5-10-2005

Executable Model Development from Architectural Description with Application to the Time Sensitive Target Problem

Luis M. Diaz Rodriguez

EXECUTABLE MODEL DEVELOPMENT FROM ARCHITECTURAL DESCRIPTION WITH APPLICATION TO THE TIME SENSITIVE TARGET PROBLEM

THESIS

Luis M. Díaz Rodríguez, Aerospace Engineer, USAF

AFIT/GSE/ENY/05-M04

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GSE/ENY/05-M04

EXECUTABLE MODEL DEVELOPMENT FROM ARCHITECTURAL DESCRIPTION WITH APPLICATION TO THE TIME SENSITIVE TARGET PROBLEM

THESIS

Presented to the Faculty

Department of Aeronautics and Astronautics

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Systems Engineering

Luis M. Díaz Rodríguez, BS

Aerospace Engineer, USAF

April 2005

AFIT/GSE/ENY/05-M04

EXECUTABLE MODEL DEVELOPMENT FROM ARCHITECTURAL DESCRIPTION WITH APPLICATION TO THE TIME SENSITIVE TARGET PROBLEM

Luis M. Díaz Rodríguez, BS

Aerospace Engineer, USAF

Approved:

/Signed/	10 May 2005
Dr. David R. Jacques, USAF (Chairman)	Date
/Signed/	10 May 2005
John Colombi, PhD, Lt Col, USAF (Member)	Date
/Signed/	10 May 2005
Maj Jörg Walter, PhD, Maj, USAF (Member)	Date

Acknowledgments

I would like to thank my advisor, Dr. David Jacques, for his guidance and patience throughout this effort. Comments and suggestion for improvement by Dr. Jacques, as well as, Lt Col John Colombi and Maj Jörg Walter contributed greatly to the quality of this document and the research in general. I would also like to thank Mrs. Lynn Curtis for helping in everything she could.

I would also like to thank the Vitech Corporation for making available a full professional version of their software for purpose of this investigation and the technical help and training provided by Vitech's staff and president.

Finally, I would like to thank my family. Their encouragement and support helped me keep things on track.

Luis M. Díaz Rodríguez

To my family

Table of Contents

Page

List of Figures

Page

Page

Page

Page

List of Tables

Abstract

As the Department of Defense (DoD) moves to a capability based approach for system definition and development, it has become necessary to evaluate System of System (SoS) characteristics for architectures. Desired capabilities are often achievable only through seamless integration of many different systems. As the classical system engineering approaches were not focused to effectively handle the complexity of SoS level concepts, an architecture-driven approach has emerged as a way of defining and evaluating these new concepts. While the use of architectures for documenting and tracking interfaces and interoperability concerns is generally understood, architectural analysis and the use of executable models for evaluation of architectures remain an open area of research. With this purpose in mind, this thesis applies architectural-based analysis to the proposed 2012 Time Sensitive Effect Operation (TSEO2012) scenario. This scenario becomes the baseline for architectural analysis, and an excursion from this baseline will add a Weapon Born Battle Damage Assessment (WBBDA) capability. By creating an executable model, the two architectural designs can be compared. The addition of a WBBDA capability to the TSEO architecture improves the efficiency of the time sensitive target (TST) operations by shortening the decision cycle for target re-strike. While this effort was successful in obtaining an executable model directly from the architectural description, it highlights the importance of having sufficient specific elements and correct information contained in the architecture products.

EXECUTABLE MODEL DEVELOPMENT FROM ARCHITECTURAL DESCRIPTION WITH APPLICATION TO THE TIME SENSITIVE TARGET PROBLEM

I. Introduction

Problem Overview (Motivation)

To obtain the desired capabilities from a new generation of weapon systems, the United States Air Force (USAF) is migrating to a capabilities based process. The implementation of this process requires a new approach to perform and implement system engineering; the classical system engineering approach does not adequately handle the new System of System (SoS) [D&S04; 03]. To meet these complex system integration challenges, the USAF like the other services has adopted an architecture-driven design approach based on the Department of Defense Architecture Framework (DoDAF).

To support this architecture driven process, the following investigation is concentrated on direct migration from a static architecture to an executable model of that architecture and the subsequent validation and/or evaluation. The desired end product (executable model and subsequent analysis) is viewed as an essential part in the validation of system design and the achievement of the desired capability. [Levis03].

There are many theories of how the executable is an integral part in the validation process of an architecture, but no real practical experience has been documented thus far. It is the intent of this investigation to get practical knowledge of the intricacies of developing and using an executable model of an architecture.

Arguably, an architecture-driven process is the most appropriate way to address the capabilities-based requirement and system development approach. Heavy reliance on architectures motivates the question, “How do I know the architecture design will accomplish the desired results?” The answer is the need to model the architecture to a sufficient level of detail in order to evaluate its behavior and performance. A simulation should uncover any problems in the architecture and support early trade study development. Results obtained in the simulations can be used early in the systems design process where changes to the architecture can be performed with minimal impact to the timeline and project cost.

Executable architectures will be an essential tool to help filter undesirable design conflicts within an architecture. They help validate the architecture as correct and complete products are needed for development of an executable model. Incomplete architectural products will be detected in the development of the executable model as essential information will be missing. By finding errors and performing trade studies early during system design, the DoD can develop an effective family of systems (FoS) that can deliver the desired capabilities on time and on budget. In the current atmosphere of budget cuts and tight schedules, USAF can not afford systems that are unable to deliver the desired capabilities within some overall joint context.

Weapon Born Battle Damage Assessment (WBBDA), the capability of attaining post-strike intelligence by integrating sensors with the munitions, has been documented for several years, but its ultimate utility to the warfighter has not yet been established. One of the reasons for this is the difficulty in determining the role and utility for

WBBDA in the battlespace without actual fielding of the weapon system. Without a utility analysis, the requirement generation for WBBDA is plagued with uncertainty. This type of capability assessment is one of the reasons the USAF is migrating to an architecture-driven process. Capabilities only attainable through SoS level interoperability can be validated through architectural analysis. Therefore, it is natural to apply this type of analysis using an executable model to the WBBDA concept. The WBBDA concept represents a good example of a system of systems problem, where the interoperability of the entire system is an essential part of the desired capability.

Consistency needs to be maintained through the generation of the executable model. To ensure consistency we need a system design tool capable of maintaining an integrated dictionary, one that can relate the executable model back to the architecture. The CORE™ system design tool was chosen for the development of the executable model and the integration of WBBDA. Preference for CORE™ over other alternatives was due mainly to its ability to maintain concordance between its many views, products, and integrated database. CORE™ was found to be more user friendly than other tools, with easy to navigate control tabs.

Related Work

The following studies and research show how architectures using the DoDAF can support the requirements and acquisition communities.

Dickerson and Soules published a study titled “Using Architectures for Research, Development, and Acquisition” [D&S04]. The goal of the study was to show how architectures could be used to enable a capabilities-based approach for research, development, and acquisition of DoD systems that must interoperate with each other to conduct military operations [D&S04; preface]. Their investigation used pilot projects to explore the utility of the architecture methodology applied to a complex capability based SoS. It was discovered that DoDAF products were not designed to analyze SoS, but they can be and have been adapted to support this function [D&S04; 148]. Therefore, architectures can be used as tools to develop integrated solutions for achieving desired mission capabilities. Dickerson and Soules (D&S) broke down the Architecture Framework products into five distinct groups [D&S04; 11] shown in Table 1.

The present investigation is primarily focused in the Architecture Performance and Behavior group. This group supports trade studies and system engineering decisions; it also happens to be the most labor intensive of the groups. The authors acknowledge the executable model as a new product required for both validation and analysis. The executable model is an essential part in the development and execution of trade studies. However, Dickerson and Soules don’t give much insight into the executable model. They only propose it be used in conjunction with the other three products (OV-6c, SV-7, and SV-10) to observe the behavior and the performance of the architecture.

Table 1: Architecture Framework product grouping [D&S04; 11]

Product Groups	Architecture Products	Description	Purpose
Operational Concept	OV-1	High-Level Operational Concept Graphic	Provide the foundation for systems development and facilitate communication by providing context, orientation, and focus.
	OV-2	Operational Node Connectivity Description	
	OV-4	Organizational Relationships Chart	
	OV-5	Operational Activity Model	
System Functional Mapping	SV-3	System Matrix	Provide the linkage and traceability of capabilities and requirements flow-down between the operational and physical views.
	SV-4	Systems Functionality Description	
	SV-5	Operational Activity to Systems Function Traceability Matrix	
System Interface Mapping	OV-2	Operational Node Connectivity Description	Check that the appropriate standards been applied. Check that the levels of interoperability have been properly aligned so that the individual systems in the FoS can be expected to interoperate with each other successfully to enable functionality.
	OV-3	Operational Information Exchange Matrix	
	SV-1	System Interface Description	
	SV-2	Systems Communications Description	
	TV-1	Technical Standard Profile	
	SV-6	System Data Exchange Matrix	
Architecture Performance and Behavior	OV-6c	Operational Event/Trace Description	Necessary to support trade studies and system selection decisions.
	SV-7	Systems Performance Parameters Matrix	
	SV-10	System Activity Sequence and Timing Description	
	New product	Executable Model	
Acquisition Planning	SV-9	Systems Technology Forecast	Provide a description of the evolution and acquisition of the system improvements for the FoS that are traceable to mission capability requirements.
	TV-2	Technical Standards Forecast	
	SV-8	Systems Evolution Description	
	CV-6	Capability Evolution Description	

According to Dickerson & Soules, the architecture cannot be validated until it can be executed [D&S04; 14]. However, the sole presence of an executable model does not imply validation.

Levis asserts that the DoDAF products along with the integrated dictionary contain all of the information needed to describe an architecture [Levis03]. The information contained in the products is necessary, but not sufficient, for evaluating the architecture. For an effective evaluation, scenarios, key threads, and metrics are required. These opposing view of evaluation are integrated through the development of the executable model. Correctly developed executable models can be implemented as a model and simulation (M&S) tool to support requirements validation and acquisition decisions. Levis proposes the use of Colored Petri Nets (CPN) as a possible basis for the generation of the executable model, and provides guidance on how to analyze an architecture, once the executable model is available. The analysis process is divided into layers; with each subsequent layer moving from abstract/general components to more concrete/specific components. Table 2 depicts the layers of Levis' analysis process [Levis04; 34-38].

In Levis' view the optimum solution for solving the architectural analysis problem is the development of a new M&S tool specifically tailored for architecture evaluation [Levis04; ASC-39]. For Levis, the executable model is the mathematical model that enables simulation and the application of analysis [Levis04; 15]. Thus, he has promoted CPN for their mathematical robustness.

Table 2: Layered analysis [Levis04; 34-38]

Layer 1 - Is the architecture logically correct?	
	Is the architecture a correct implementation of the CONOPS? Does the CONOPS work or are there logical inconsistencies? Analytical tools and simulation are appropiate. A discrete event dynamical executable model is essential for this analysis.
Layer 2 - Does the architecture exhibit the desired behavior?	
	Are the desired behaviors in the operational view? Analytical/algorithmic approaches as well as Modeling and Simulation approaches are appropriate.
Layer 3 - Do instantiations of this architecture exhibit the desired performance characteristics?	
	To evaluate performance, system characteristics need to be included. May cross hard-to-define architecture and system design boundary. Requires the use of discrete dynamical system models and time-driven models. Need to resolve the challenge of interconnecting time driven and event driven models.
Layer 4 - Do systems built in conformance to this architecture provide the desired capability?	
	Need to articulate capabilities and express them in measurable terms. Formal construction of key threads.
Layer 5 - Analysis of alternatives	
	The desired end capability of comparing two distinct architectures that are designed under the same CONOPS. What metrics are appropriate for an impartial comparison? How do I trace to architectural issuess the differences in performance

While Levis proposed the development of a new M&S tool for evaluation, Zinn explored the possibility of migrating information from a system's architectural description (products) into an existing M&S tool. The desired end result is to create a collaborative and quantitative infrastructure between the system acquisition community and the operational warfighting commands. His proposed approach may be more palatable to the existing M&S community in that it uses existing, proven tools and processes. Zinn investigated whether or not the DoDAF products, unmodified, contained all the data required for population of an agent based model with subsequent analysis. Through his study he found out that fully implemented DoDAF products should provide most or all of the information necessary to model a weapon system in an agent based simulation [ZinnB04], thus avoiding a redefinition of the system. He also identified eight important products to describe a concept system for agent based modeling (OV-1, OV-5, OV-6a, SV-7, SV-2, SV-6 or OV-3, SV-1, and OV-4). The products SV-7, OV-4 and SV-1 provide general information to the simulation endeavor, Figure 1. The second set

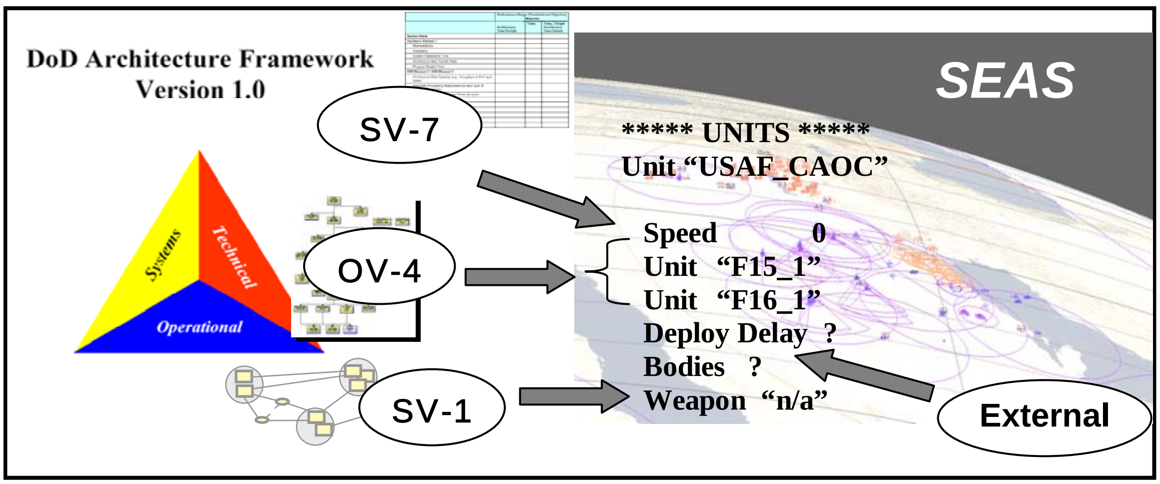

Figure 1: Mapping general attributes from DoDAF [ZinnB04; 59]

of products, SV-7 and SV-6 or OV-3, provide specific communication descriptions as seen on Figure 2. The last set of products, OV-1, OV-5, OV-6a, and SV-7, provide the

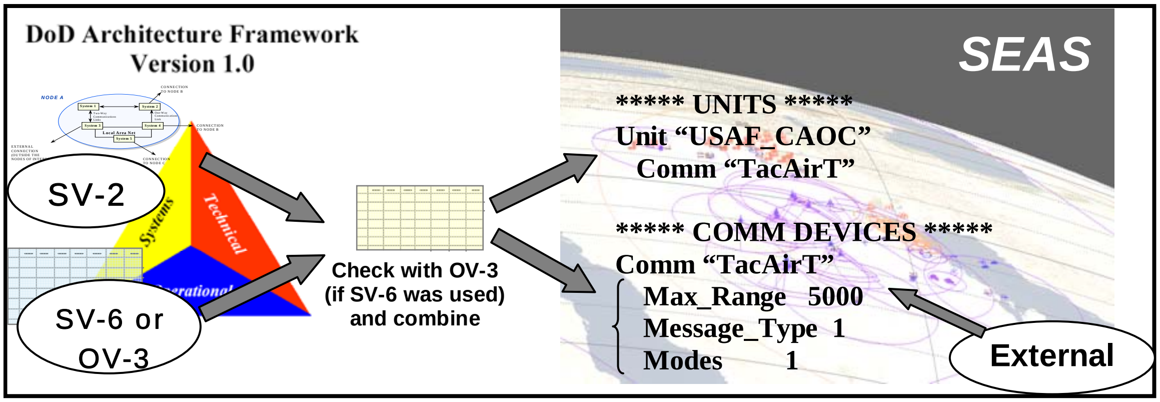

Figure 2: Mapping communications from DoDAF [ZinnB04; 62]

ability to make logical decisions regarding behavior, Figure 3. He concluded that OV-5 and OV-6a were the most important products, as they provide most of the basis for the logical code of the executable model [ZinnB04; 59-68].

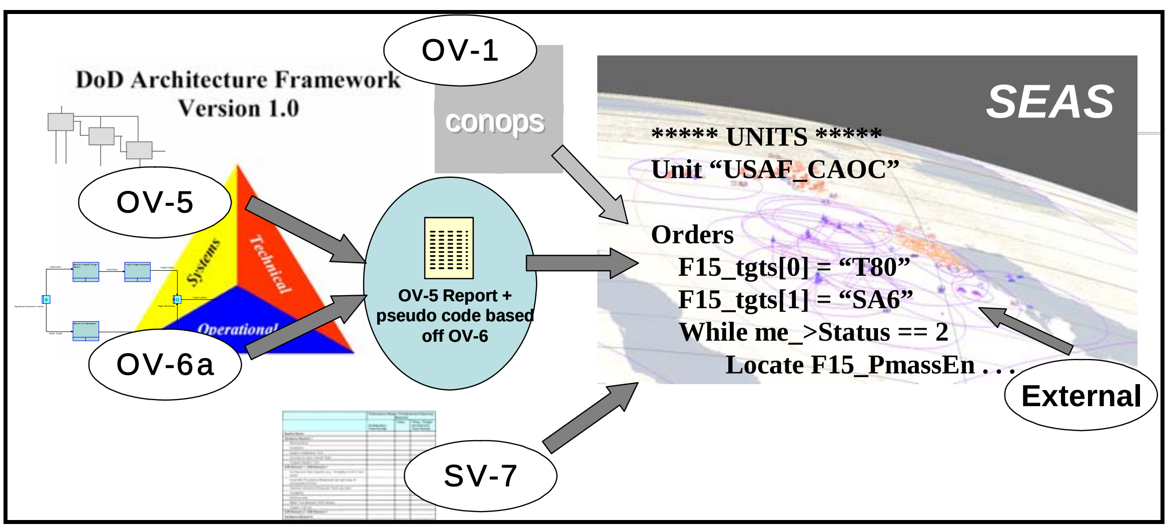

Figure 3: Mapping orders from DoDAF [ZinnB04; 68]

Approach

To develop an executable model from a static architecture description, one needs to review thoroughly the product's consistency and completeness. After identifying any inconsistencies or errors, the architecture will be converted from Popkin System Architect to the CORE™ software environment. The first step of the migration is the creation of the Integration Definition for Function Modeling (IDEF0) diagrams in CORE™. The IDEF0, along with the rules model and the information exchange matrix, are the key to identifying the threads necessary to introduce the WBBDA capability to the architecture. Once the threads are identified, the products are then modified to reflect the addition of WBBDA. Both the baseline and the WBBDA enabled architectures will be converted to executable models. To initiate the development of the executable model, the information obtained from the IDEF0 diagram and rules model will be used to help generate the enhance FFBD. The enhanced FFBD is an executable graphic based model which combines the activity/data flow diagram from IDEF0 with the control and logic structure represented in the FFBD. Once the models are created a set of parameters are introduced to the different models. The executable model will be exercised for varying values of the lethality parameters. The data will be reduced and analyzed to compare the two architectures. Care must be taken to develop a set of metrics that fairly compares the architectures. A simple metric was selected for the comparison of the two architectures; number of sorties to an effective kill. This metric is a top level assessment of the architectures that doesn't depend on architecture specifics.

Document Outline

The second chapter of this thesis discusses the current use and importance of architectures in the Department of Defense. This is followed by the description and current status of how Time Critical Targets are handled, both with and without the proposed Battle Damage Assessment capability in chapter three. Chapter four provides details of the process used to perform the architectural analysis. The results obtained and analysis performed is included in chapter five. Finally conclusions and recommendations are covered in chapter six.

II. Use of Architectures in DoD

Department of Defense Architecture Framework (DoDAF)

The DoD Architecture Framework (DoDAF) provides a common description for architecture development, presentation, and integration [DoDAFI03; es-1]. The new approach needs to be able to break communication barriers throughout the acquisition community. The desired end result is a common language that can communicate and relate architectures and systems across the many DoD and industrial organizations. DoDAF strives for a transparent exchange of information by developing three concurrent views; Operational View (OV), System View (SV), and Technical View (TV) as shown in Figure 4, illustrates the relationship between the three views.

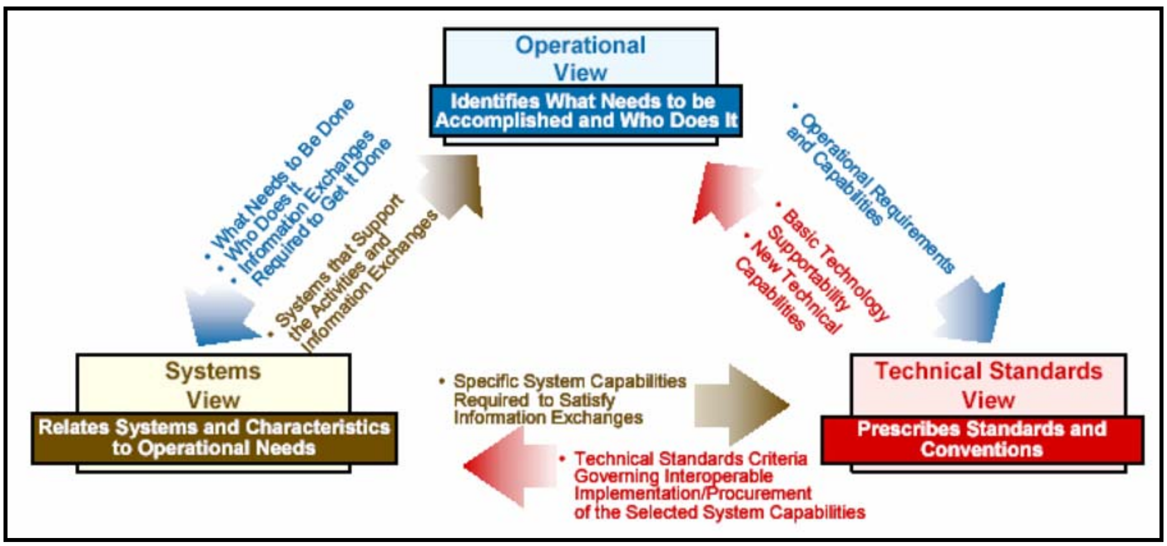

Figure 4: DoDAF architectural views interrelations [DoDAFI03; es-1]

The views can be thought as of as photographs of the same system that are taken from different angles. The operational view identifies “What needs to be accomplished?” and “Who does it?” The System View relates systems and characteristics to operational needs. The Technical View (technical standards view) prescribes standards and conventions. The framework is partitioned into two volumes and a deskbook. The resulting 26 individual views (see Table 3) each have a purpose, and the combinations of the views yield a complete and accurate representation of the system.

The DoDAF defines an integrated architecture as an architecture description that has integrated Operational, Systems, and Technical Standards Views. An architecture description is defined to be an integrated architecture when products and their constituent architecture data elements are developed such that architecture data elements defined in one view are the same as architecture data elements referenced in another view [DoDAFI03; ES-1]. A new product proposed by Dickerson and Soules, an executable model, can be used to help in the validation of the architectural design of a system [D&S04; 15]. Unfortunately the process and benefits of a direct migration from architectural products to an executable architecture are not well defined or understood.

Table 3: Essential and Supporting Framework Products [DoDAFI03; 1-4]

Applicable View	Framework Product	Framework Product Name	General Description
All Views	AV-1	Overview and Summary Information	Scope, purpose, intended users, environment depicted, analytical findings
All Views	AV-2	Integrated Dictionary	Architecture data repository with definitions of all terms used in all products
Operational	OV-1	High-Level Operational Concept Graphic	High-level graphical/textual description of operational concept
Operational	OV-2	Operational Node Connectivity Description	Operational nodes, connectivity, and information exchange needlines between nodes
Operational	OV-3	Operational Information Exchange Matrix	Information exchanged between nodes and the relevant attributes of that exchange
Operational	OV-4	Organizational Relationships Chart	Organizational, role, or other relationships among organizations
Operational	OV-5	Operational Activity Model	Capabilities, operational activities, relationships among activities, inputs, and outputs; overlays can show cost, performing nodes, or other pertinent information
Operational	OV-6a	Operational Rules Model	One of three products used to describe operational activity—identifies business rules that constrain operation
Operational	OV-6b	Operational State Transition Description	One of three products used to describe operational activity—identifies business process responses to events
Operational	OV-6c	Operational Event-Trace Description	One of three products used to describe operational activity—traces actions in a scenario or sequence of events
Operational	OV-7	Logical Data Model	Documentation of the system data requirements and structural business process rules of the Operational View
Systems	SV-1	Systems Interface Description	Identification of systems nodes, systems, and system items and their interconnections, within and between nodes
Systems	SV-2	Systems Communications Description	Systems nodes, systems, and system items, and their related communications lay-downs
Systems	SV-3	Systems-Systems Matrix	Relationships among systems in a given architecture; can be designed to show relationships of interest, e.g., system-type interfaces, planned vs. existing interfaces, etc.
Systems	SV-4	Systems Functionality Description	Functions performed by systems and the system data flows among system functions
Systems	SV-5	Operational Activity to Systems Function Traceability Matrix	Mapping of systems back to capabilities or of system functions back to operational activities
Systems	SV-6	Systems Data Exchange Matrix	Provides details of system data elements being exchanged between systems and the attributes of that exchange
Systems	SV-7	Systems Performance Parameters Matrix	Performance characteristics of Systems View elements for the appropriate time frame(s)
Systems	SV-8	Systems Evolution Description	Planned incremental steps toward migrating a suite of systems to a more efficient suite, or toward evolving a current system to a future implementation
Systems	SV-9	Systems Technology Forecast	Emerging technologies and software/hardware products that are expected to be available in a given set of time frames and that will affect future development of the architecture
Systems	SV-10a	Systems Rules Model	One of three products used to describe system functionality—identifies constraints that are imposed on systems functionality due to some aspect of systems design or implementation
Systems	SV-10b	Systems State Transition Description	One of three products used to describe system functionality—identifies responses of a system to events
Systems	SV-10c	Systems Event-Trace Description	One of three products used to describe system functionality—identifies system-specific refinements of critical sequences of events described in the Operational View
Systems	SV-11	Physical Schema	Physical implementation of the Logical Data Model entities, e.g., message formats, file structures, physical schema
Technical	TV-1	Technical Standards Profile	Listing of standards that apply to Systems View elements in a given architecture
Technical	TV-2	Technical Standards Forecast	Description of emerging standards and potential impact on current Systems View elements, within a set of time frames

Representative use of DoDAF architectural Views

The Joint Capability Integration and Development System (JCIDS), is established to satisfy the need for a joint concepts-centric capabilities identification process, see Table 4. It is especially relevant as the TSEO2012 is a capability driven net-centric operation.

