forthcoming implementation.

7. References

1. Collaborative point paper on active protective systems. North American Technology and Industrial Based Organization. Prepared by DCMA Industrial Technology Center; 2006 Oct.
2. Chen M, Graham MJ. Transition and implementation of survivability suite engineering simulation (SSES) for active protective systems. Aberdeen Proving Ground (MD): Army Research Laboratory (US); 2017 July. Report No.: ARL-TR-8048.
3. Chen M, Graham MJ, McNeir M. Preliminary development of survivability suite engineering simulation (SSES): software architecture and design. Aberdeen Proving Ground (MD): Army Research Laboratory (US); 2016 Oct. Report No.: ARL-TR-7855.
4. Analyst's manual for the end-game model, v4.7.7. Columbia (MD): Gleason Research Associates, Incorporated (GRA); 2014 July 31. GRA Report No.: 001A-034C-03, GRA Task 001A-034.
5. Bentley M, Gleason T. Fragment end-game model. Aberdeen Proving Ground (MD): Army Research Laboratory (US); 2009 Apr 30. Gleason Research Associates, Incorporated (GRA) Report No.: 2009-0001A-008.
6. User's manual for the end-game model, v4.8.8. Columbia (MD): Gleason Research Associates, Incorporated (GRA); 2016 Apr 29. GRA Report No.: 012A-001C-02, GRA Task 012A-001.
7. Entity relationship diagram. SmartDraw, LLC [accessed 2017 Aug 11]. <https://www.smartdraw.com/entity-relationship-diagram>.
8. How to make a database using MS Access. WikiHow [accessed 2017 Sep 20]. <https://www.wikihow.com/Make-a-Database-Using-MS-Access>.
9. Microsoft Visual Studio. Wikipedia [accessed 2017 Sep 20]. https://en.wikipedia.org/wiki/Microsoft_Visual_Studio.
10. How to: create connections to Access Database. Microsoft [accessed 2017 Sep 20]. [https://msdn.microsoft.com/en-us/library/5ybdbtte\(v=vs.90\).aspx](https://msdn.microsoft.com/en-us/library/5ybdbtte(v=vs.90).aspx).
11. SQL – SELECT Query. TutorialsPoint, Simply Easy Learning [accessed 2017 Sep 20]. <https://www.tutorialspoint.com/sql/sql-select-query.htm>.

Appendix. List of Structured Query Language (SQL) Queries

The SQL queries were issued following the prompt “Please enter another SQL query:” on the output window, which was an attempt to validate the retrieved information.

1. “Select from * CIAPS” Displays all the information in table CIAPS.
2. “Select from * Fragment” Displays all the information in table Fragment.
3. “Select from * THREAT” Displays all the information in table THREAT.
4. “Select from * Associate_CIAPS” Displays all the information in table Associate CIAPS.
5. “Select from * Critical_Components” Displays all the information in table Critical_Components.
6. “Select from * Tandem” Displays all the information in table Tandem.
7. “Select from * Activates” Displays all the information in table Activates.
8. “Select from * Delays” Displays all the information in table Delays.
9. “Select from * Positions” Displays all the information in table Positions.
10. “Select from * Detonation_Chain” Displays all the information in table Detonation_Chain.

Methods for selecting to display certain information from requested tables:

CIAPS

1. “Select CIAPS_ID from CIAPS” Displays 1
2. “Select Num_of_Frag from CIAPS” Displays 280
3. “Select Frag_Mass from CIAPS” Displays 0.001232
4. “Select Frag_Side from CIAPS” Displays 0.004
5. “Select Drag_Coef from CIAPS” Displays 0.62
6. “Select Description from CIAPS” Displays short range countermeasure

Fragment

1. “Select Frag_ID from Fragment” Displays all 280 frag IDs.
2. “Select CIAPS from Fragment” Displays the number “1” 280 times.
3. “Select x from Fragment” Displays all the values of x in the fragment table.
4. “Select y from Fragment” Displays all the values of y in the fragment table.
5. “Select z from Fragment” Displays all the values of z in the fragment table.
6. “Select vx from Fragment” Displays all the values of vx in the fragment table.
7. “Select vy from Fragment” Displays all the values of vy in the fragment table.

8. "Select vz from Fragment" Displays all the values of vz in the fragment table.
9. "Select SD_dir from Fragment" Displays all the values of SD_dir in the fragment table.
10. "Select SD_v from Fragment" Displays all the values of SD_v in the fragment table.
11. "Select yaw from Fragment" Displays all the values of yaw in the fragment table.
12. "Select phid from Fragment" Displays all the values of phid in the fragment table.
13. "Select thtd from Fragment" Displays all the values of thtd in the fragment table.
14. "Select phsid from Fragment" Displays all the values of phsid in the fragment table.
15. "Select * from Fragment where Frag_ID=280" Displays all the items in Frag_ID 280 only.