Table 4: Principle JCIDS Analyses [DoDAFI03; 3-16]

Functional Area Analysis (FAA)	identify the tasks to be reviewed
Functional Needs Analysis (FNA)	based on the tasks identified in the FNA, identify capability gaps or redundancies
Functional Solution Analysis (FSA)	for the capability gaps or redundancies identified in the FSA, assess the potential DOTMLPF approaches

The Functional Area Analysis (FAA) is based on cross-capability analysis and identifies the tasks to be reviewed in the Functional Needs Analysis (FNA). The Operational Activity Model (OV-5) used in association with the Universal Joint Task List (UJTL) can provide insight into the tasks to be accomplished, the relationships and information flows between those tasks, and the system functions from Systems Interface Description (SV-1) supporting the tasks. Operational Rules Model, State Transition Description, and Event-Trace Description (OV-6) provide critical timing and sequence attributes, and documents the operational threads. Operational Activity to Systems Function Traceability Matrix (SV-5) provides a basis for identifying activities not supported by existing materiel solutions [DoDAFv1VolI03; 3-16, 3-17].

FNA is performed for the tasks identified in the FAA step. Key players and the operational information exchange requirements for tasks/activities of interest are identified in the Operational Node Connectivity Description (OV-2). Systems

Communication Description (SV-2) provides the basis for identifying existing connectivity. SV-5 in conjunction with the system functions to systems mapping described in SV-1, contributes toward identifying capability gaps and redundancies [DoDAFv1VolI03; 3-17].

The first step for a Functional Solution Analysis (FSA) is to determine if the integrated Doctrine, Organization, Training, Leadership & Education, Personnel, and Facilities (DOTLPF) approach can fill the capability needs identified in the FNA. The DOTLPF attributes are:

- Doctrine influencing the activities (controls from OV-5)
- Organizations responsible for activities (OV-2, operational nodes)
- Training or skill set needed to conduct the activities (human roles represented by operational nodes in OV-2)
- Leadership and education (through OV-2 nodes and their association with the organizational hierarchy of OV-4)
- Personnel conducting operations
- Facilities specified as systems nodes in SV-1, as well as operational threads (OV-6c) that describe capabilities

The Functional Solution Analysis (FSA) identifies the most promising approach to providing the capability, but should not define a specific system solution. The FSA sets the boundary conditions within which the Analysis of Alternative (AoA) should be performed [DoDAFv1VolI03; 3-17]. The SV-5 can be used with SV-1 and/or SV-2 and possibly SV-4 to provide a basis for assessing various approaches for achieving a capability via materiel approach. OV-3 may be used to describe information exchange requirements. Technical standards (TV-1) may be applicable to factor technical constraints to the JCIDS analysis process [DoDAFv1VolII03; 3-17].

Department of Defense Architecture Framework (DoDAF) Extensions

As previously discussed Dickerson and Seouls have divided the products into five groups to support Architectural Analysis. Grouping these products allows the system engineer to go directly to the products necessary to perform a specific analysis. This saves the system engineer from the task of reviewing all the products to obtain the relevant information concerning his analysis. Figure 5, represents how products support architectural analysis.

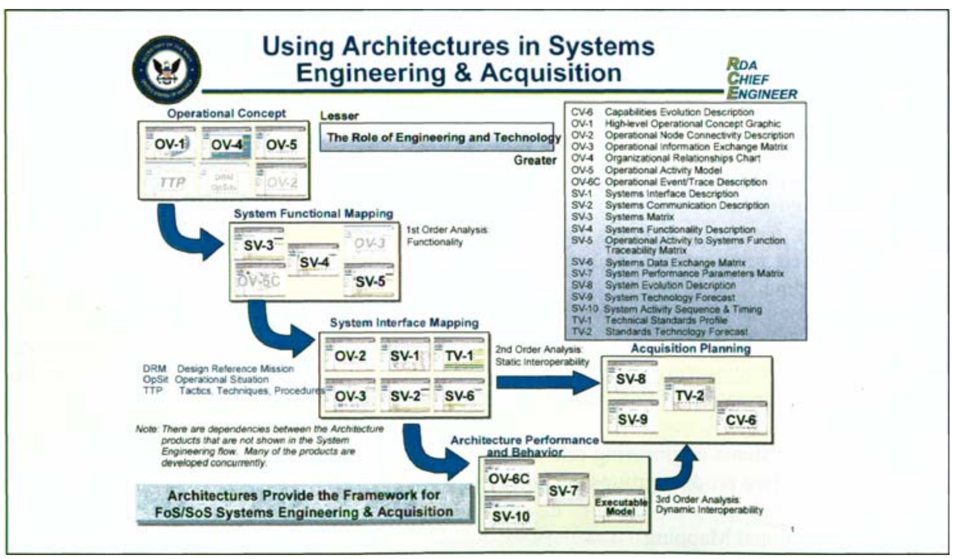

Figure 5: Architectural product grouping [D&S04; 12]

From the figure we can see a new architectural product, the executable model. The executable model was discovered to be essential in the Dynamic Interoperability analysis. How this is the real enhancement, as it transform the diacritic products into dynamic analytical resources. The products become dynamic, because as the model is executed the products become the parameters that are perturbed to obtain the desired performance and behavior (trade studies). If the executable model is developed from the OV-1, OV-5 and OV-6a, as is performed later on the study, we can study the behavior of the system. For a Dynamic assessment, Dickerson and Seouls require the following products, OV-1, OV-5, OV-6a, OV-6c, SV-7, and SV-10. Comparing these products to those Zinn regarded as essential to populate an Agent based analysis we see an overlap of four products OV-1, OV-5, OV-6a, and SV-7. We are narrowing down to the essential products needed to develop an executable model.

Levis enforces the necessity of executable models as an important part of the system architecture validation process [Levis04] and favors CPN for the development of the executable model. CPN have the ability to handle the high demand for mathematical rigor but fall short when modeling a variable (dynamic) environment.

Levis' take on Architecture validation takes the form of a layered process, with each increasing step requiring more effort and information. The first layer tackles the logical correctness of the architecture. The second layer verifies the architecture exhibits the desired behavior. The third level evaluates the performance characteristics of the architecture. The fourth level explores the feasibility that the architecture provides the desired capability, and the last (fifth) layer compares/analyzes different alternatives. Figure 7 is a representation of how the Levis' layered process fits our present problem.

Table 5: Layered analysis process

Layer descriptions pertaining to the investigation (TSEO2012):	
First layer (logic)	Detect - Find disturbances Elect - Rationalize the effect Select - Build course of action Affect - Shape the effect
Second layer (behavior)	Engage target Re-engage target in a non-kill scenario Avoid infinite loops
Third layer (performance)	Number of sorties to destroy target
Four layer (capability)	Capability to terminate a high priority targets
Fifth layer (alternative analysis)	Provability of system to effectivaly engage and terminate a high priority target given maximum number of sorties per mission.

An executable model is an essential part of all the layered levels. For Levis an executable model needs to have the following characteristics:

1. It is derived from the architecture design in a traceable way
2. It has an underlying mathematical model that enables the application of analytical tools
3. It enables simulation

Modeling and Simulation (M&S)

The executable architecture can be a good starting point for any modeling and simulation effort. The combination of information contained in the products and the behavior obtained from the executable model, give the M&S developer good insight into the intent of the architecture designer. A good number of modeling errors arise from the miscommunication of the System Engineer (SE) and M&S developer. An executable model can serve as a communication tool/enhancer between the SE and M&S developer.

The degree of detail of the executable model is left to the systems engineer. The right amount of detail is essential to determine the validity of the system, but too much detail can be a hindrance and/or unnecessarily drive up analysis costs. The amount of detail in an executable model is not a constant that can be set as a rule for all executable architectures.

Products provided by DoDAF should provide most of the information necessary to model a weapon system in an agent based simulation [ZinnB04; 91]. Eight important products for weapon system description for agent based modeling were identified (SV-7, OV-4, SV-1, SV-2, SV-6 or OV-3, OV-1, OV-5, and OV-6a). We could use these products as a starting point for the development of the executable model. The transition from an architecture to a M&S is similar to the migration to an executable model. The differences lie in the level of detail that is wanted/needed for a good M&S model. The M&S model tends to be more detailed than an executable model, requiring more extensive and precise supporting information. Analysis using modeling and simulation is important at all stages of system development.

Summary

The purpose of the executable model is to support architecture-based analysis. There are many perspectives regarding the use of architectural products to help in the development of an executable model. Even as there is some debate as to which products are essential in the development of the executable model, there is consensus in that the products should be as complete as possible to provide for a smooth development and ease of traceability. The architectural baseline used in this investigation had a limited number of architecture products available. The three major contributors to the executable model where the OV-1, OV-5, and OV-6a. Choice of architecture products is limited to the relevant available products.

III. Operational Concepts

Baseline Architecture Review

Before establishing the architectural base line, the evolution of the Time Sensitive Effect Operation (TSEO2012) architecture will be discussed. The TSEO2012 architecture evolved from the Time Critical Target architecture (TCT2005) that is part of the Command and Control (C2) Constellation. The C2 Constellation needed a near-term (2005) and a midterm (2012) perspective to manage the Time Sensitive Target Problem.

To accommodate the new capability based approach adopted by the DoD, a new architecture is proposed to address the Time Sensitive Target Problem. The new architecture addresses effects management instead of target prosecution. By managing desired effects we can create the desired outcomes in the battlespace, thus controlling the tempo of the conflict. This is a very different focus than the one which dominated during the Vietnam era mentality, which focused on quantity of kills rather than imposing our will on the enemy.

The proposed TSEO2012 architecture is a planned evolution of the current TCT2005 architecture. The targets' hierarchical importance is determined by the effect obtained by removing it from the battlespace. In contrast, the TCT2005 architecture gives hierarchal importance depending on the type of target, not taking into account its relevance in the battlespace or any effect obtained by removing the target from the battlespace.

Table 6: Available Framework Products for C2, TCT2005 and TSEO2012 architectures

Applicable Architecture View	Product Reference	Architecture Product	C2	TCT2005	TSEO2012
All Views	AV-1	Overview and Summary Information	Available	N/A	N/A
All Views	AV-2	Integrated Dictionary	Available	N/A	N/A
Operational	OV-1	High-level Operational Concept Graphic	Available	Available	Available
Operational	OV-2	Operational Node Connectivity Description	N/A	N/A	Available
Operational	OV-3	Operational Information Exchange Matrix	Available	N/A	Available
Operational	OV-4	Command Relationships Chart	N/A	N/A	N/A
Operational	OV-5	Activity Model	N/A	Available	Available
Operational	OV-6a	Operational Rules Model	N/A	Available	Available
Systems	SV-2	Systems Communications Description	N/A	N/A	N/A
Systems	SV-4	Systems Functionality Description	Available	N/A	N/A
Systems	SV-5	Operational Activity to System Function Traceability Matrix	Available	N/A	N/A
Systems	SV-6	System Information Exchange Matrix	N/A	N/A	N/A
Systems	SV-7	System Performance Parameters Matrix	N/A	N/A	N/A
Systems	SV-9	System Technology Forecast	Available	N/A	N/A
Technical	TV-1	Technical Architecture Profile	Available	N/A	N/A
Technical	TV-2	Standards Technology Forecast	Available	N/A	N/A

Table 6 lists the architecture products available for the C2, TCT2005, and TSEO2012 architectures. As shown, a limited number of products are available for both the TCT2005 and TSEO2012. Fortunately the products available are important to develop the executable model. We don’t have a specific description for the Overview and Summary (AV-1) for the TCT2005 and TSEO2012 architectures, as they are threads within the C2 Constellation, they share the AV-1 for that architecture. The AV-1 for the C2 Constellation is shown in Table 7.

Table 7: AV-1 pertaining to TCT2005 and TSEO2012 architecture

Purpose	Define the highest level aspects of the C2 Constellation but does not include all of the functionalities and associated systems.
Scope	Represent a baseline and near-term view (2005) of the C2 Constellation, along with a midterm (2012) perspective. By implementing a Monitor, Plan, Execute and Assess Framework.
Intended Users	Joint Force Air Component Commander (JFACC).
Environment	Theater wide, time sensitive operations to include Combat Search and Rescue (CSAR).

The TSEO2012 architecture is an evolution of the TCT2005 concept, this evolution will now be discussed. As shown in Figure 6 the Operational Concept Graphic

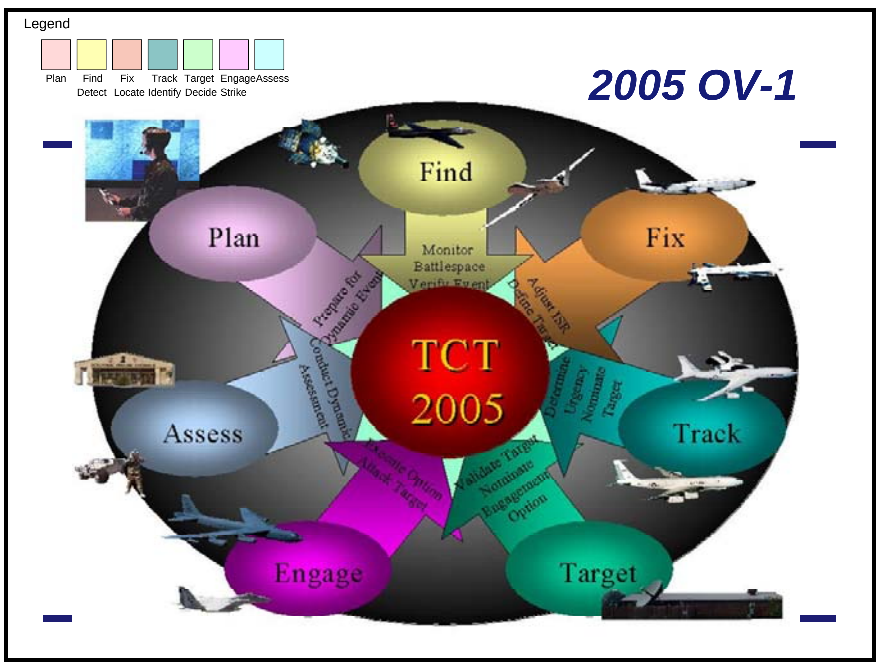

Figure 6: TCT2005 Operator perspective. Focuses on immediate unplanned/unanticipated targets (TST)

(OV-1), the TCT2005 focus is on immediate unplanned/unanticipated targets (TST's). The Operational Concept follows the Air Force paradigm of P-F2T2EA, which stands for Plan, Find, Fix, Track, Target, Engage, and Assess. Unfortunately, the OV-1 graphic comes with no formal description other than the TCT framework of P-F2T2EA used for its creation. An OV-1 without a textual description is an incomplete product. Graphics alone are not sufficient for capturing the necessary architecture data [DoDAFII03; 4-1]. It was necessary to review other architecture products in order to discern the complete concept of operations. Table 8 provides some of the information that could be included with the OV-1 product.

Table 8: TCT2005 activity breakdown [Vittori03]

Phase	Related Activity
Plan	TCT-Analyze ATO period for dynamic targeting opportunities
Find	TCT-Monitor battlespace for dynamic events TCT-Verify event/indication is of interest
Fix	TCT-Adjust Theater ISR to support dynamic air operations TCT-Define target/target set
Track	TCT-Determine target significance/urgency
Target	TCT-Validate target/target set TCT-Nominate engagement option
Engage	TCT-Execute engagement option TCT-Attack target
Assess	TCT-Conduct dynamic assessment of target

Air Operations Center (AOC) staff use available Intelligence, Surveillance and Reconnaissance (ISR) resources to update changes to the enemy status and build a list of potential targets that may disrupt the upcoming Air Tasking Order (ATO) flow (Plan). Throughout the ATO period, the AOC uses current information sources to discern anomalies or questionable occurrences. A triggering event (e.g., a target appearance) must be verified, and a determination must be made as to whether or not it is a previously identified dynamic target or a new one (Find). ISR resources may need to be adjusted to focus on the desired target or area. The AOC uses target information to verify the location of, or fix the target/target set (Fix). Utilizing track data and target information, the target/target set disposition and availability is determined (Track). The target/target set is examined to see if it fits Joint Force Commander (JFC)/Joint Force Air Component Commander (JFACC) guidance, its impact on planned operations, and other imposed restrictions. An engagement option is nominated through a weighted comparison (Target). Execution orders are created and assets are instructed to engage the target (Engage). Attacking assets and focused ISR provide damage assessment. Target/target set status is determined and, if necessary, a decision to re-attack made (Assess). With the previous modification, the OV-1 for TCT2005 is complete.

The TSEO2012 architecture will now be discussed. Starting with the Operational Concept Graphic (OV-1), Figure 7, the crucial textual description is missing/ not available again. Enough information is scattered among the available products and supporting documents to determine that the TSEO2012 is focused on managing effects to influence the battlespace.

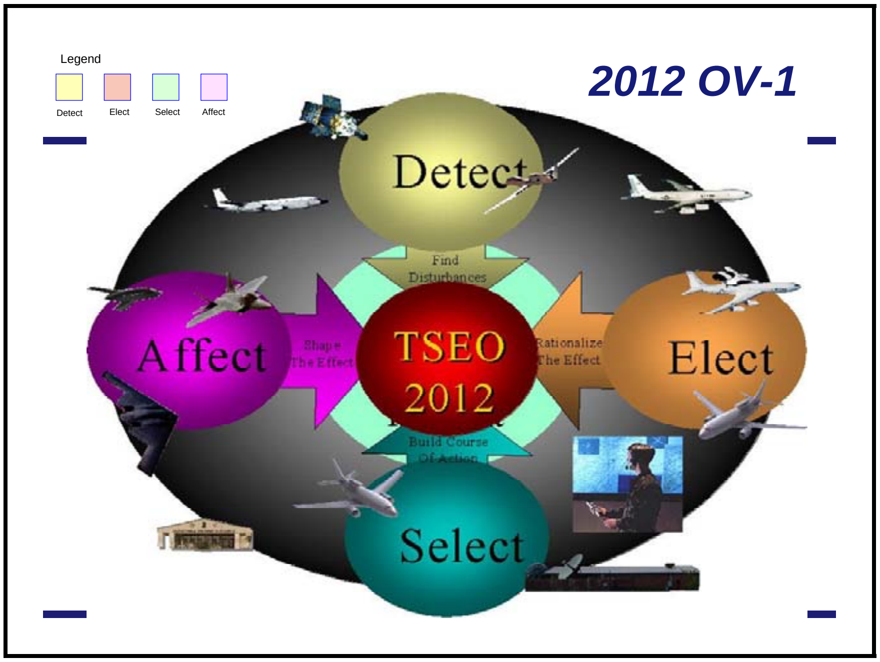

Figure 7: TSEO2012 Simple construct that focuses on effect management, not merely target prosecution.

The depicted construct looks simple compared to the TCT2005 construct, but both have the same number of underling activities. These activities are modified from the previous architecture to better represent the effect driven focus of TSEO2012. Table 9 relates the activities to the OV-1. Shown, the activities are divided into four phases, with the "Detect" phase carrying the bulk of the activities.

Table 9: TSEO2012 activity breakdown [Vittori03]

Phase	Related Activity
Detect	TSEO-Analyze Unplanned Immediate Targeting Opportunities TSEO-Monitor Battlespace for Dynamic Events TSEO-Verify Event/Indication is of Interest TSEO-Adjust ISR to Support Dynamic/Time Sensitive Effects Operations TSEO-Conduct Combat Identification (CID)
Detect/Elect	TSEO-Predict Effect/Urgency
Elect	TSEO-Validate Desired Effect TSEO-Nominate Engagement Option
Elect/Affect	TSEO-Manage Engagement
Affect	TSEO-Produce Effect
Detect	TSEO-Conduct Dynamic Assessment of Effect

A description of the TSEO cycle is shown in Figure 8, and this will serve as a textual description for the TSEO OV-1. The TSEO cycle starts with an Unplanned Immediate Target Watch List (UITWL). This is monitored for Unplanned Immediate

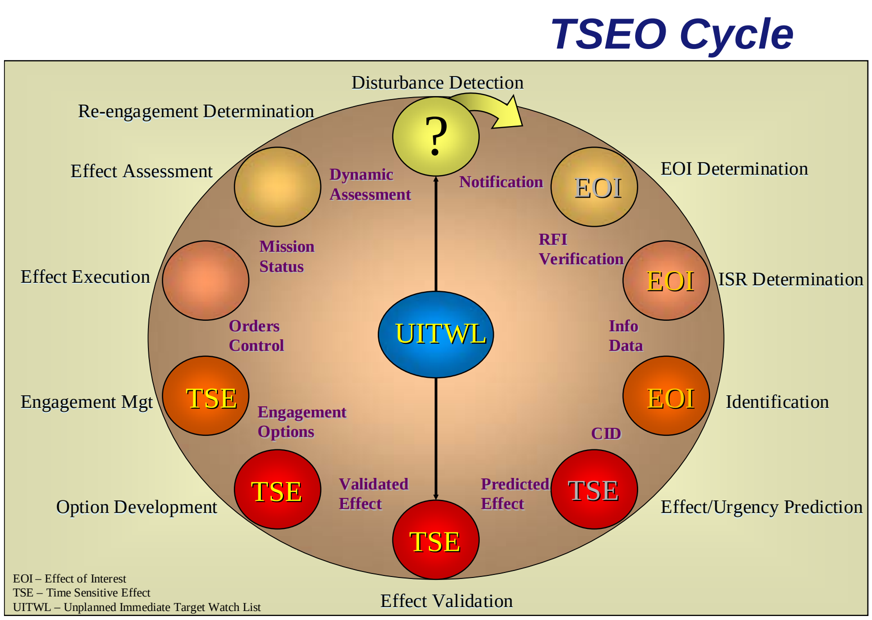

Figure 8: TSEO Cycle [Vitori03]

Targets (UIT). Upon detection of an anomaly or disturbance, concerned parties are notified and they begin to assess if it is an Effect of Interest (EOI). The EOI is something that may impact ongoing or future operations. It could be a threat or an exploitable opportunity. Regardless, we need to identify the root causes of the effect. Next, we analyze the EOI's movement, predict its significance, and determine its urgency. The EOI is designated as a Time Sensitive Effect (TSE); the desired outcome or effect is determined. The TSE is then validated, to determine if it is desirable and worthwhile? If so, engagement options are established and one is selected for execution. AOC manages the engagement while assets carry out operations to create the desired effect. ISR and engaging assets provide feedback; engagements are assessed rapidly. This may result in a decision to re-engage [Vittori04]. Note that the architecture considers the engaging assets as part of the feedback loop. An unfortunate side effect of autonomous, precision guided munitions with significant stand-off range is that the traditional "eyes on target" assessment information has become difficult to obtain.

The noted difference from the previous architecture is the fact that the TSEO2012 scrutinizes a list of available effects, selects the desired effect/outcome, and generates an engagement option. In contrast the TCT2005 focuses on the available engagement options, reviews their effects and selects the engagement through a weighted comparison. The key words between the two are "engagement options and effects". TCT2005 has a set of engagements available to it when a target is identified the effects of the preset options are compared and the most advantageous is selected. In contrast the TSEO2012 look at the unpredicted events and decides what outcome is more desirable. By selecting an outcome and pairing it with an engagement option, the architecture is not restricted by a fixed set of engagement options. This new approach makes available a wider set of possibilities that were not available before.

Because the TSEO2012 will serve as the basis for our analysis, important nodes and interactions will be identified by reviewing the Operational Activity Model (OV-5) and the Operational Activity Sequence (OV-6a). The OV-5 is in the form of an IDEF0 and supporting activity description document. The graphical representation is mostly complete and legible, but sometime it tends not to follow strict IDEF0 graphical structures as shown in Figure 9 and Figure 10. The IDEF0 was reconstructed in CORE™ to facilitate its review. The activity description document supporting the IDEF0 was

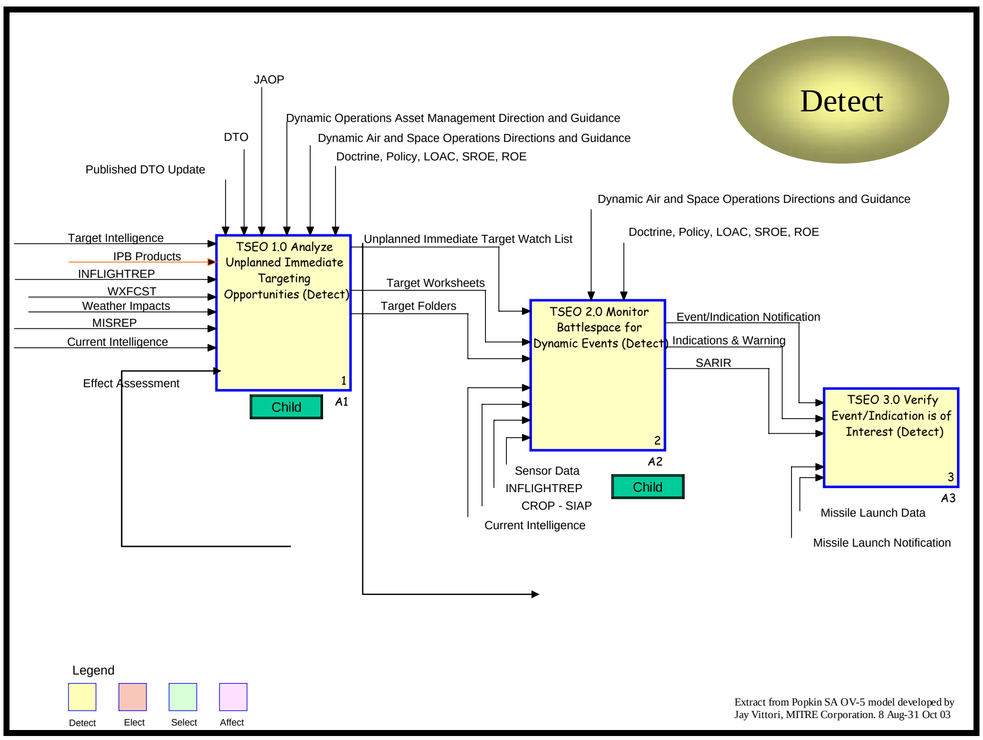

Figure 9: IDEF0 is segregated into blocks of three functions each. [Vittori03]

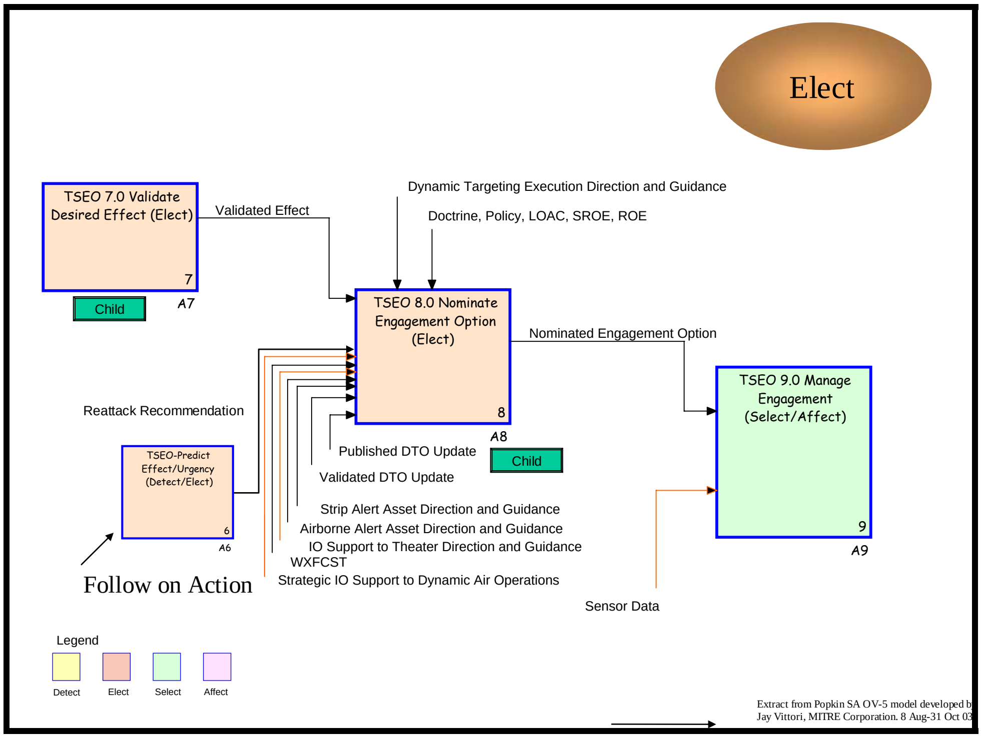

Figure 10: Functions on the corners are missing some inputs and outputs. [Vittori03]

adequate for this top level investigation, but it will need some expansion for more in depth investigation.

The OV-6a is in the form of a Function Flow Block Diagram (FFBD) having a model description and unit behavior description supporting documents. The OV-6a was lacking a top level system function FFBD diagram, an important piece of information for performing a top level analysis. As observed in Figure 11, subfunctions FFBD diagrams are lacking important interface reference blocks.

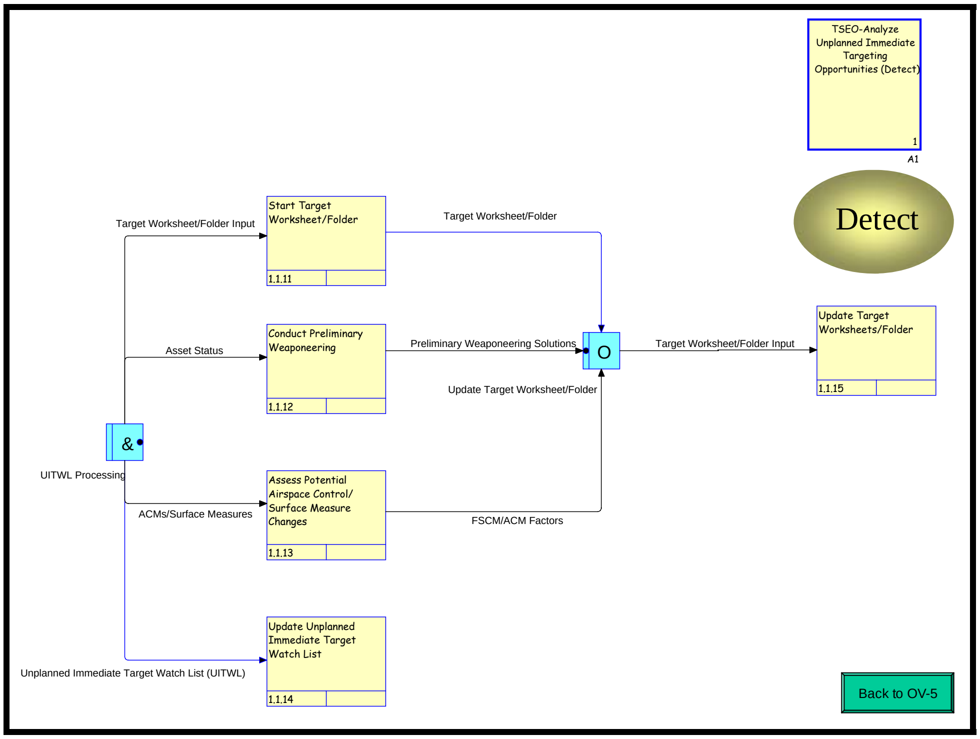

Figure 11: Subfunction FFBDs have no supporting interface reference blocks. [Vittori03]

OV-6a documents describe elements of the diagrams, but does not provide any of the underlying logic, see Table 10 and Table 11. This made deciphering the FFBDs more difficult. The lack of precise structured rules makes difficult the study of lower level components, limiting the study to a top level interactions. As the Top level FFBD is missing, a new one needs to be generated to support the development of an executable model.

Table 10: Exert from OV-6a model description, provides no logic insight. [Vittori03]

Name	2012 OV-6a Model Description	Need	Purpose	Scope
Manage Unplanned Immediate Target Watch List	This model describes the rules, processes and units of behavior associated with building and managing the Unplanned Immediate Target Watch List. This model is based on a Major Theater War/Decisively Defeat scenario. This model focuses on ops requirements and considerations associated with heavy planned tasking (2000+) missions within a 24 hour execution period; it could also apply to small scale tasking (less than 500 missions).	Successful Time Sensitive Effects Operations begin with a clear understanding of those time-sensitive targets that may affect an upcoming execution period, but were not included in the planning process or have not been weaponeered. The UITWL provides the basis for that understanding.	This model establishes the baseline procedures to build and maintain the Unplanned Immediate Target Watch List (UITWL)	The model begins with a review of the current Dynamic Tasking Order and continues through the development and update of the UITWL. It includes prioritization and preliminary weaponeering of Unplanned Immediate Targets.
Monitor Battlespace for Dynamic Events	This model describes the rules, processes and units of behavior associated battlespace monitoring during the DTO execution period. This model is based on a Major Theater War/Decisively Defeat scenario. This model focuses on ops requirements and considerations associated with heavy planned tasking (2000+) missions within a 24 hour execution period; it could also apply to small scale tasking (less than 500 missions).	In 2012, C2 Warriors will face hi-tech, effective challenges to ongoing operations and processes. To face these threats, our Warriors need to cull the required data and information and quickly translate disturbances to provide warnings or notifications.	This model establishes the baseline procedures to monitor the battlespace and provide feedback through warnings or notifications.	The model begins with C2 Warriors looking out into the battlespace for anomalies. Utilizing culled data and information inputs (to include inflight reports), C2 Warriors prepare and produce notifications or warnings.

Table 11: Exert from OV-6a Unit of Behavior (UOB) description. [Vittori03]

Name	2012 OV-6 Unit Of Behavior Description
Abort Engagement	Engager is in a position to strike, but can not confirm the target. JP 3-09.3, pg. V-11, para 10
Abort Release	Engager determines target should not be engaged at that time. For example, the TCT (a moving vehicle) has moved within a hospital compound and the engager fears collateral damage. JP 3-09.3, pg. V-11, para 10
Acknowledge Weapon Control Status (WCS)/Changes	Engage and support assets acknowledge WCS and changes for the Time Sensitive Effects Operation. NOTE: Weapons Free/Weapons Tight Control Orders impose a status or condition applicable to weapon systems within a defined volume of airspace. Established US doctrine does not allow for further interpretation of weapons control orders against specific target under any circumstance. Any reception of Weapons Free/Weapons Tight Control Orders against specific targets should be immediately clarified via voice request to higher authority. JP 3-01.3 (FC Draft, Nov 02), pg. IV-4, para 3e.
Analyze AMTI	Utilizing subscribed Air Moving Target Indicator data assets, C2 Warriors analyze tracks for anomalies. Derived from JP 2-01.1, Pg. D-2, para 2a(2)

Battle Damage Assessment & Weapon Born Battle Damage Assessment

Battle Damage Assessment (BDA) is an indispensable part of any force application mission. The side that can correctly assess damage in a relatively short amount of time will have an advantage over his enemy and will likely control the tempo on the battlefield. To obtain an accurate damage assessment we need sufficient and timely data regarding the engagement. This has historically been accomplished by satellites, Intelligence Surveillance and Recognizance (ISR) platforms, aircraft controllers, and pilot feedback. These monitoring capabilities are now being augmented with Uninhabited Air Vehicles (UAVs) and further improvements in surveillance technology. The great amount of information taken from the encounters and sometimes sluggish reaction time of ISR platforms makes for a slow and sometimes back logged process. Even with the sometimes slow pace of BDA, there is the potential for information overload, brought about by the large quantity of missions engaging targets of opportunity (Combat Air Patrol missions). Due to increased standoff distance of modern weapons, real time BDA obtained from pilots and air-controllers observations is being lost at an increasing rate, and the accuracy of these assessments is often questionable at best. Regardless, the observations gathered by these resources were valuable as an initial damage assessment to the target. Proposed concept to address these limitations is that of WBBDA. There are a number of proposed alternatives to implement WBBDA. The basic concept is to obtain and utilize critical information available just before and immediately after a munitions strike. The idea is to capture critical indicators that can be exploited. To capture these critical events we will mount dedicated sensors on the munitions. Placing the sensors which are both inexpensive and expendable, in close proximity to the point of impact, will enable fast and accurate data capture. Figure 12 is one of the many WBBDA concepts currently under development and testing. The information desired by the WBBDA will greatly benefit from the Weapon Data Link (WDL), an enabling concept for WBBDA. The WDL is a technology development and

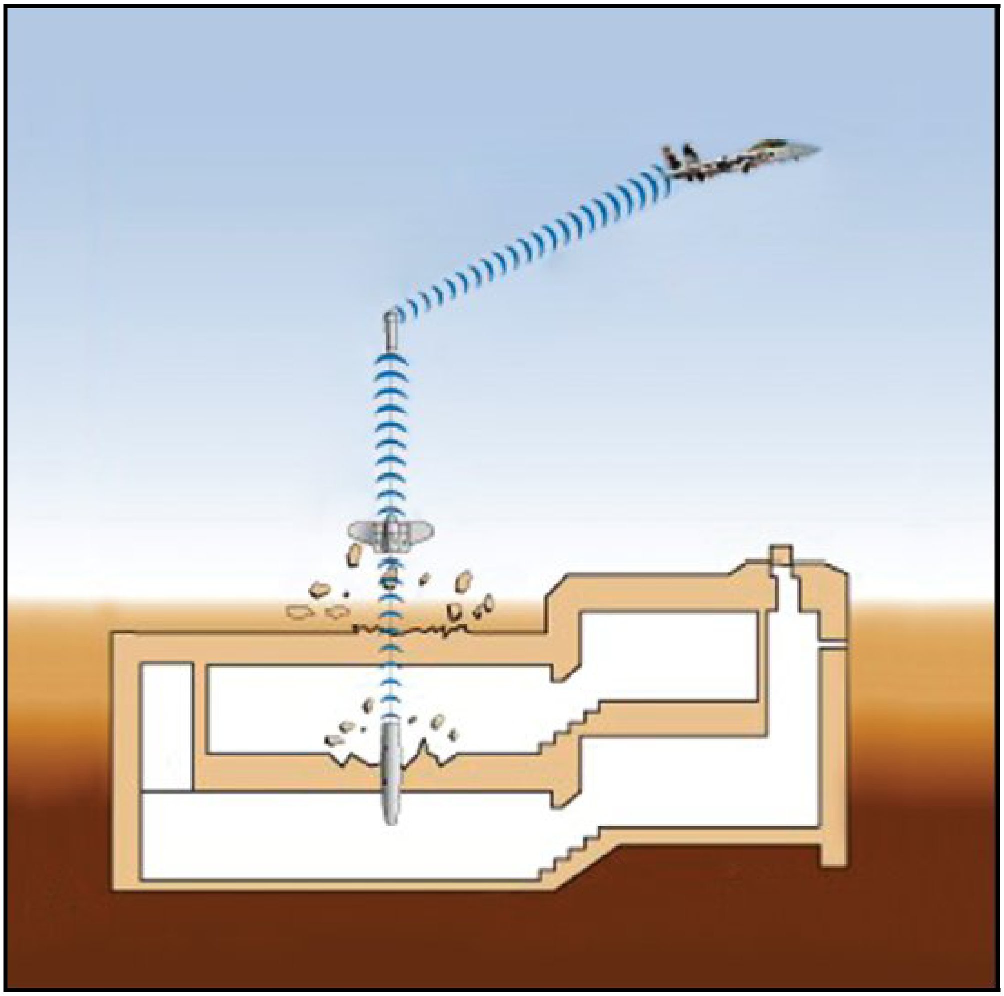

Figure 12: Fuze BDA is one of the many proposed WBBDA

demonstration effort to enhance the functionality of autonomous and semi-autonomous munitions through the use of a common, shared data link among strike and C2 platforms. For the purpose of this investigation we are assuming the presence of an effective WDL. The data obtained from WBBDA can be used to determine the status of the target, near real-time, at the time of engagement. Due to the fast turn around time, re-strike orders can be given to attacking or close by assets. By adding these new threads (information paths) to the TSEO2012, we expect an improvement in the effectiveness of the architecture to engage and destroy time sensitive, high priority targets.

The addition of WBBDA capability to the proposed TSEO2012 architecture provides insight into the type of analysis and studies that can be performed by the executable models. By modifying the architectural baseline and observing the change in behavior of the executable model, we can get a perspective on the usefulness of the executable model to detect/highlight changes in the architecture. The architectural baseline was modified to include the effects of weapon born BDA. Through the executable model, we can observe the effects BDA had on the behavior of the architecture. To obtain a good representation of the behavior of the system two types of warheads will be used, raising the number of effective executable models to four. The four models will provide enough variation to develop a good representation of the systems behavior.

Summary

As depicted in Table 12, only four products match the recommendation given by Zinn for populating an Agent Based simulation. These products are the following; OV-1, OV-3, OV-5, and OV-6a. Out of these four, OV-5 and OV-6a will form the building blocks of our executable model. Even as the TSEO2012 provided a good amount of supportive information in the form of architecture products, it still represents a challenge to understand its structure. Due to the lack of a complete rules model that clearly defines sublevel logic the analysis is limited to top level components. This limitation will not hinder the analysis as we are interested in how executable models support architectural base analysis and how WBBDA affects the overall behavior of the system.

Table 12: Products necessary for simulation

Applicable Architecture View	Product Reference	Architecture Product	Necessary for Agent Based Simulation	TCT2005	TSEO2012
Operational	OV-1	High-level Operational Concept Graphic	Necessary	Available	Available
Operational	OV-3	Operational Information Exchange Matrix	Necessary	N/A	Available
Operational	OV-4	Command Relationships Chart	Necessary	N/A	N/A
Operational	OV-5	Activity Model	Necessary	Available	Available
Operational	OV-6a	Operational Rules Model	Necessary	Available	Available
Systems	SV-2	Systems Communications Description	Necessary	N/A	N/A
Systems	SV-6	System Information Exchange Matrix	Necessary	N/A	N/A
Systems	SV-7	System Performance Parameters Matrix	Necessary	N/A	N/A

IV. Methodology

Architecture Modifications

Before starting the development of the executable model we need to determine how to integrate the WBBDA capability into the existing architecture. The first step is to identify the critical functions that will facilitate a re-strike action. Table 13 depicts the two identified functions that will enable the integration of WBBDA. Function TSEO 9.0, Manage Engagement (Select/Affect), is responsible for assigning the assets to create the desired effect. Function TSEO 11.0, Conduct Dynamic Assessment of Effect (Detect), is responsible for determining the target status. Because of its responsibilities TSEO 9.0 is a good candidate function to receive the WBBDA assessment output from TSEO 11.0. This will bypass TSEO6.0-Predict Effect/Urgency (Detect/Elect), TSEO 7.0-Validate Desired Effect (Elect), and TSEO 8.0-Nominate Engagement Option (Elect) in the re-strike feedback loop, see Figure 13, greatly reducing time to re-engage. It is important to

Table 13: Critical functions for WBBDA implementation [Vittori03]

Name	2012 OV-5 Activity Description
TSEO 9.0 Manage Engagement (Select/Affect)	The selection process resulting in an engagement option and execution orders directing assets to create the desired effect. From: CAOC 4.5.2.9; 2005 C2 Constellation 3.2.5.5; 2005 TCT Activity 9; AOC CONOPS 3.2.5.2.1.5; AFOTTP 2-3.2, pg. 373, para A3.11.1.2.1.4
TSEO 11.0 Conduct Dynamic Assessment of Effect (Detect)	Attacking assets as well as focused ISR provide effect assessment. This information may be used to cycle back through the TSE process to determine effect status and if necessary, urgency of reattack. From: 2005 C2 Constellation 3.2.5.7; 2005 TCT Activity 11; AFOTTP 2-3.2, pg. 368, para A3.10.2.2.5.4.2 and pg. 360, A3.6.2.11; AFPAM 14-210, pg. 71, para 9.4; JP 2-01.1, pg. VI-3, para 2d(3) and pg. D-3, para 2f.

realize that we are assuming a quick response time from the WBBDA; the short response time will not allow any significant status changes/permutations in any of the three functions (TSEO 6,7,8). This assumes that the validated effect and engagement options

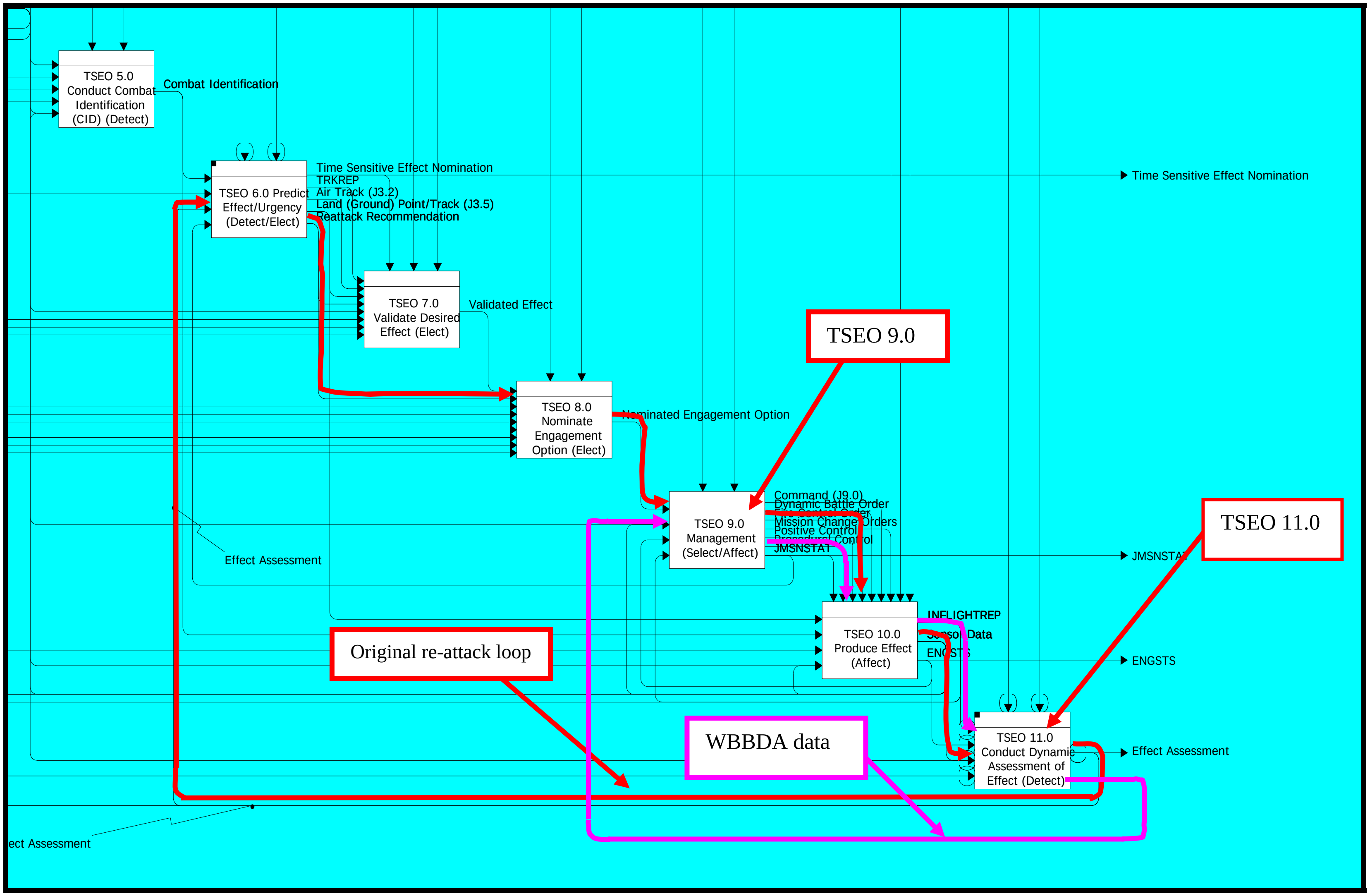

Figure 13: TSEO 9.0 and TSEO 11.0 are identified as important functions. Existing links can accommodate any extra data.

obtained from this chain of functions are still valid and appropriate due to a relative short elapsed time. Taking a top level perspective, we need only to add a new link (WBBDA assessment) from TSEO 11.0 to TSEO 9.0; no other links are needed because we are taking advantage of the existing data links by expanding their domain to accommodate any new information that may be generated. To integrate this new data stream with the WBBDA, an upgrade to the existing overburdened communication hardware must be implemented. The WDL provides the capability of in-flight communications to maintain command authority until detonation; receives in-flight coordinate updates, transmits weapon position and status up to time of impact, transmits damage assessment data, and communicate through direct line-of-sight or SATCOM reachback.

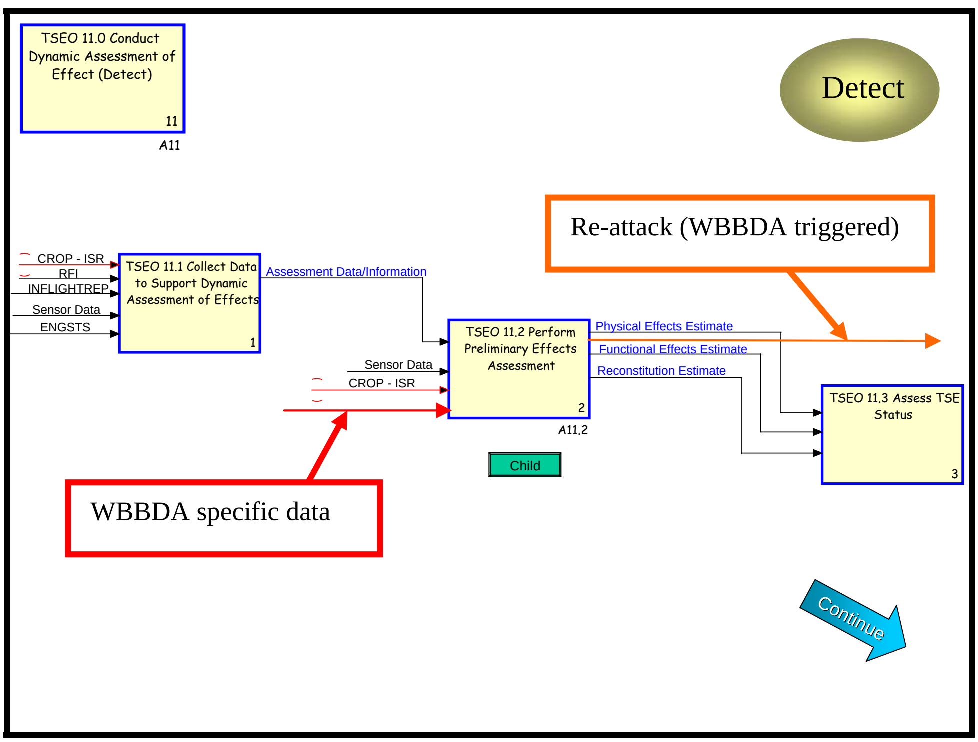

Figure 14: Level 1 of the TSEO 11.0

Below this top level modification, it is necessary to further decompose TSEO 11.0 to can support the addition of the desired WBBDA thread. Figure 14 shows that it can support WBBDA via function TSEO 11.2, Perform Preliminary Effects Assessment. In this function, sensor data can be analyzed to obtain an assessment of the physical condition of the target. The domain of “ Sensor Data” can be increased to accommodate the extra information made available by the WBBDA. This use of the domain set can decrease the complexity of the modified IDEF0.