THREAT

1. "Select Threat_ID from THREAT" Displays the Threat_ID in the THREAT table.
2. "Select Aim_Point_X from THREAT" Displays the Aim_Point_X, which is ".276" in the THREAT table.
3. "Select Aim_Point_Y from THREAT" Displays the Aim_Point_Y, which is "0" in the THREAT table.
4. "Select Aim_Point_Z from THREAT" Displays the Aim_Point_Z, which is "0" in the THREAT table.
5. "Select Length from THREAT" Displays the Length, which is "1.3" in the THREAT table.
6. "Select Num_of_Crit_Comp from THREAT" Displays the Num_of_Crit_Comp, which is "6" in the THREAT table.
7. "Select Nose_to_CG from THREAT" Displays the Nose_to_CG, which is "0.2" in the THREAT table.

Associate CIAPS

1. "Select CIAPS_ID from Associate" Displays the CIAPS_ID in the Associate table.
2. "Select Crit_C_ID from Associate" Displays the Crit_C_ID (s) in the Associate table.
3. "Select Threat_ID from Associate" Displays the Threat_ID in the Associate table.
4. "Select PbFIR from Associate" Displays the PbFIR in the Associate table.

5. "Select PbFID from Associate" Displays the PbFID in the Associate table.
6. "Select Stopper_flg from Associate" Displays the Stopper_flg in the Associate table.
7. "Select Min_flg from Associate" Displays the Min_flg in the Associate table.
8. "Select Min_vel from Associate" Displays the Min_vel in the Associate table.
9. "Select Maxoff_Axis_Angle from Associate" Displays the Maxoff_Axis_Angle in the Associate table.
10. "Select DWH_Code from Associate" Displays the DWH_Code in the Associate table.
11. "Select *from Associate where Crit_C_ID =1" Display all data in that ID field.

Critical_Components

1. "Select Crit_C_ID from Critical_Components" Displays the Crit_C_ID in the Critical_Components table.
2. "Select Threat_ID from Critical_Components" Displays the Threat_ID in the Critical_Components table.
3. "Select Comp_Name from Critical_Components" Displays the Comp_Name in the Critical_Components table.
4. "Select CC_Num from Critical_Components" Displays the CC_Num in the Critical_Components table.
5. "Select Apex from Critical_Components" Displays the Apex in the Critical_Components table.
6. "Select Base from Critical_Components" Displays the Base in the Critical_Components table.
7. "Select Not_flg from Critical_Components" Displays the Not_flg in the Critical_Components table.
8. "Select WH_flg from Critical_Components" Displays the WH_Flag in the Critical_Components table.
9. "Select * from Critical_Components where Crit_C_ID= 6" Displays all the information Associated with that ID.

Tandem

1. "Select Threat_ID from Tandem" Displays the Threat_ID in the Tandem table.
2. "Select Case_Num from Tandem" Displays the Case_Num in the Tandem table.
3. "Select Main_WH from Tandem" Displays the Main_WH in the Tandem table.

4. "Select Outcome from Tandem" Displays the Outcome in the Tandem table.
5. "Select Precursor from Tandem" Displays the Precursor in the Tandem table.
6. "Select * from Tandem where Case_Num=5" Displays all information associated with that case number only.

Activates_Components

1. "Select Act_Com_ID from Activates_Components" Displays the Act_Com_ID in the Activates_Components table.
2. "Select Crit_C_ID from Activates_Components" Displays the Crit_C_ID in the Activates_Components table.
3. "Select Threat_ID from Activates_Components" Displays the Threat_ID in the Activates_Components table.
4. "Select CC_Num1 from Activates_Components" Displays the CC_Num1 in the Activates_Components table
5. "Select CC_Num2 from Activates_Components" Displays the CC_Num2 in the Activates_Components table
6. "Select * from Act_Com_ID where Crit_C_ID=2" Displays all information in that table associated with that ID number.

Delays

1. "Select Delays_ID from Delays" Displays the Delays_ID in the Delay table
2. "Select Crit_C_ID from Delays" Displays the Crit_C_ID in the Delay table
3. "Select Threat_ID from Delays" Displays the Threat_ID in the Delay table
4. "Select Mean_Delay0 from Delays" Displays the Mean_Delay1 in the Delay table
5. "Select Mean_Delay1 from Delays" Displays the Mean_Delay2 in the Delay table
6. "Select SDDelay0 from Delays" Displays the SDDelay0 in the Delay table
7. "Select SDDelay1 from Delays" Displays the SDDelay1 in the Delay table
8. "Select * from Delays where Delays_ID=5" Displays the information in that table that is associated with that ID