We add TSEO 11.2 that performs a quick analysis of the functional target status; this will be implemented in a lower level decomposition of TSEO 11.2. As shown in Figure 15 the new WBBDA data is processed and an assessment is obtained.

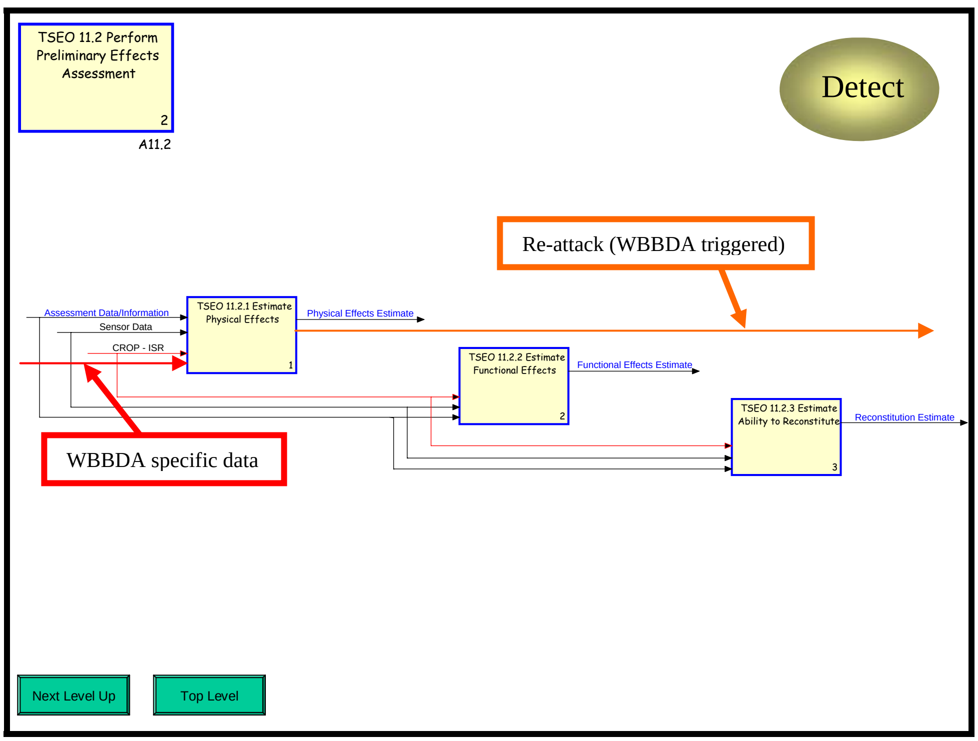

Figure 15: WDBBA data is analyzed with high priority on TSEO 11.2.1.

While there is no real advantage to simulating down to level 2, it is beneficial to design at this level to get a clear understanding of the underlying process.

While the processing of the WBBDA data has been shown, we need to establish where the data is obtained. Returning to level 1, see Figure 14, we see that the “Sensor Data” is an input to the functions TSEO 11.1, Collect Data to Support Dynamic Assessment of Effects, and TSEO 11.2, Perform Preliminary Effects Assessment. Moving one level up we can now see two set of inputs labeled Sensor Data, see Figure 16. We are only interested in the data stream provided by the TSEO 10.0, Produce Effect (Affect), because this function houses the information required to perform WBBDA.

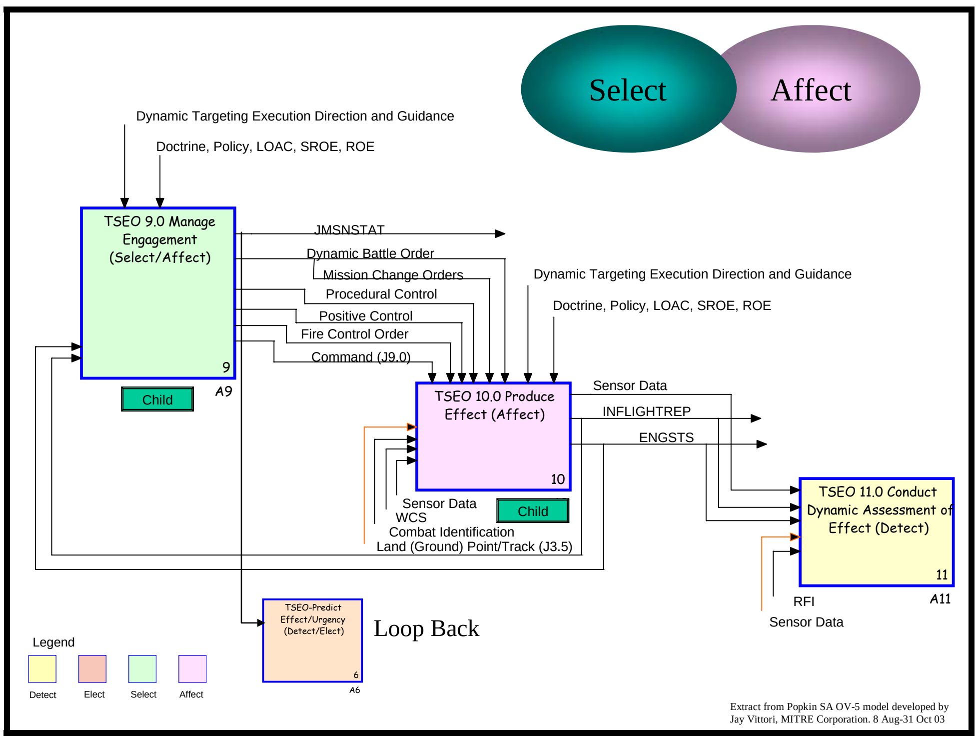

Figure 16: TSEO 11.0 is feed sensor data by two different functions.

Looking at TSEO 10.0, an obvious choice for the correct placement for the WBBDA is in the subfunction TSEO 10.6-Perform Weapon Delivery, see Figure 17.

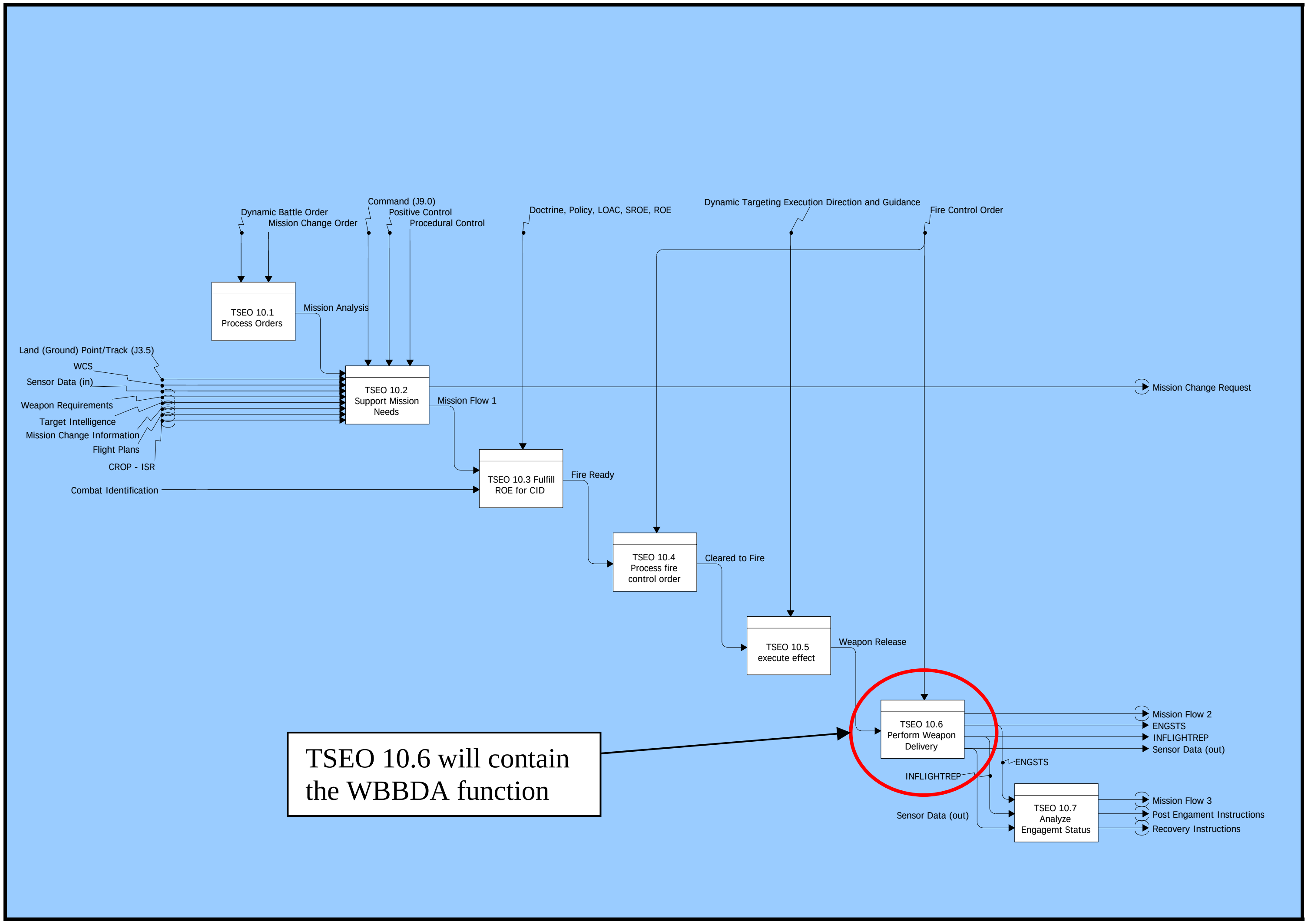

Figure 17: Decomposition of TSEO 10.0

This function represents the actual attack on the target; the information gathered just prior to and post detonation is essential to the BDA. The ability to acquire the necessary data can be obtained by adding the subfunction TSEO 10.6.4-WBBDA, Figure 18. Figure 19 depicts the original decomposition of function TSEO 10.6-Perform Weapon Delivery.

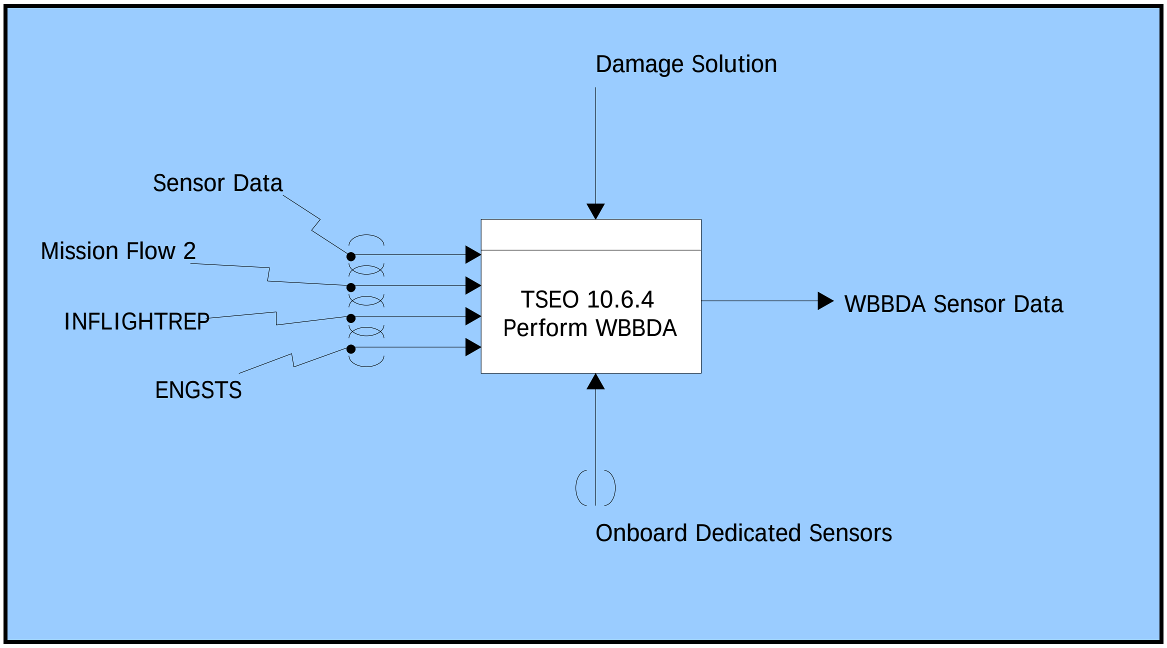

Figure 18: Proposed TSEO10.6.4 WBBDA

We can see a number of essential flows (i.e., Sensor Data, INFLIGHTREP, etc.) are available as defaults in the original decomposition. Taking advantage of this data, and additional information gathered by dedicated sensors to perform a quick BDA. In Figure 20 we see how the proposed function integrates into TSEO 10.6. The TSEO 10.6.4 function assumes the seamless integration of available relevant data sources. In theory this will provide economy of volume for the dedicated WBBDA hardware.

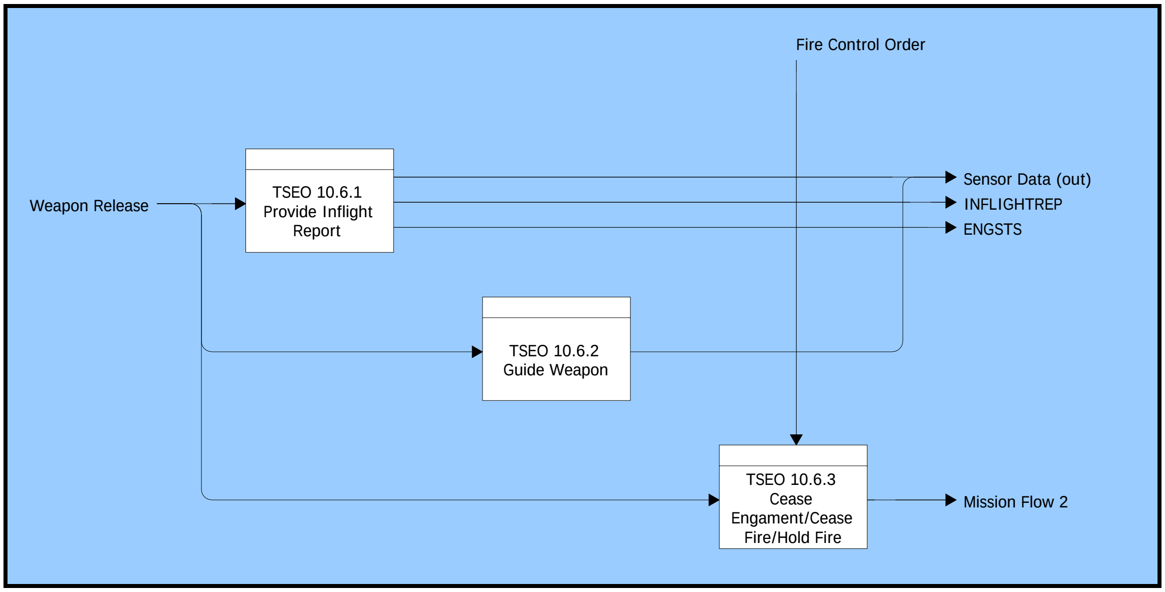

Figure 19: Original TSEO 10.6 Perform Weapon Delivery

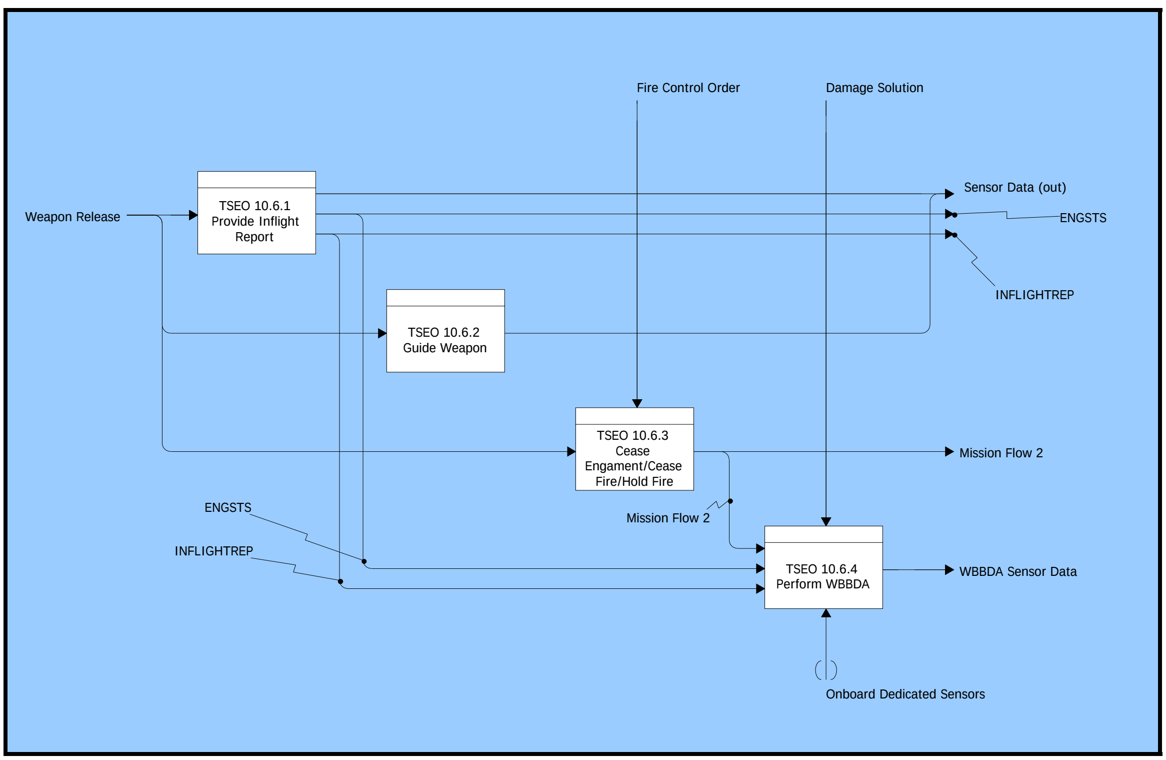

Figure 20: Modified TSEO 10.6 Perform Weapon Delivery

Having identified the necessary additions to the architecture, the rules model can then be modified. As previously mentioned, the top level FFBD was missing, thus it needed to be recreated. This top level FFBD is especially important for an executable model to perform architectural analysis. The lack of reference blocks, as discussed in the previous chapter, can be a major road block as transitions from one FFBD to another can be confusing. Figure 21 represents an example of this type of road block. The exit state of function TSEO 1.0 is not clearly defined; the reference blocks are missing along with exiting thread information. This leaves ambiguity as to how subfunctions TSEO 1.1.18 connects to subfunctions TSEO 2.1.1 and 2.1.8, if they connect at all. To make maters worse OV-6a goes directly to the lowest available level of detail with a different numbering system as OV-5, making traceability to the top levels harder.

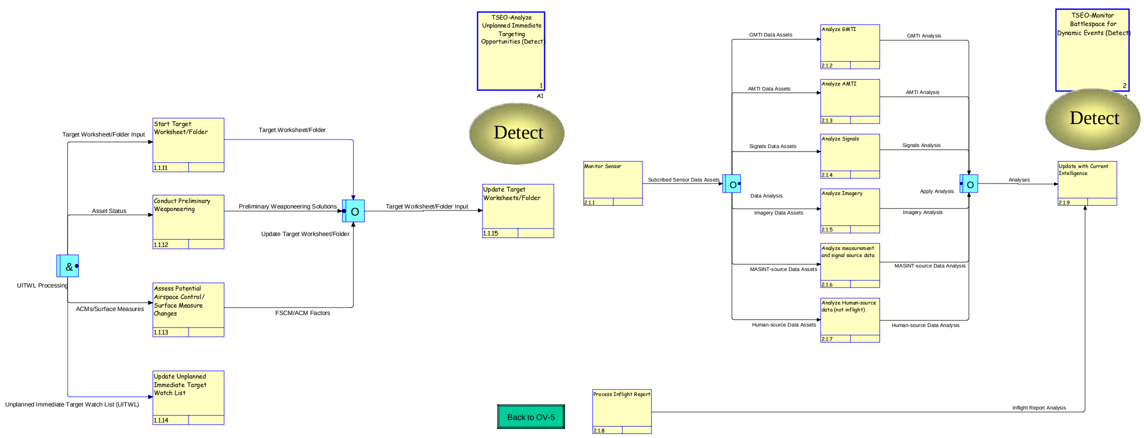

Figure 21: FFBD interactions

In moving to a level 1 FFBD, Figure 22 shows how to reduce the parallel threads as they are part of TSEO 1.2, yielding a sequential representation. The same can be done for the

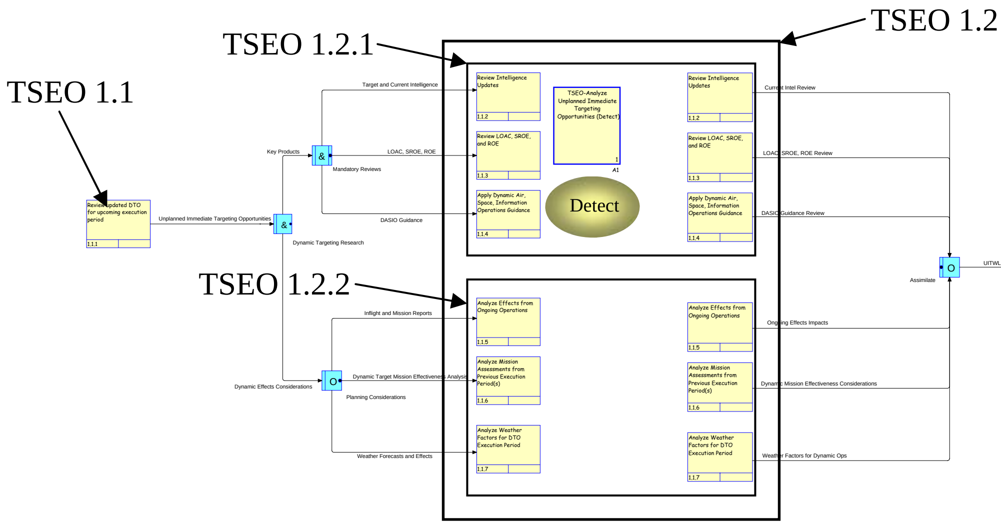

Figure 22: Reducing the FFBD from level 2 to level 1

following subfunctions in Figure 23. Again at level 1, the model is reduced to a sequential order. The simplification of the TSEO 2.0 function is shown in Figure 24, which represents an interesting occurrence as there is a redundant representation of

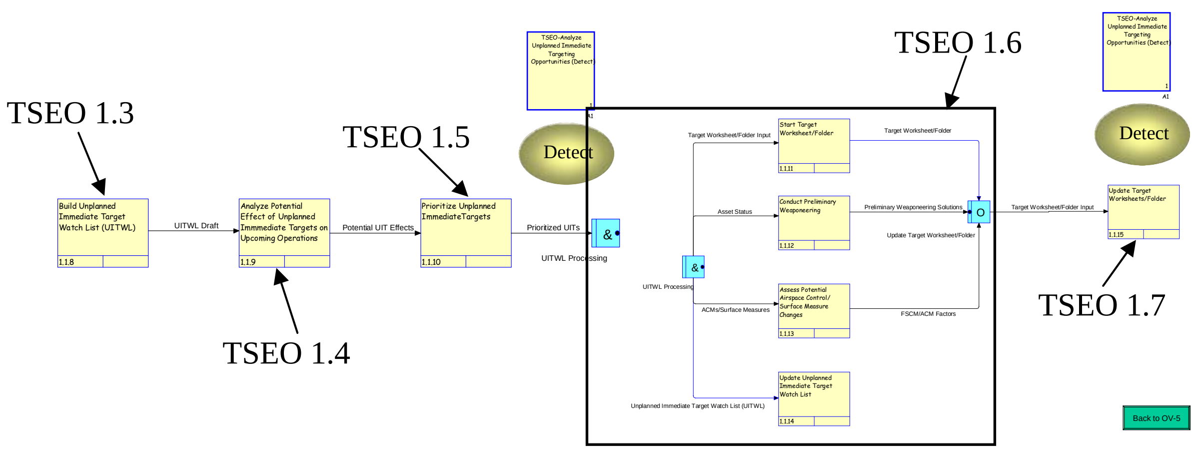

Figure 23: FFBD reduction from level 2 to level 1

TSEO 2.1, Monitor Sensor. They will be considered as one single TSEO 2.1 and this will yield a parallel configuration with TSEO 2.2, Process Inflight Report. It will be assumed that TSE0 1.7 feeds both TSEO2.1 and TSEO 2.2 at the same time. But as can be seen there is no construct to determine if both functions are needed to continue or if one is

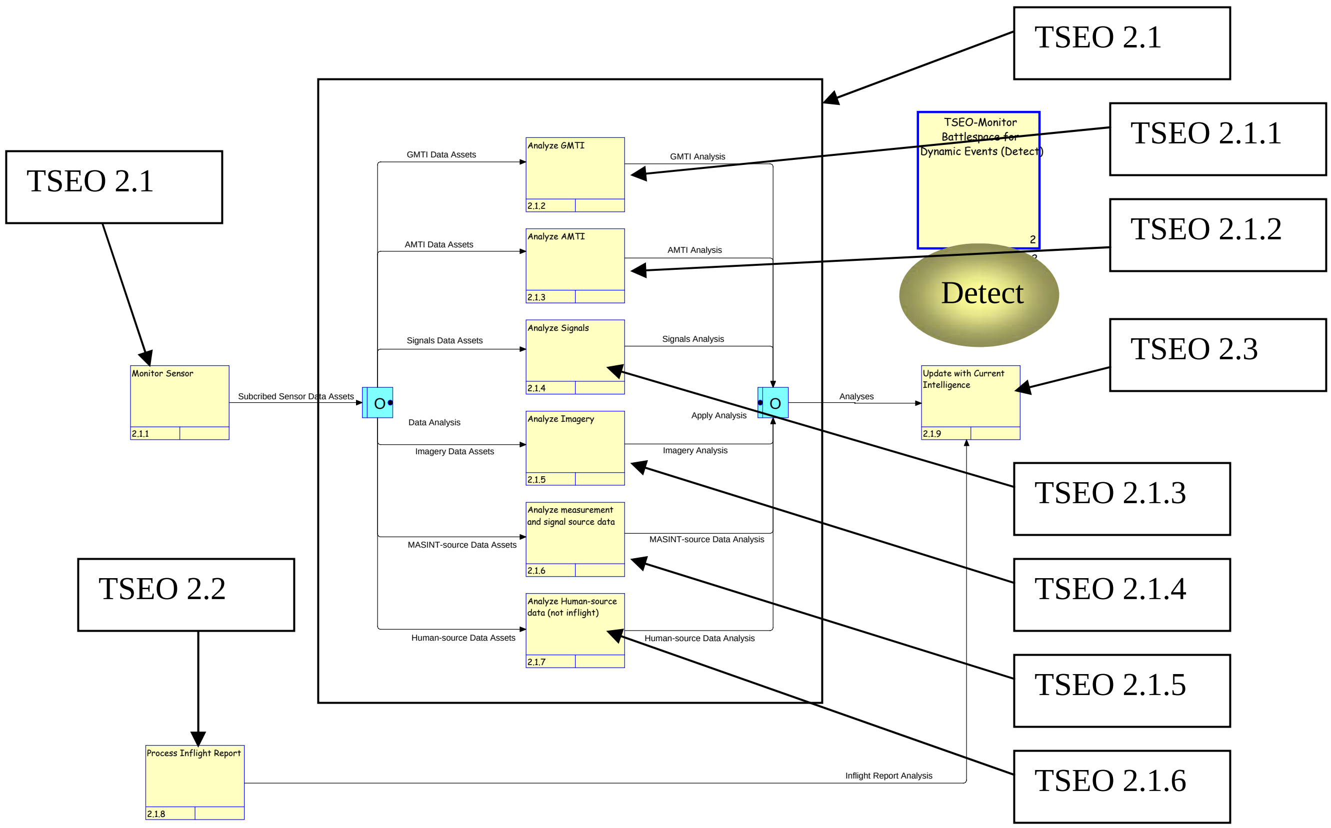

Figure 24: Irregularities in TSEO 2.0 consolidation

sufficient to continue. Because both functions handle dynamic information we will assume that either one is sufficient to continue the execution. Figure 25 shows the simple reduction of functions TSEO 2.3 to TSEO 2.6. In this figure we have an exit function that we assume goes to reference. Figure 26 represents the reduction of TSEO 3.1. Most of the subfuctions are in parallel and only one is needed to continue the execution of the

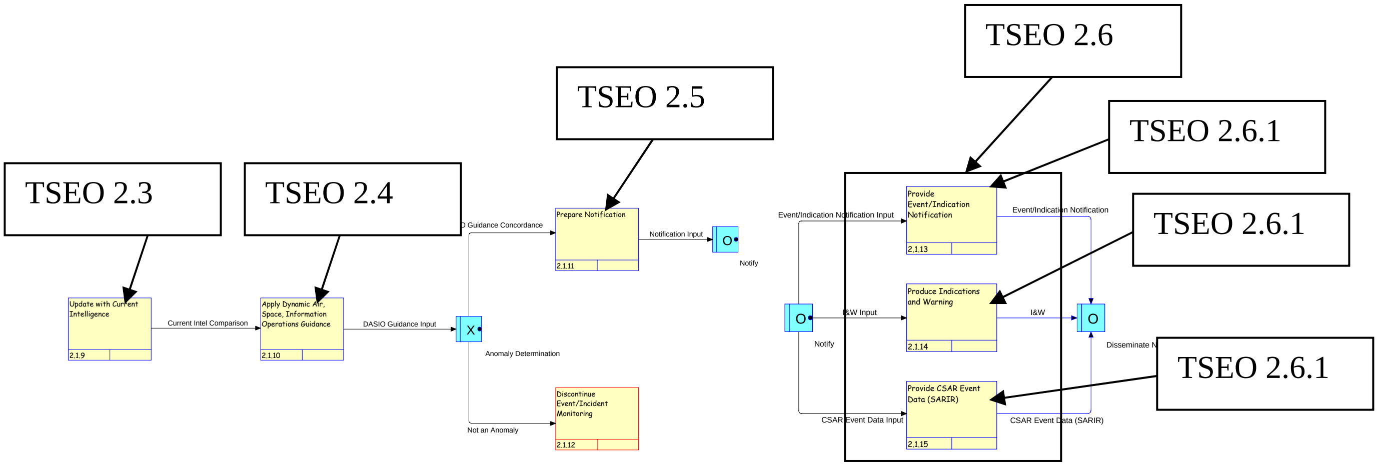

Figure 25: Cont. TSEO 2.0 Monitor Battlespace for Dynamic Events

other functions. TSEO 3.1 feeds directly to TSEO 3.2 see Figure 27. As shown there is an anomaly with function TSEO 3.3, the exclusion construct splits the paths of function

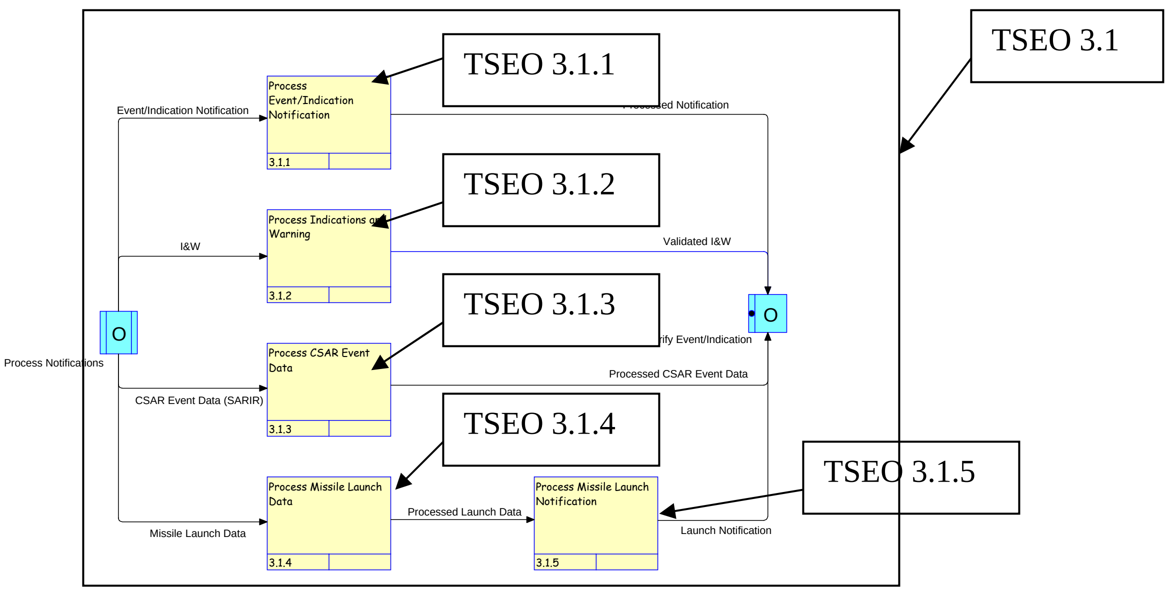

Figure 26: TSEO 3.0 Verify Event/Indication is of Interest

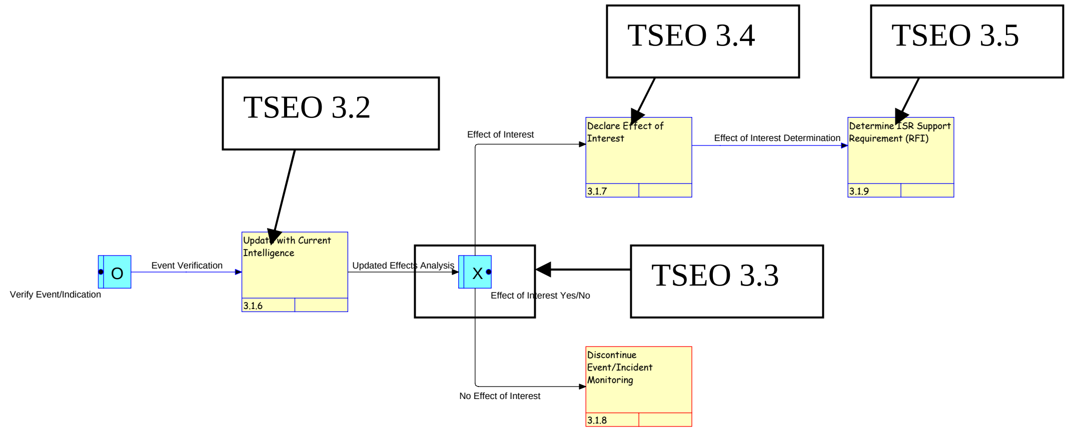

Figure 27: Cont. TSEO 3.0 Verify Event/Indication is of Interest

TSEO 3.3. The construct has two alternatives which continue to TSEO 3.4 or exit to reference. Figure 28 depicts the start of reduction of the TSEO 4.0 function. The underlying subfunction structure of the TSEO 4.0 has multiple ramifications that make up

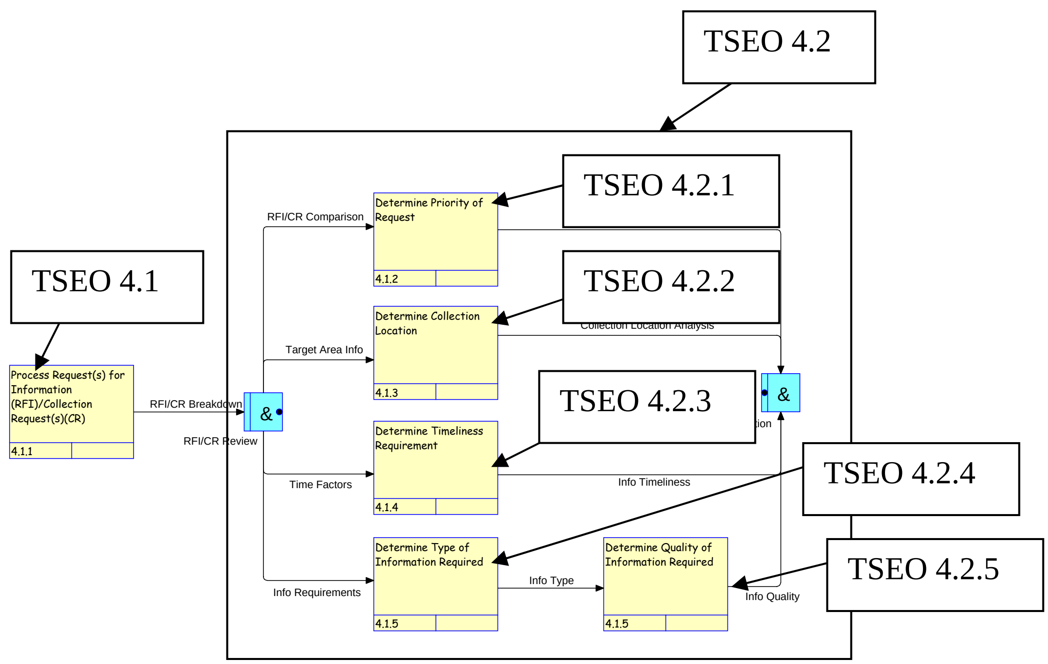

Figure 28: TSEO 4.0 Adjust ISR to Support Dynamic/Time Sensitive Effects Operations

the logic of this top level function. The branching starts after the TSEO 4.0 function, see Figure 29, where it branches to the TSEO 4.4 function (Figure 30) or functions TSEO 4.3.4 and 4.3.5 (Figure 36). The branching is internal to the TSEO 4.0 function and represents the different alternatives, see Figure 30 to Figure 37. All alternatives end in the same exiting subfunction TSEO 4.10.

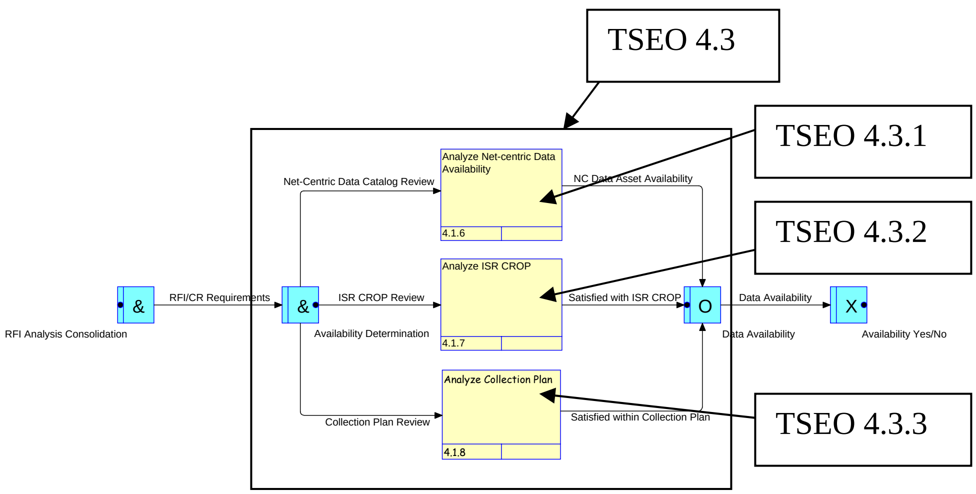

Figure 29: Cont. TSEO 4.0 Adjust ISR to Support Dynamic/Time Sensitive Effects Operations

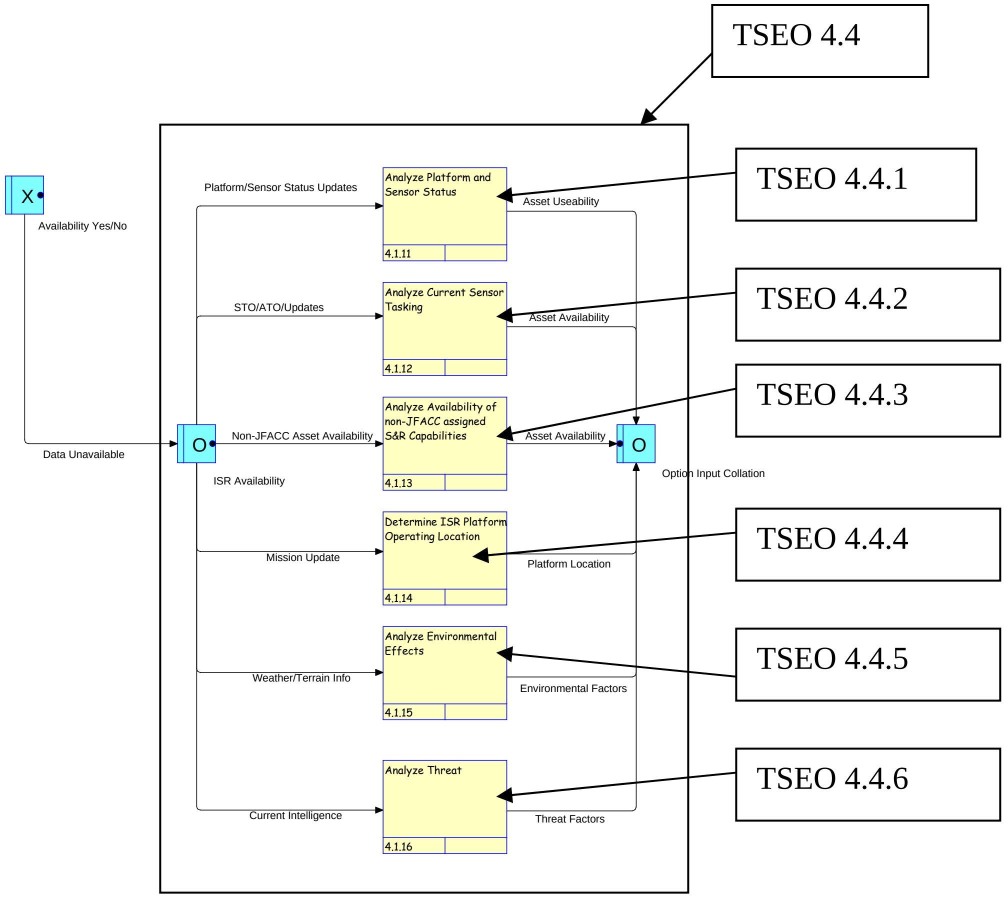

Figure 30: Cont. TSEO 4.0 Adjust ISR to Support Dynamic/Time Sensitive Effects Operations

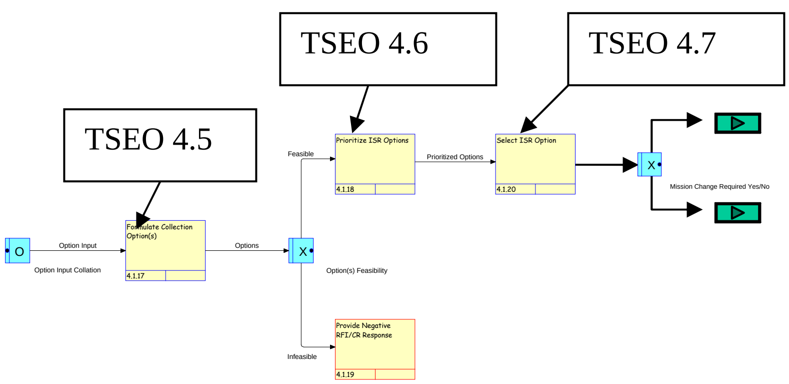

Figure 31: Cont. TSEO 4.0 Adjust ISR to Support Dynamic/Time Sensitive Effects Operations

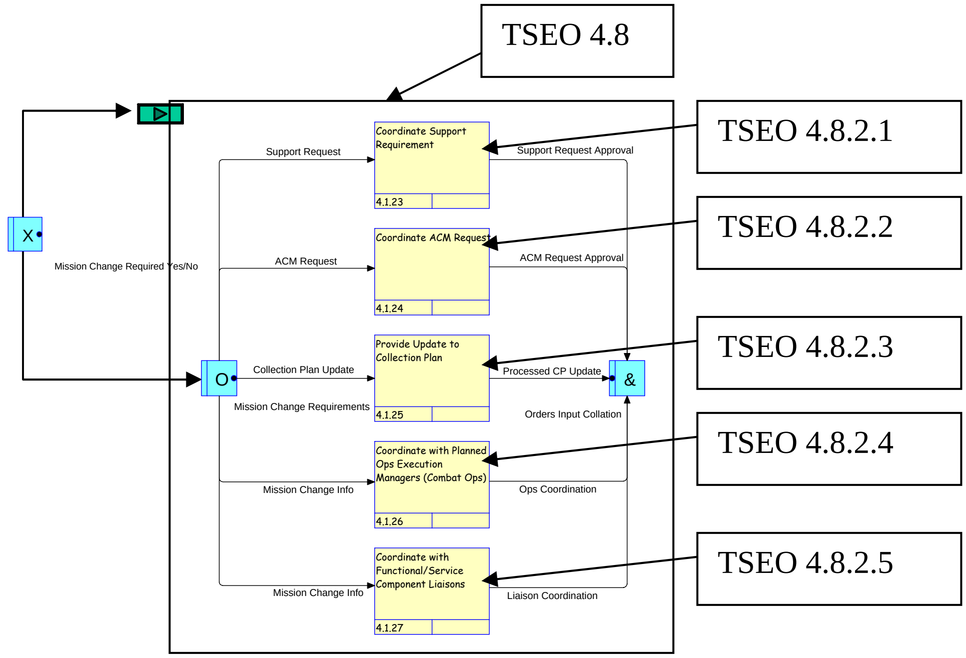

Figure 32: Cont. TSEO 4.0 Adjust ISR to Support Dynamic/Time Sensitive Effects Operations

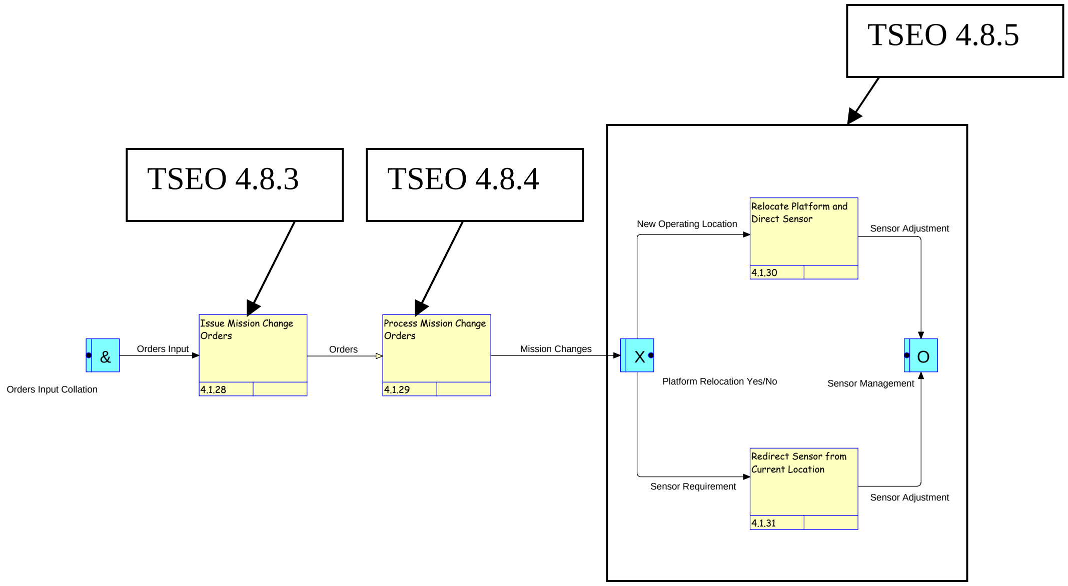

Figure 33: Cont. TSEO 4.0 Adjust ISR to Support Dynamic/Time Sensitive Effects Operations

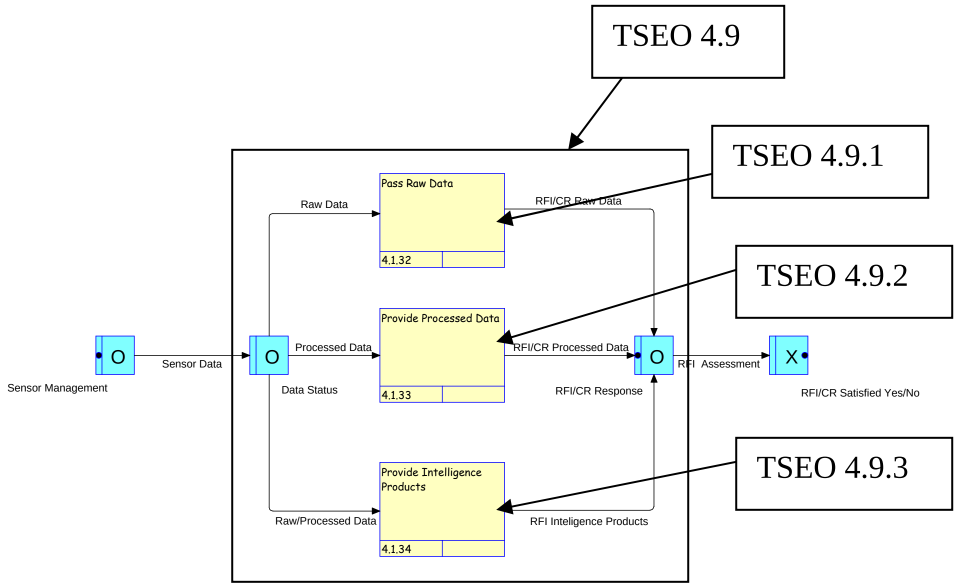

Figure 34: Cont. TSEO 4.0 Adjust ISR to Support Dynamic/Time Sensitive Effects Operations