Positions

1. "Select Pos_ID from Positions" Displays the Pos_ID in the Positions table

2. "Select Crit_C_ID from Positions" Displays the Crit_C_ID in the Positions table
3. "Select Threat_ID from Positions" Displays the Threat_ID in the Positions table
4. "Select Top_X from Positions" Displays the Top_X in the Positions table
5. "Select Top_Y from Positions" Displays the Top_Y in the Positions table
6. "Select Top_Z from Positions" Displays the Top_Z in the Positions table
7. "Select Bot_X from Positions" Displays the Bot_X in the Positions table
8. "Select Bot_Y from Positions" Displays the Bot_Y in the Positions table
9. "Select Bot_Z from Positions" Displays the Bot_Z in the Positions table
10. "Select * Positions where Pos_ID =4" Displays the information associated with that ID.

Detonation_Chain

1. "Select Det_ID from Detonation_Chain" Displays the Det_ID in the Detonation_Chain table
2. "Select Crit_C_ID from Detonation_Chain" Displays the Crit_C_ID in the Detonation_Chain table
3. "Select Threat_ID from Detonation_Chain" Displays the Threat_ID in the Detonation_Chain table
4. "Select Oper_Seq1 from Detonation_Chain" Displays the Oper_Seq1 in the Detonation_Chain table
5. "Select Oper_Seq2 from Detonation_Chain" Displays the Oper_Seq2 in the Detonation_Chain table
6. "Select Oper_Seq3 from Detonation_Chain" Displays the Oper_Seq3 in the Detonation_Chain table
7. "Select Oper_Seq4 from Detonation_Chain" Displays the Oper_Seq4 in the Detonation_Chain table
8. "Select Oper_Seq5 from Detonation_Chain" Displays the Oper_Seq5 in the Detonation_Chain table
9. "Select Oper_Seq6 from Detonation_Chain" Displays the Oper_Seq6 in the Detonation_Chain table
10. "Select Oper_Seq7 from Detonation_Chain" Displays the Oper_Seq7 in the Detonation_Chain table
11. "Select Oper_Seq8 from Detonation_Chain" Displays the Oper_Seq8 in the Detonation_Chain table

List of Symbols, Abbreviations, and Acronyms

1-D	1-dimensional
2-D	2-dimensional
APS	active protection system
ARL	US Army Research Library
ATGMs	antitank guided missiles
BISODJ	Built-In Stand-off with Damaged Jet
BISONJ	Built-In Stand-off with Normal Jet
CIAPS	Close-In APS
CLR	common language runtime
CMs	countermeasures
DUD	Dud
DWH	Dismembered Warhead
EGM	End-Game Model
EFP	explosively formed projectiles
EINJ	Early Initiation with Normal Jet
EIDJ	Early Initiation with Damaged Jet
E-R	entity relationship
FID	Fragment Induced Detonation
FIR	Fragment Induced Reaction
IC	Iron Curtain
KE	kinetic energy
MAPS	Modular Active Protective System
OLE DB	object linking and embedding database
RDB	relational database
RPG	rocket-propelled grenade
SLAD	Survivability/Lethality Analysis Directorate

SQL	Structured Query Language
SRCM	Short Range CM
SSES	Survivability Suite Engineering Simulation

1 (PDF)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA	RDRL WMP F N GNIAZDOWSKI RDRL WMP G R EHLERS RDRL WML N TRIVEDI RDRL WML A W OBERLE RDRL WML B N TRIVEDI RDRL WML C S AUBERT RDRL WML D D BEYER RDRL WML E P WEINACHT RDRL WML F M ILG RDRL WML G J SOUTH RDRL WML H J NEWILL RDRL WMM M VANLANDINGHAM RDRL-WMM-D R CARTER B CHEESEMAN RDRL SLB S M PERRY J AUTEN J SHINDELL
2 (PDF)	DIR ARL IMAL HRA RECORDS MGMT RDRL DCL TECH LIB	
1 (PDF)	GOVT PRINTG OFC A MALHOTRA	
2 (PDF)	RDECOM TARDEC M ARCHER D KHAN	
1 (PDF)	RDECOM AMRDEC C WILLIAMS	
41 (PDF)	DIR ARL RDRL WM J ZABINSKI B FORCH S SCHOENFELD A RAWLETT RDRL WMP D LYON T VONG D HOGGE RDRL WMP A S BILYK M CHEN R YAGER M GRAHAM M MCNEIR C WOLFE G THOMSON M COPPINGER L VANDERHOEF J FLENIKEN J CAZAMIAS W UHLIG RDRL WMP B C HOPPEL S SATAPATHY RDRL WMP C T BJERKE RDRL WMP D A BARD M KEELE J RUNYEON RDRL WMP E D HACKBARTH P BARTKOWSKI P SWOBODA	

INTENTIONALLY LEFT BLANK.