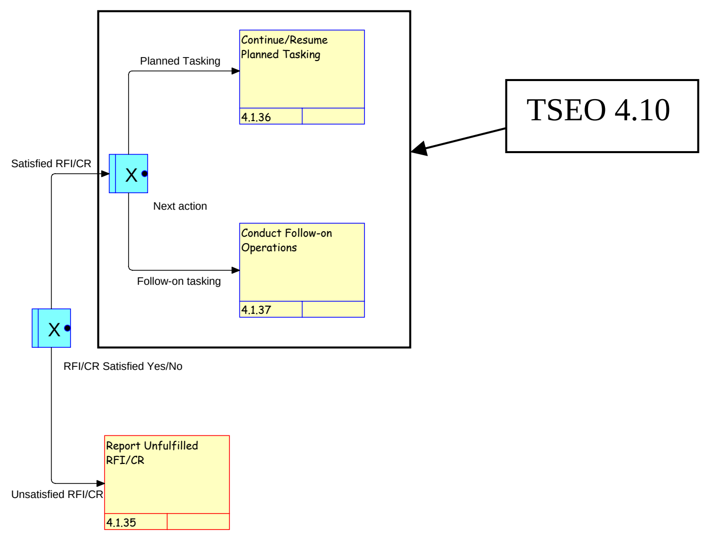

Figure 35: Cont. TSEO 4.0 Adjust ISR to Support Dynamic/Time Sensitive Effects Operations

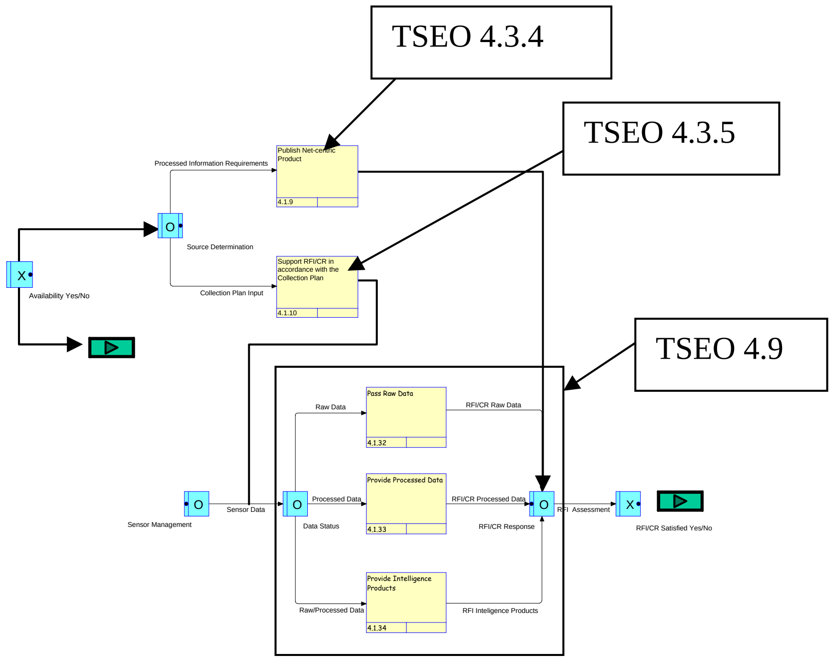

Figure 36: Cont. TSEO 4.0 Adjust ISR to Support Dynamic/Time Sensitive Effects Operations

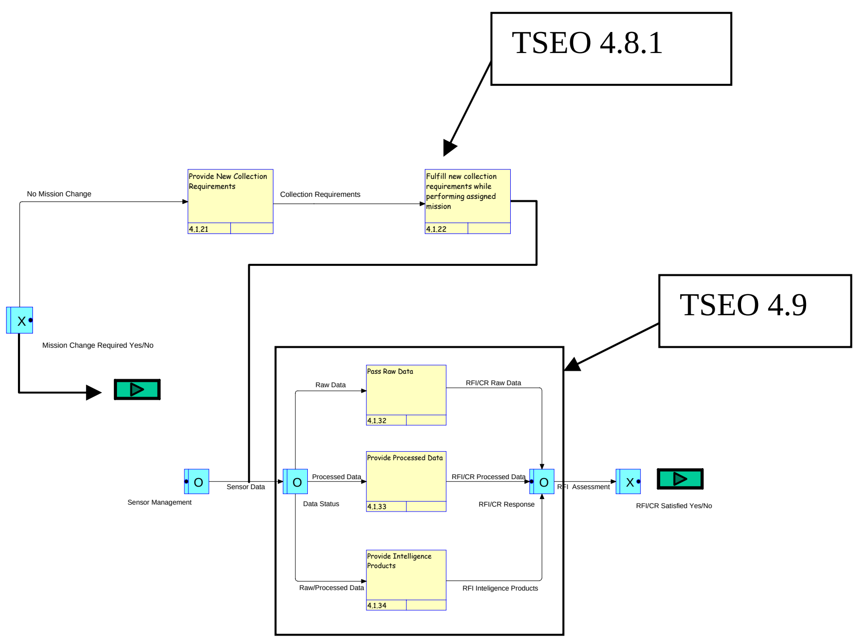

Figure 37: Cont. TSEO 4.0 Adjust ISR to Support Dynamic/Time Sensitive Effects Operations

As can be seen in Figure 38 the reduction of TSEO 5.0 is very simple due to its small number of functions. TSEO 6.0 decomposition starts in Figure 39. It has a number of parallel functions that end in an “or” construct that branches to TSEO 7.0 or TSEO 8.0.

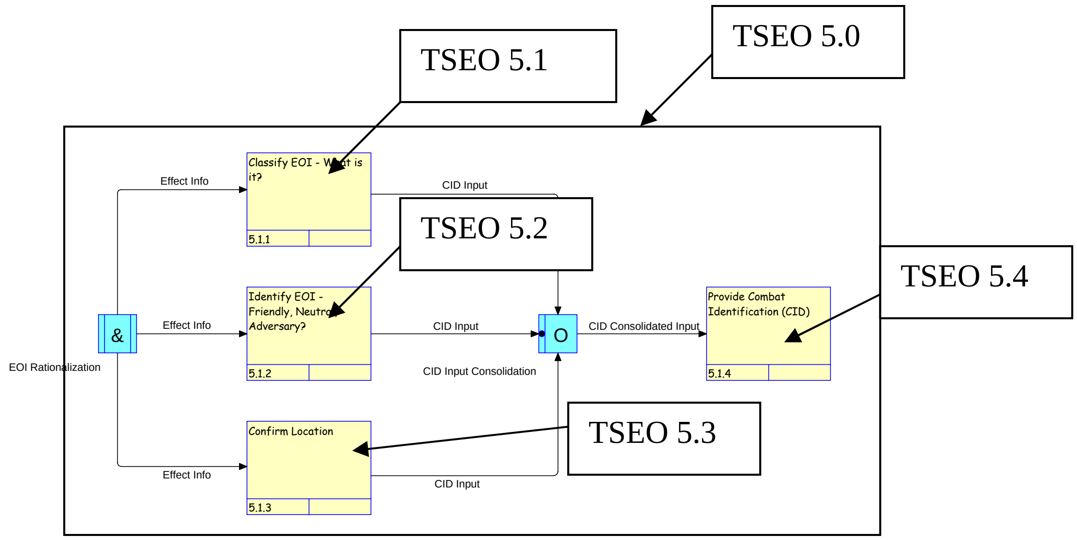

Figure 38: TSEO 5.0 Conduct Combat Identification

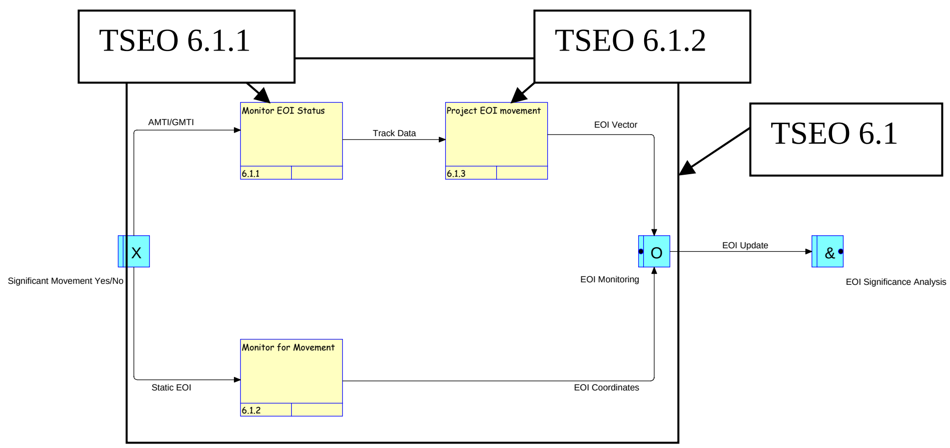

Figure 39: TSEO 6.0 Predict Effect/Urgency

For the first attack the branch selected leads to the TSEO 7.0 function and for every consecutive attack, the branch selected leads to TSEO 8.0. This can be observed from Figure 39 to Figure 41, where Figure 41 contains the “or” construct.

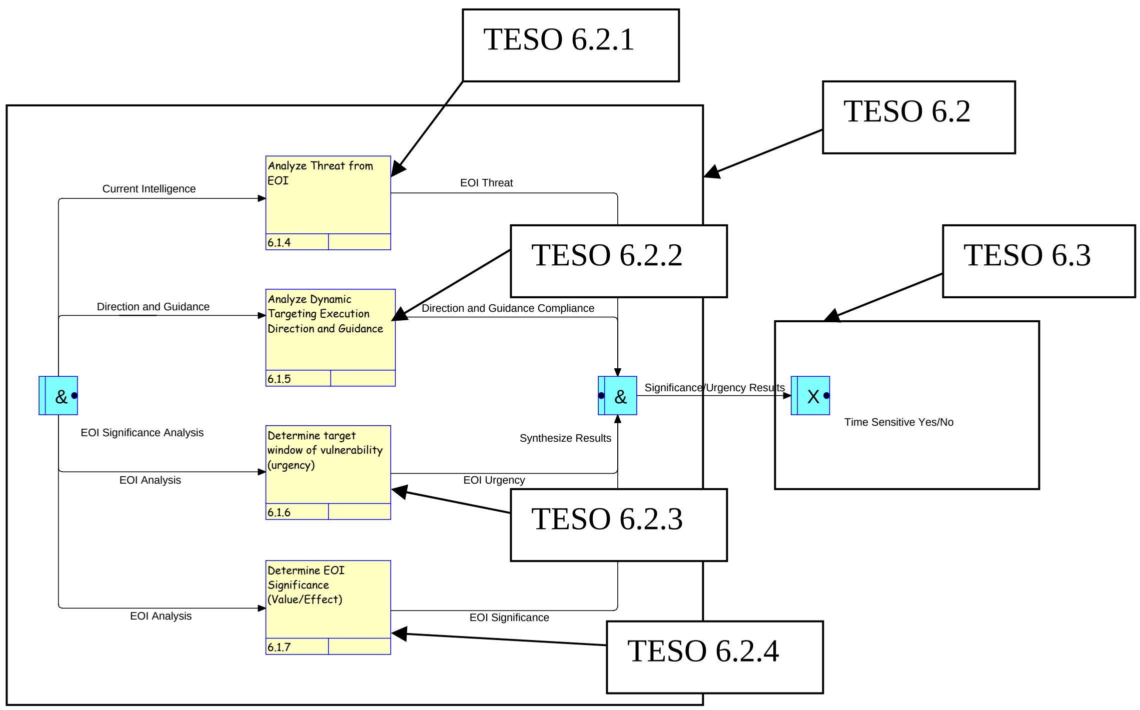

Figure 40: Cont. TSEO 6.0 Predict Effect/Urgency

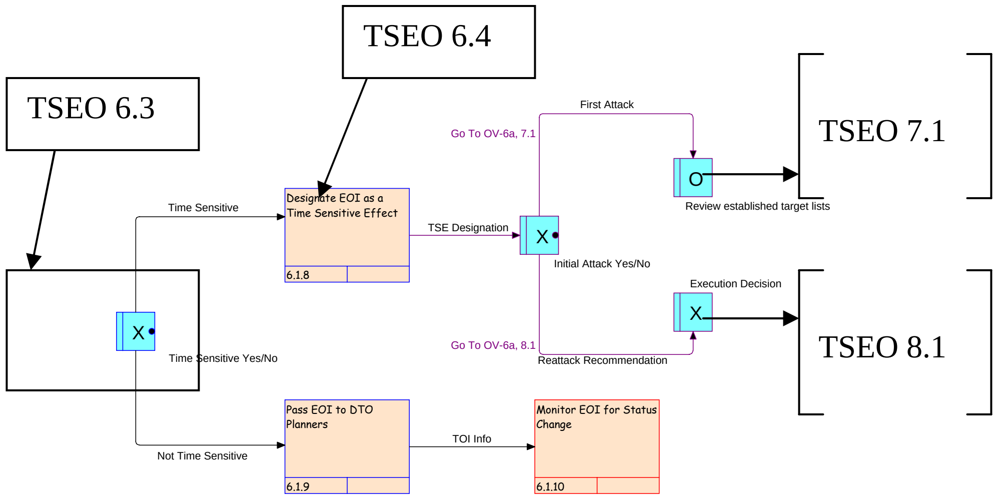

Figure 41: Cont. TSEO 6.0 Predict Effect/Urgency

TSEO 7.0, Validate Desired Effect, is the function that it is circumvented, in the original architecture, to expedite the reattack actions. The subfuctions that compose TSEO 7.0 can be observed on Figure 42 to Figure 45. The logic structure is somewhat confusing

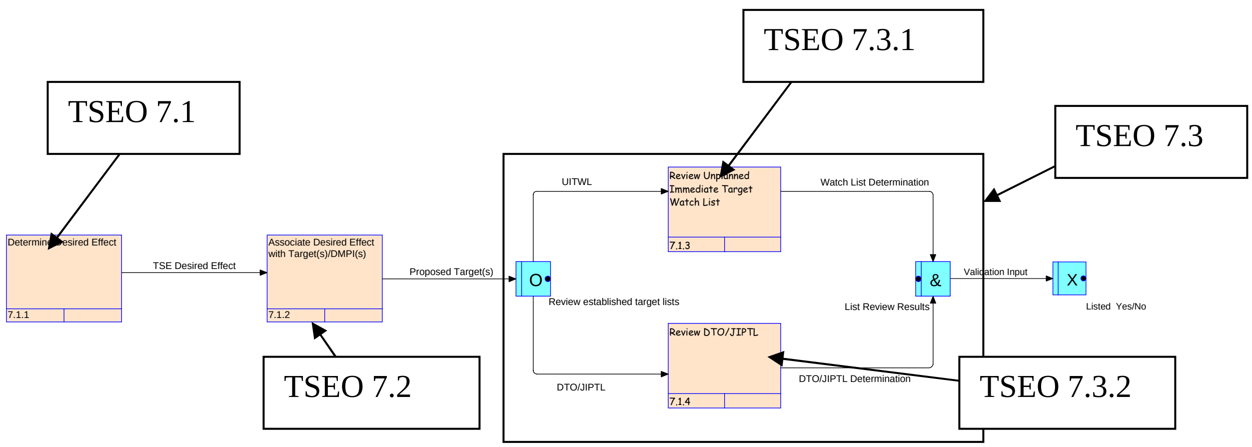

Figure 42: TSEO 7.0 Validate Desired Effect

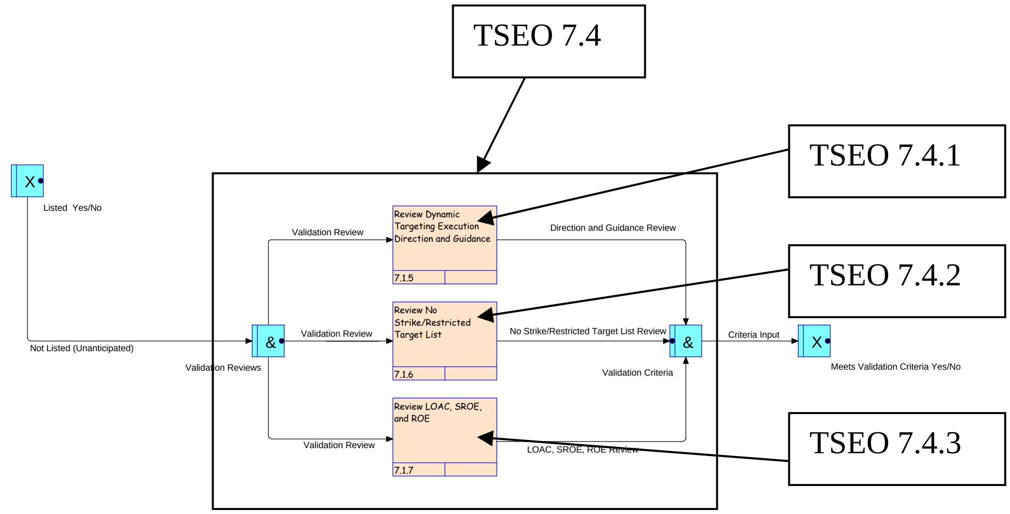

Figure 43: Cont. TSEO 7.0 Validate Desired Effect

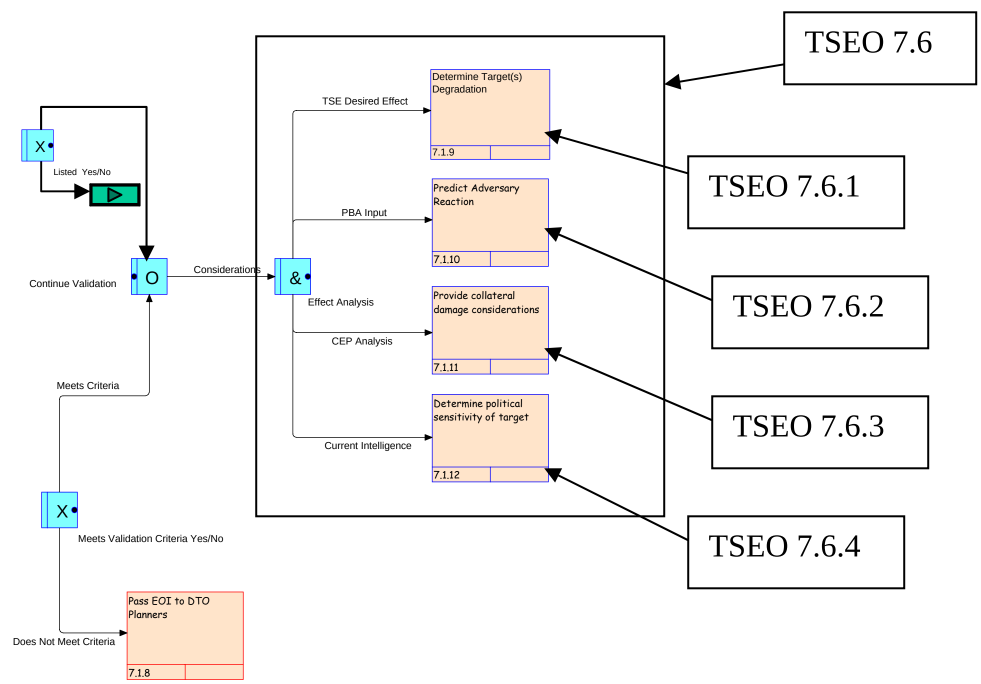

Figure 44: Cont. TSEO 7.0 Validate Desired Effect

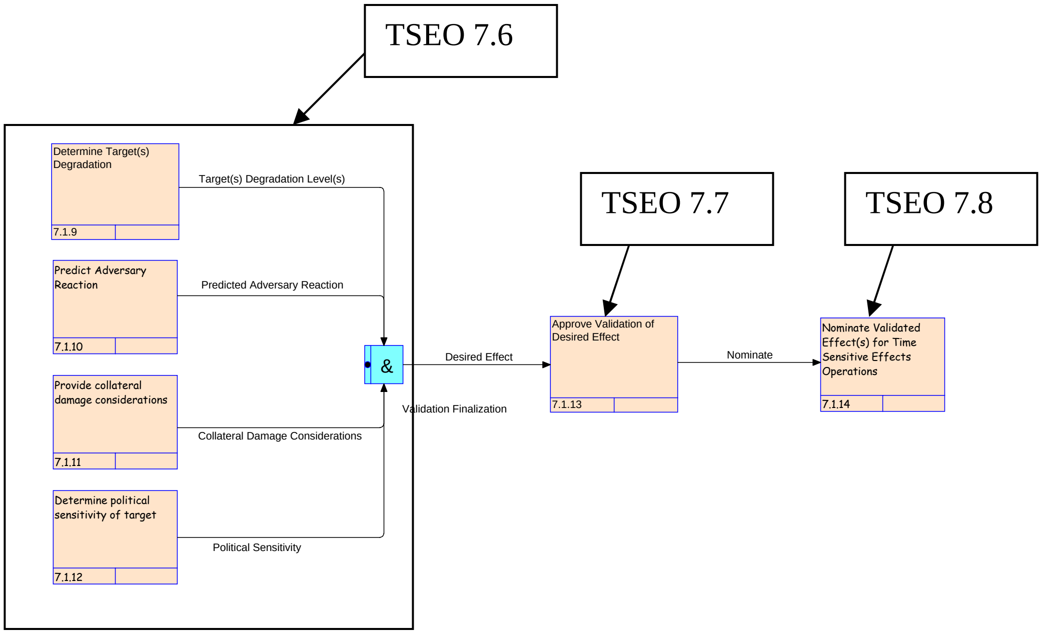

Figure 45: Cont. TSEO 7.0 Validate Desired Effect

but all branches lead to the same exit TSEO 7.8 function. TSEO 8.0, Nominate Engagement Option, is a number of groupings of parallel functions connected in series.

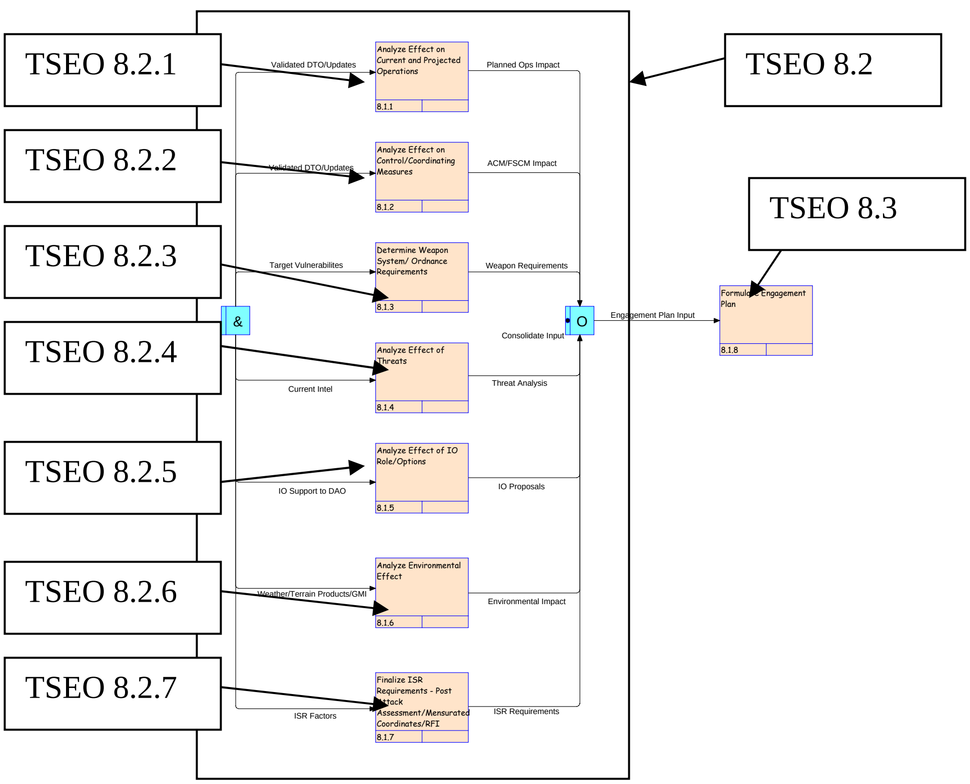

Figure 46: TSEO 8.0 Nominate Engagement Option

Compared to the previous functions, TSEO 8.0 has a straightforward logic construct. The TSEO 8.0 functions can be observed on Figure 47 to Figure 49. The functional

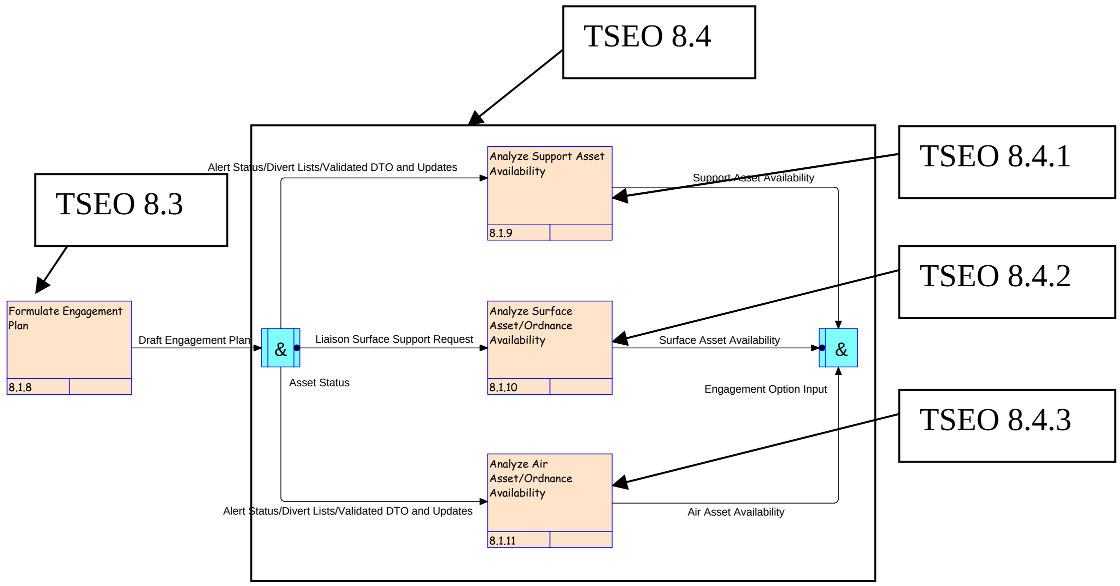

Figure 47: Cont. TSEO 8.0 Nominate Engagement Option

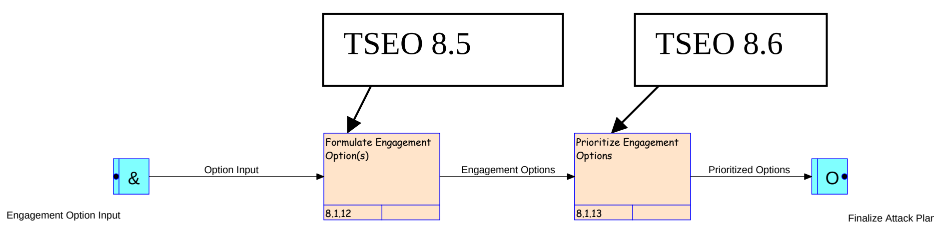

Figure 48: Cont. TSEO 8.0 Nominate Engagement Option

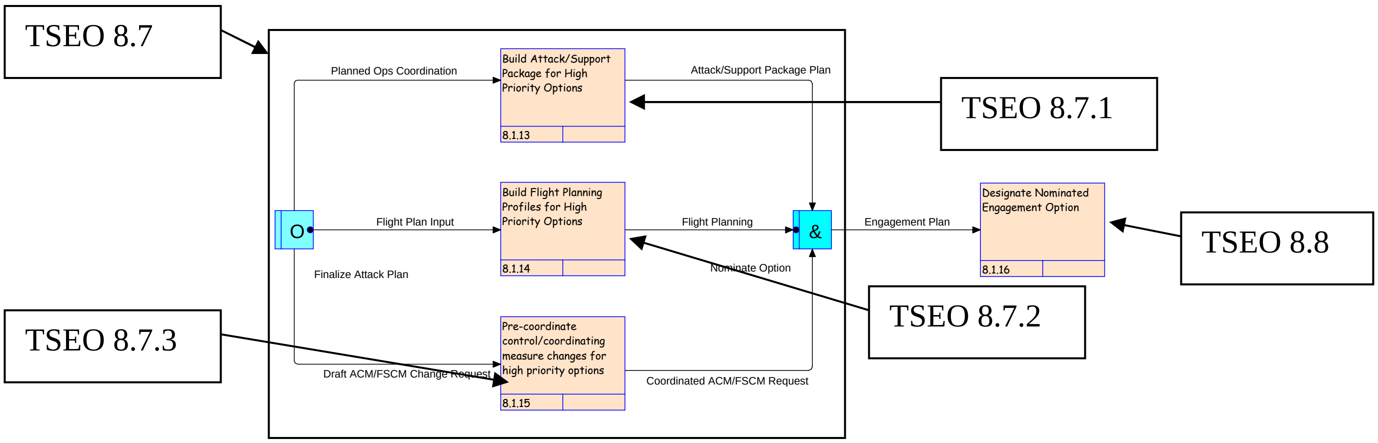

Figure 49: Cont. TSEO 8.0 Nominate Engagement Option

decomposition for TSEO 9.0 starts in Figure 50 and ends in Figure 52. TSEO 9.0 has an exit point to reference which can be seen on Figure 50. The functions end on TSEO 9.5. This function is a collection of parallel subfuctions; unfortunately, there is no guidance of how the functions of TSEO 9.5 connect to the following function (see Figure 52).

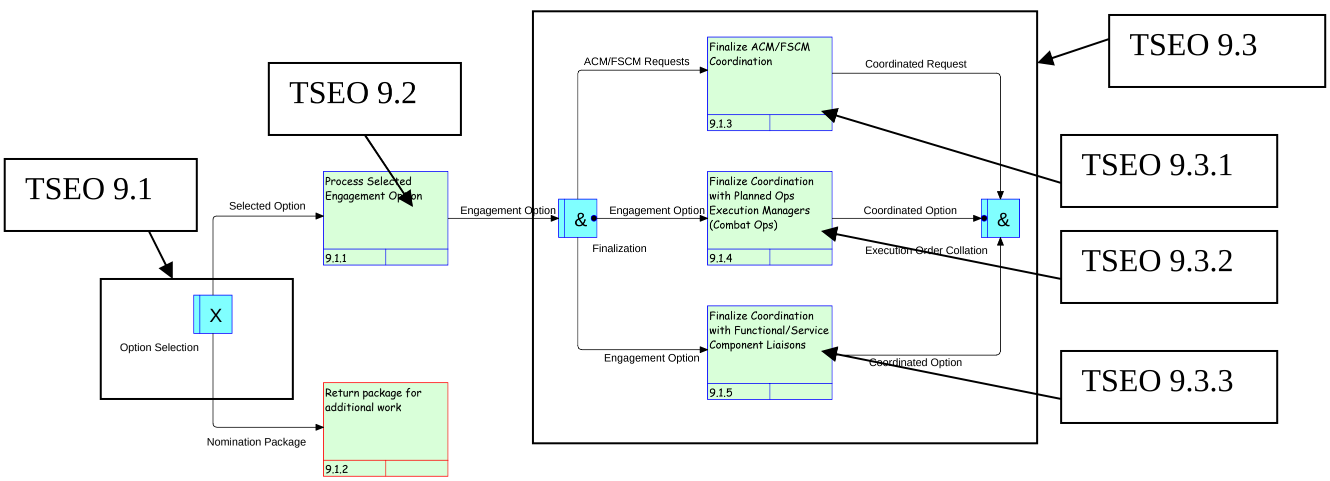

Figure 50: TSEO 9.0 Manage Engagement

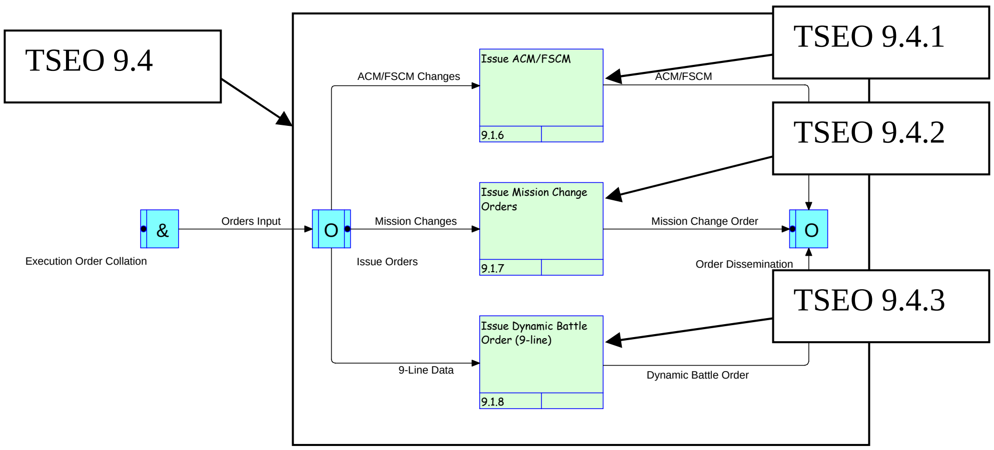

Figure 51: Cont. TSEO 9.0 Manage Engagement

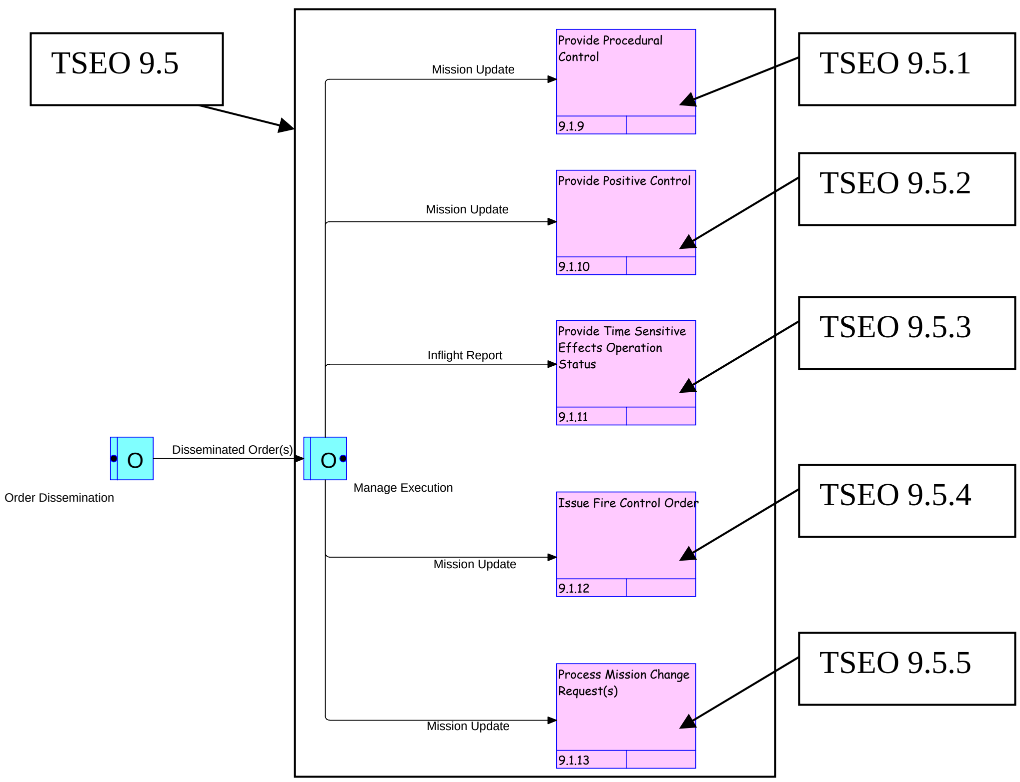

Figure 52: Cont. TSEO 9.0 Manage Engagement

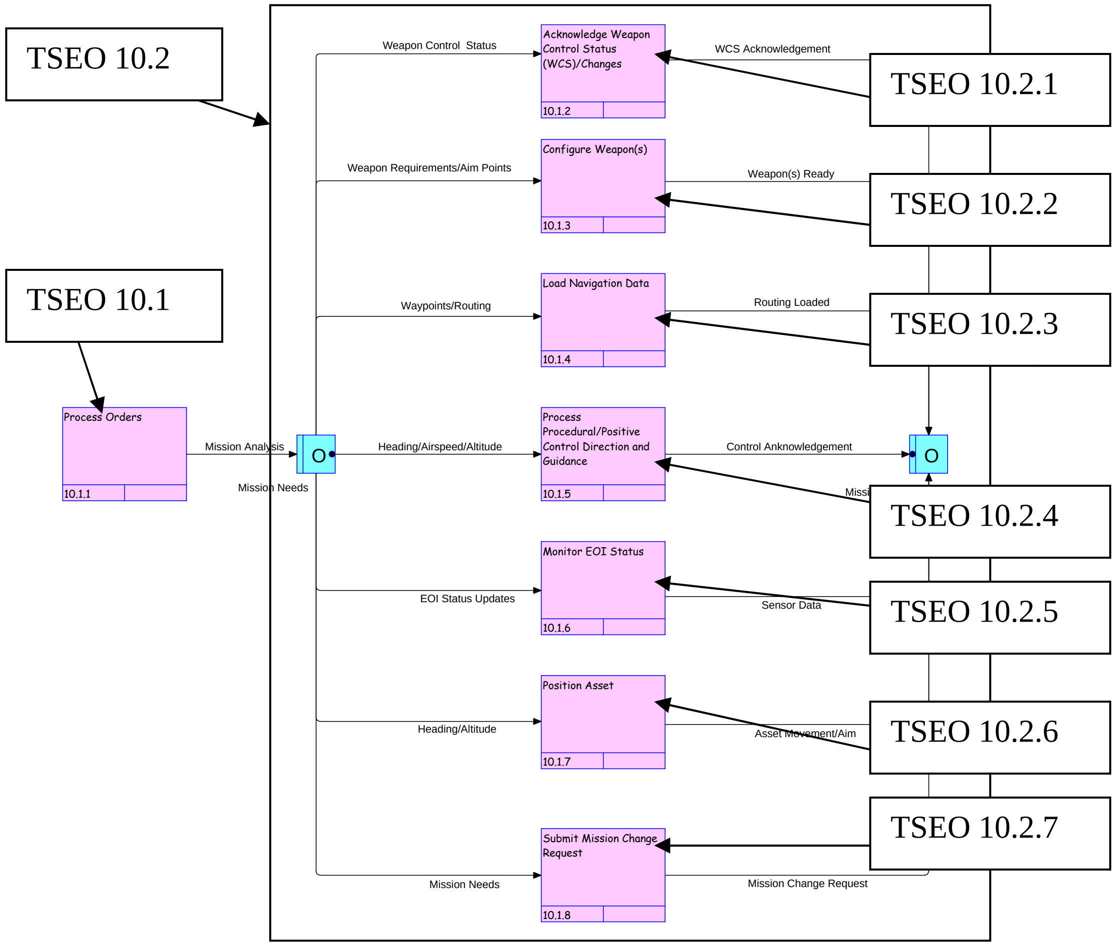

Figure 53: TSEO 10.0 Produce Effect

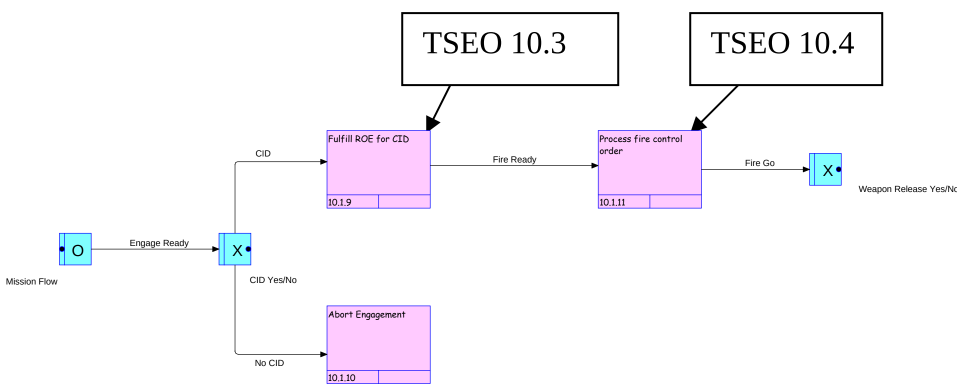

Figure 54: Cont. TSEO 10.0 Produce Effect

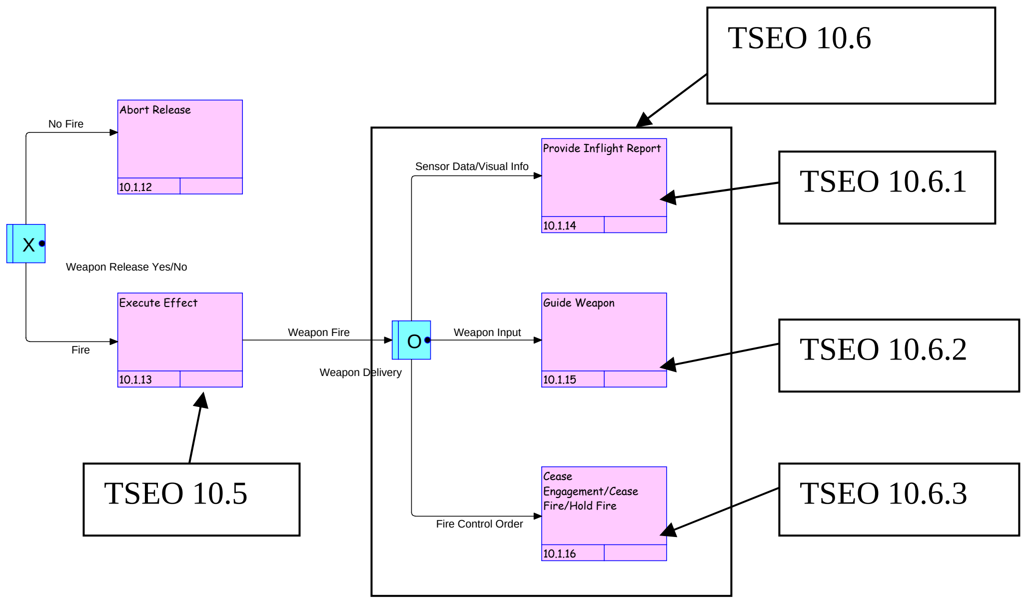

Figure 55: Cont. TSEO 10.0 Produce Effect

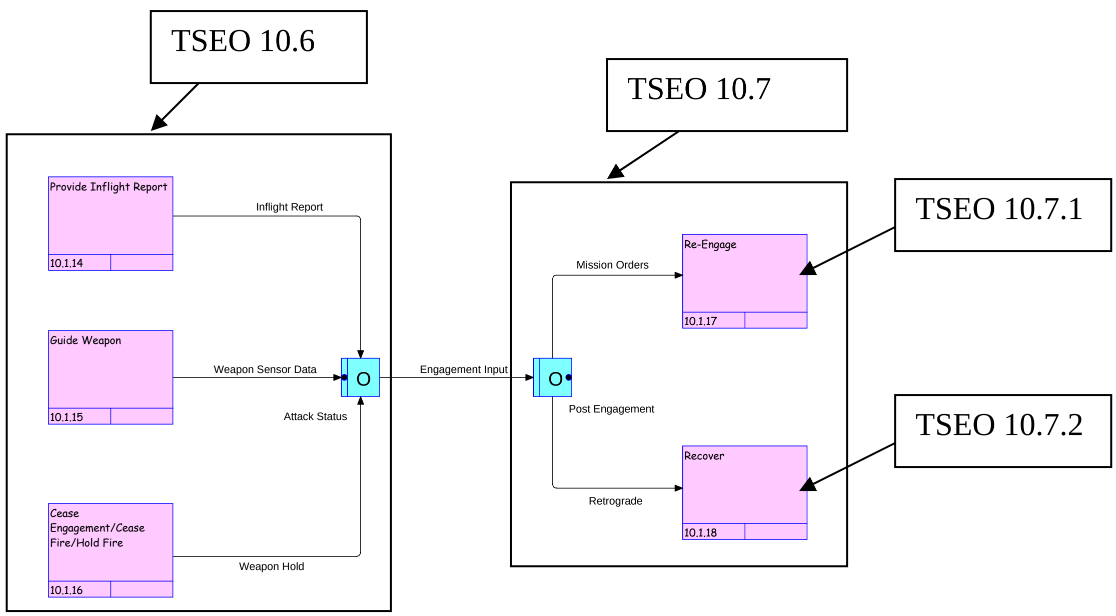

Figure 56: Cont. TSEO 10.0 Produce Effect

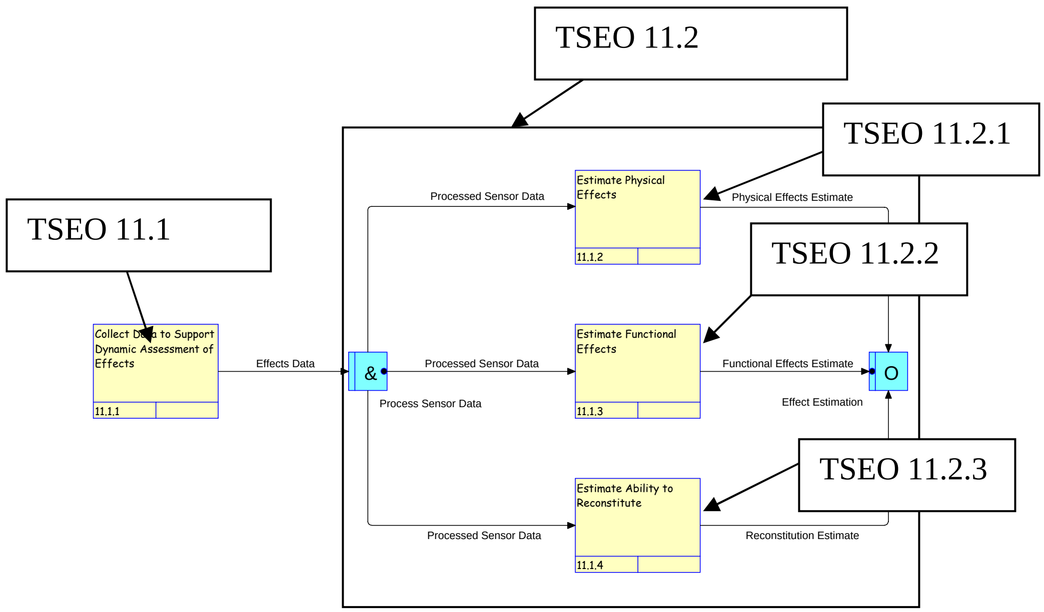

Figure 57: TSEO 11.0 Conduct Dynamic Assessment of Effect

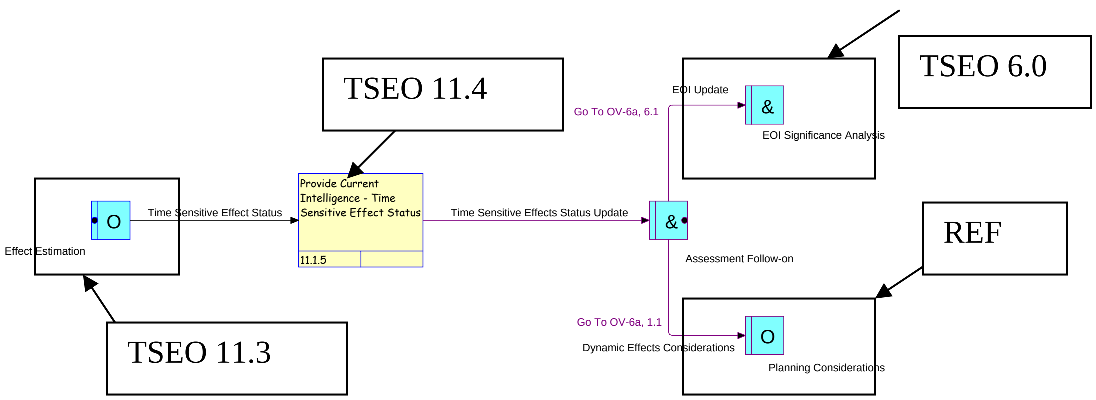

Figure 58: Cont. TSEO 11.0 Conduct Dynamic Assessment of Effect

Figure 59 represents the top level FFBD of the baseline TSEO 2012. This figure shows two constructs that are important to the executable model. The first one is the conditional exit criteria from TSEO 6.0 that allows the system to bypass function TSEO 7.0 when performing a re-attack order. The second construct is the loop feature after function TSEO 11.0. The loop will be active as long as the target is considered alive. The most important functions for executable development will be TSEO 6.0, TSEO 10.0, and

TSEO 11.0. TSEO 6.0 must determine when the executable is performing a re-attack loop. The damage will be calculated on TSEO 10.0, the target status and executable termination is determined on the function TSEO 11.0.

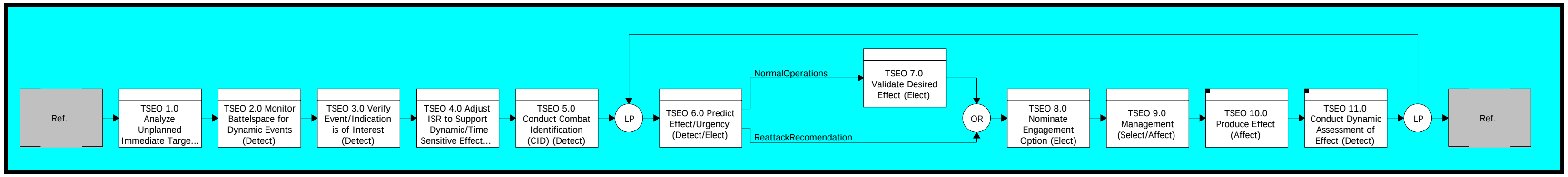

Figure 59: Top level FFBD

Core Environment

Having a more comprehensive view of the architectural logic, the creation of the executable model can begin. The first step is to import all the functions generated for the IDEF0 on CORE™ [Vitech02] to the Enhanced FFBD CORE™ environment. The import from IDEF0 also makes available the inputs, outputs, resources, and triggers associated with each function. By default, CORE™ displays the top level functions in a serial configuration. It is the responsibility of the investigator to arrange the functions in a configuration that supports the logic of the architecture.

This study will concentrate on the thread that supports the re-strike capabilities of the architecture. In doing so, only part of the architecture will be simulated. This limits the simulation to a top level perspective, saving extensive time in modeling. Each function will have a time duration drawn from a probability distribution. In essence we are going to do a first level FFBD (TSEO X.X) to determine the logic of our model, but it is represented and executed in the Enhanced FFBD as a level zero (TSEO X.0). For some of our critical functions we will go to a level one to develop the underlying algorithms that will determine the behavior of the system.

The top level FFBD is relatively simple to generate in CORE™. Once the functions have been transferred to the EFFBD CORE™ environment we need to arrange the functions in logical order before we can introduce any logical constructs. The first logical construct is to create two exit criteria for function TSEO 6.0. The first exit branch will connect function TSEO 6.0 to TSEO 7.0, the second one will bypass TSEO 7.0 and connect TSEO 6.0 directly to TSEO 8.0. The bypass will be activated on a re-attack loop, as the desired effect is assumed to be still valid. This rule can be observed in Figure 41. The exit criteria for this logical construct can be observed in Table 14 and the FFBD construct can be observed in Figure 60. The remaining function blocks are set in series.

Table 14: Exit criteria for TSEO 6.0

Criteria	Exit Condition	Criteria	Exit Condition	Criteria	Exit Condition	Criteria
Is the target Time Sensitive?	No	Exit to REF				
	Yes	Is it a new target?	Yes	Continue to TSEO 7.0		
			No	Does the target require immediate re-attack?	No	Continue to TSEO 7.0
					Yes	Continue to TSEO 8.0

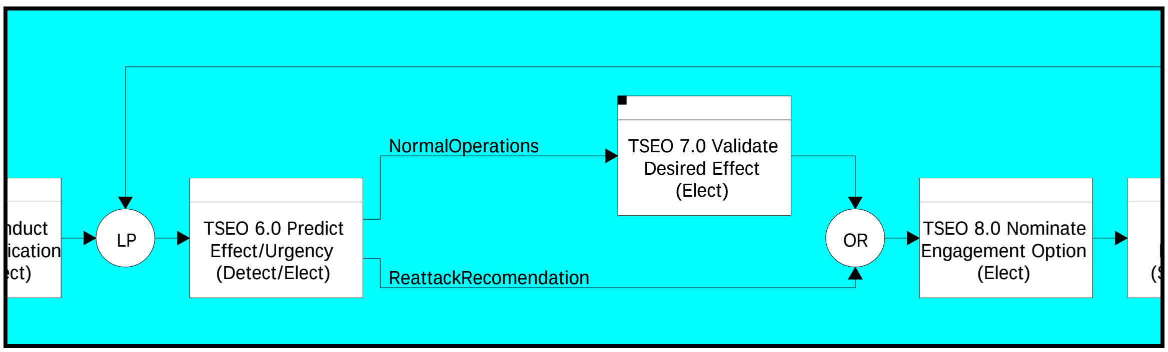

Figure 60: TSEO 6.0 exit construct

The second logical construct is housed in function TSEO 10.0, where kinetic energy is applied to the target. The underlying construct can be observed in Figure 61. The function “time delay 3” sets the statistical parameters for the time duration of the top level function TSEO 10.0, as the remaining factions will be set to zero duration. The “Effect” function will determine the target level health and its exit criteria will assess the actual damage to the target. The BDA damage assessment algorithm is integrated into function TSEO 11.0; see Figure 62. “Time Delay 4” will impart a stochastic time duration distribution to the top level function TSEO 11.0, the “Sensor” function has zero duration. The exit criteria from function “Sensor” will determine the status of the targets health. The exit criteria can be seen on Table 15. To change the current FFBD to support

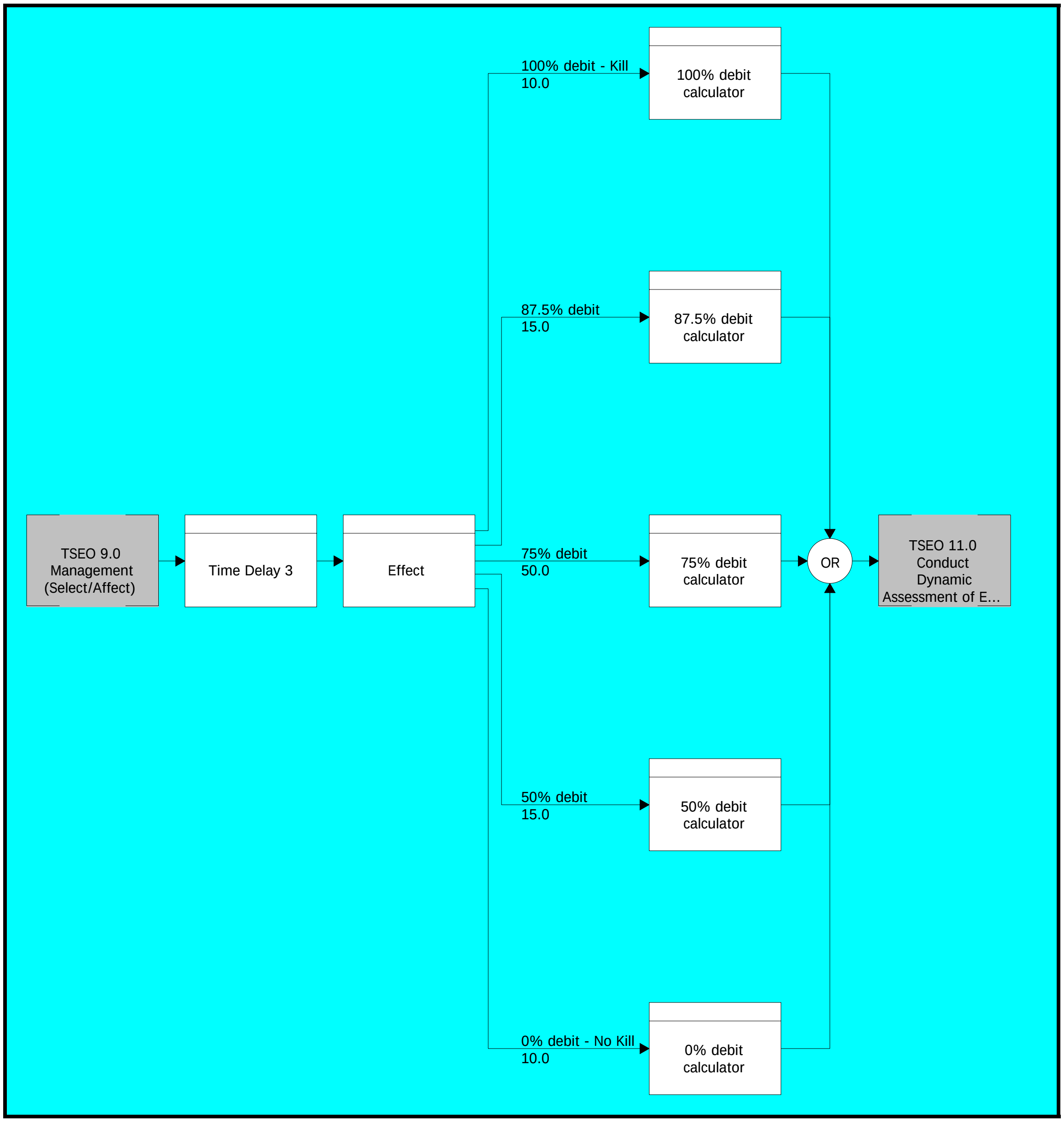

Figure 61: Effect assessment algorithm

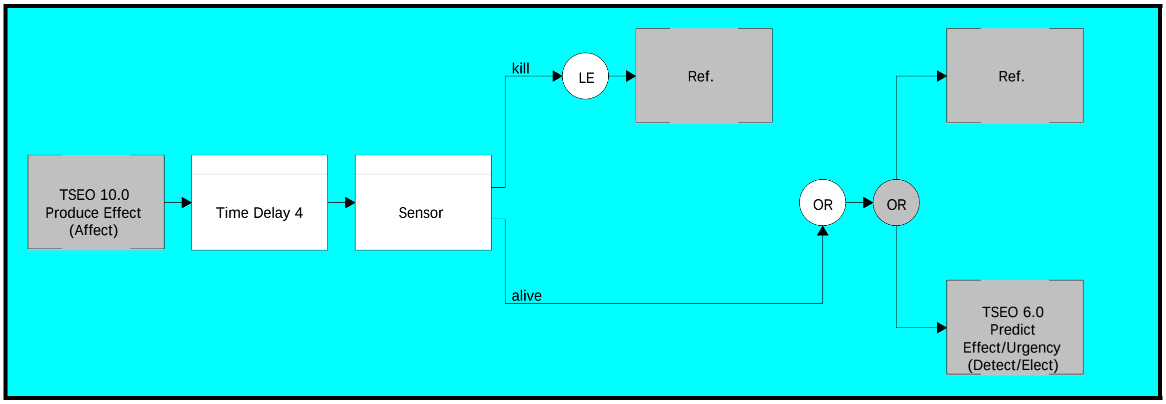

Figure 62: BDA damage assessment algorithm

Table 15: Exit criteria for "Sensor" function

Criteria	Exit Condition	Criteria
Is the TargetHealth < 20%?	Yes	Exit to REF
	No	Continue to re-strike loop

WBBDA we need to add an iteration construct that houses the following functions (see Figure 63); TSEO 9.0, TSEO 10.0, and TSEO 11.0. The iteration construct allows for the shortened decision cycle obtained through WBBDA. A quick assessment of the damage can be obtained by the attacking assets, allowing rapid reallocation resources to the

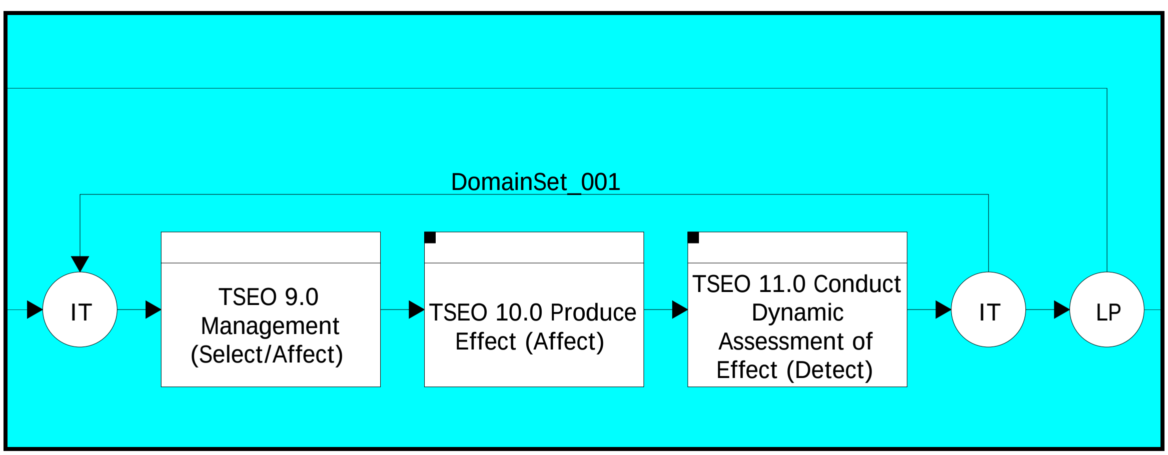

Figure 63: Iteration construct for modified FFBD

target. To further emphasize the quickness of the assessment, the stochastic time delay that governs the TSEO 11.0 function is half of the original time delay. Five iterations will be the maximum iterations to be performed per attack loop cycle. The five iterations are meant to be representative of an aircraft carrying five warheads. After the iterations are completed the loop function will be activated, just like the original system, and a new WBBDA supported attack run will be performed. Figure 64 and Figure 65 are a comparison of the top level FFBDs of the two systems. As it can be seen the most

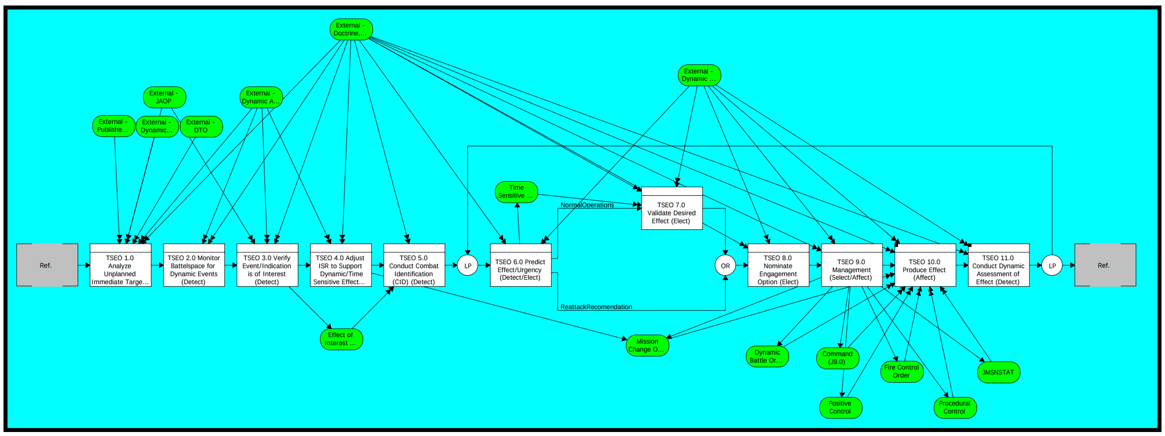

Figure 64: Top level original EFFBD

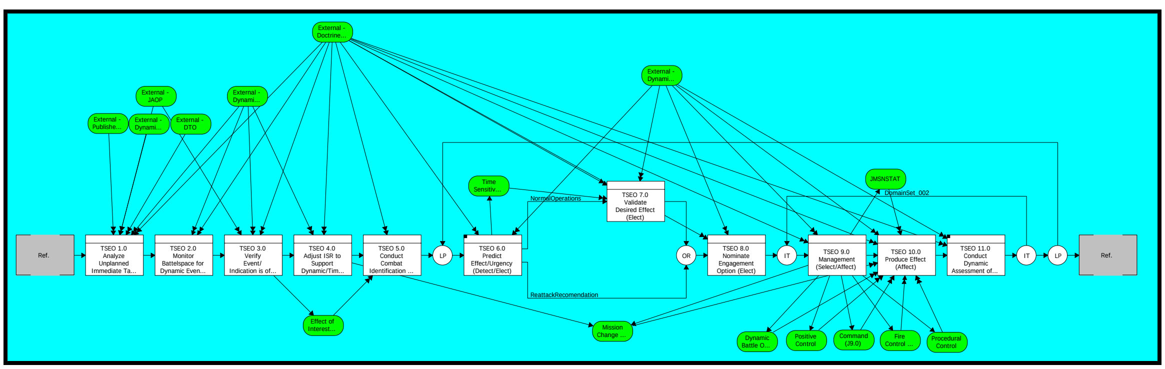

Figure 65: Top level modified EFFBD

notable difference is the iterating construct that encompasses the TSEO 9.0, TSEO 10.0 and TSEO 11.0 functions.

Summary

Through the analysis and modification of the rules model, it was determined there is no advantage in representing the rules model to the lowest levels available. The extensive breakdown of the FFBD forces the investigator to work with varying logic details across the executable model. This provides a more natural way of understanding the critical logic that governs the architecture. Once the appropriate FFBD levels were generated the appropriate level of detail for the executable model could be determined. It was decided to implement a top level executable model for a preliminary capability assessment of the viability of WBBDA.

With the breakdown of the different levels we can discern and model the parts necessary to evaluate the level of target destruction. Parts not modeled are given a stochastic time duration representative of that function only. No effort was made to implement other “what if?” scenarios (i.e. the loss of a link, loss of attacking asset). The focus is to determine the effect of WBBDA on sortie effectiveness. The functions were only simulated to the lowest level needed to capture their behavior; functions were discomposed enough to decipher the underlying logic that determines the system behavior.

To obtain the desired behavior of a weapon born BDA concept, we added an extra sub-function that determined the probability of kill by the use of a simple probability distribution. Two different warheads where selected, one with 50% lethality and the second with 75% lethality. This will provide some indication of the lethality for the

WBBDA concept. New data links where added to the architecture to simulate the shortening of the chain of events to perform a re-strike to a time sensitive target.

For an effective comparison and assessment of the WBBDA capabilities an unbiased metric needs to be adopted. The metric must be one that can directly compare the two architectures. For this metric we chose the number of sorties needed to destroy a target. By performing a Monte Carlo analysis we can develop a probability distribution for the number of sorties needed to destroy a target.

V. Results and Analysis

Test Set Up

Assets can be viewed as five distinct attack aircraft, an aircraft that carries five warheads, or a combination of both. To better depict WBBDA the TSEO 11.0 function has a reduced stochastic time component, the shorter time will benefit both threads (i.e., loop and iteration constructs). The amount of damage to the target will be determined in function TSEO 10.0. Targets are assumed stationary with the ability to regenerate with the passage of time. This is an artificial construct taken to highlight the time critical aspect of the target. It is important to mention that the ability to obtain shorter response time through the use of WBBDA is not in question; this is assumed. The amount of regeneration is dependent on the elapsed time until re-strike. For purposes of the executable model the target is considered destroyed when target integrity is below 20% of maximum value.

The executable marks the time in seconds but we will transform to minutes. The execution time for each function will be determined by a stochastic time variable.

The number of sorties per kill is used as the comparison metric between the performance of the baseline and the modified system. There will be four data points; each data point will be provided by a different executable model or valve for warhead lethality. For purpose of obtaining good statistical data we will perform one hundred Monte Carlo simulations per data point. A 95% confidence level will be applied to the test data and respective error bounds will be calculated. The data will be decomposed by histogram and a best statistical fit will be found.

To start the executable model we need to generate some global function triggers (see Figure 66). These triggers are a fallout of the IDEF0 and Rules model and need to be generated a number of times equal to the number of time we want to activate an specific function. These external triggers represent the external influences that help govern the system. The triggers can serve as a fail safe to avoid the case of infinite

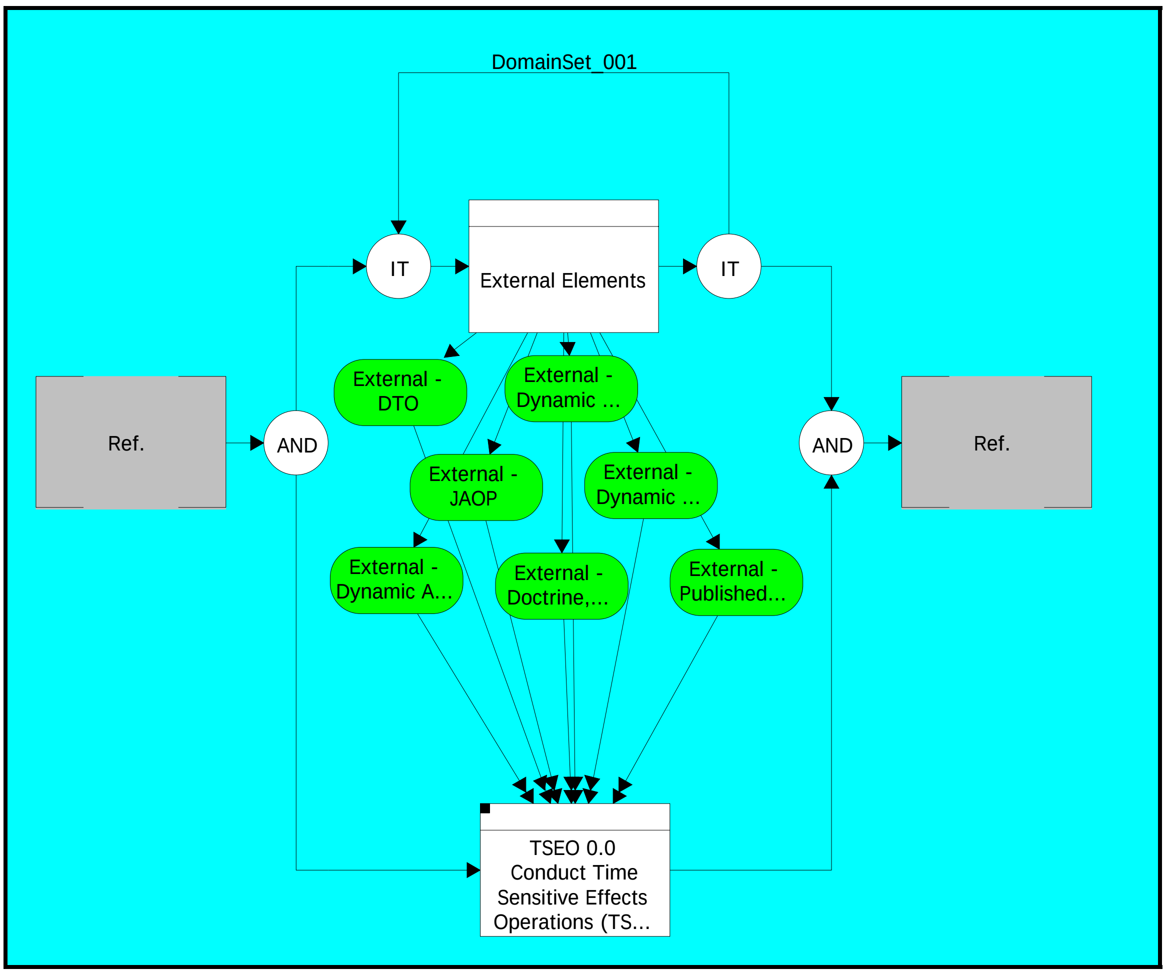

Figure 66: FFBD of Universe function

execution. These triggers are generated through an iteration construct and the iteration domain was set to an arbitrary number large enough not to inferior with the execution of the model. After the triggers are generated the simulation starts on function TSEO 1.0

and continues through to function TSEO 5.0. As these function are in series, no special manipulation is necessary. Once the TSEO 6.0 function is executed a query is made to determine if the current thread is a re-strike operation. If the answer is NO the function will continue to function TSEO 7.0; otherwise the function will continue to the TSEO 8.0 function. Until this point both executables have a similar construct, see Figure 67 and Figure 68. Both executables also flow through TSEO 9.0, TSEO 10.0 and TSEO 11.0.

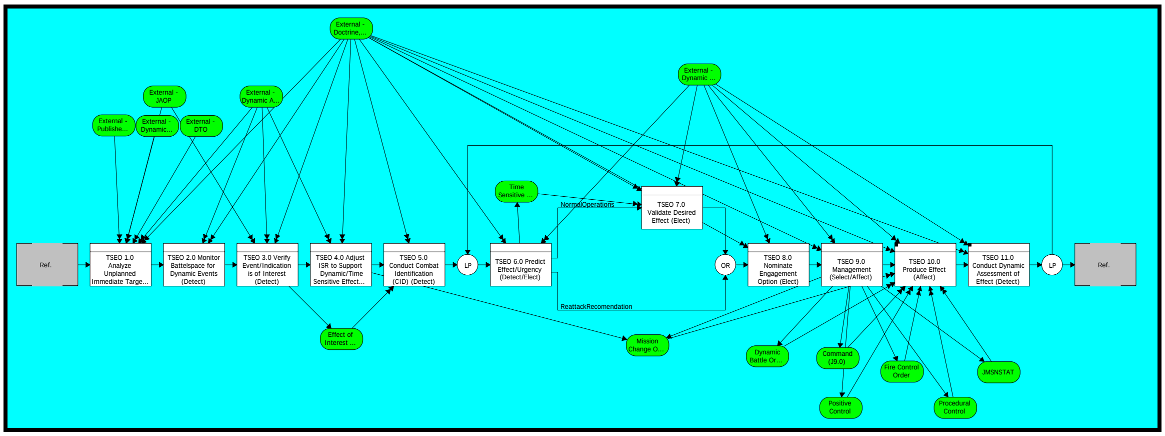

Figure 67: EFFBD of the original system

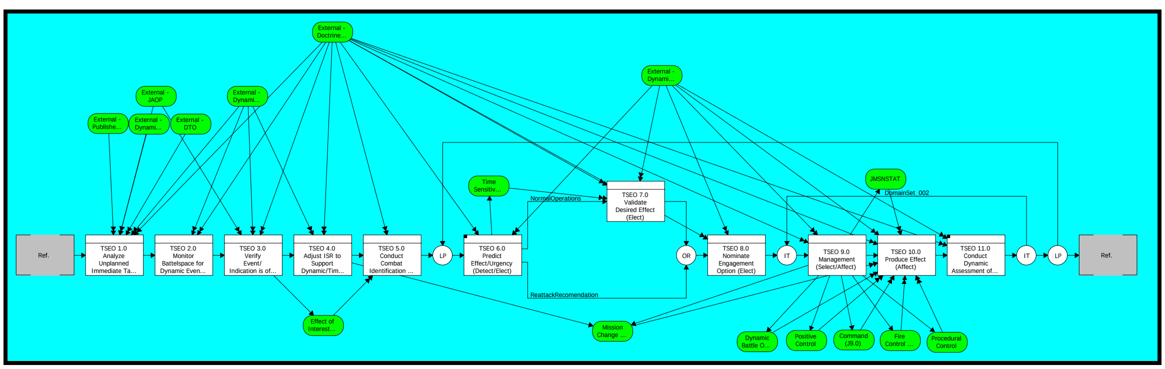

Figure 68: EFFBD of the modified system

On TSEO 10.0 we determine the damage that has occured to the target and the amount of regeneration (zero for first attack) undergone by the target. After damage is applied function TSEO 11.0 will determine if enough effects were seen. For a confirmed kill, both simulations will exit to reference, ending the execution. This is the point where the threads of the executables diverge. For an unsuccessful kill the original model will be dominated by the loop construct, returning to the TSEO 6.0 function. This time the exit criteria will identify the thread as a re-attack action and the bypass will be executed. The function flow will continue from TSEO 8.0 through TSEO 11.0 where another exit query is performed. The loop will continue until the target is destroyed or the global triggers are exhausted. In contrast the modified model will execute the iteration construct returning to TSEO 9.0 and continue through function TSEO 11.0 where the exit query is performed. The iteration is performed until the target is destroyed or the iteration domain is satisfied (five iterations). If the iteration domain is satisfied before the target is killed the loop construct will be activated. By activating the loop construct the execution thread will revert to the TSEO 6.0 function, effectively resetting the previous chain of events. The loop construct is always active, so as long that there is no kill confirmation and the iteration domain is satisfied the loop will execute.

Baseline vs. WBBDA

As previously mentioned we decided on a metric based on the number of sorties necessary to destroy a target. In Figure 69 we can see the cumulative probability distribution (sorties to kill target) for our two systems with the varying warhead lethality value perturbation. We will select a 75%-tile; what number of sorties will provide a kill 75% of the time, to help compare the data obtained. Two things can be noticed about the behavior of the two systems. First, there was an overall improvement in the number of sorties needed to destroy a target with the migration to the modified system. Second, for high lethality warheads we get minimal improvement when migrating from the original system to the modified one; while the average lethality warhead benefits more from

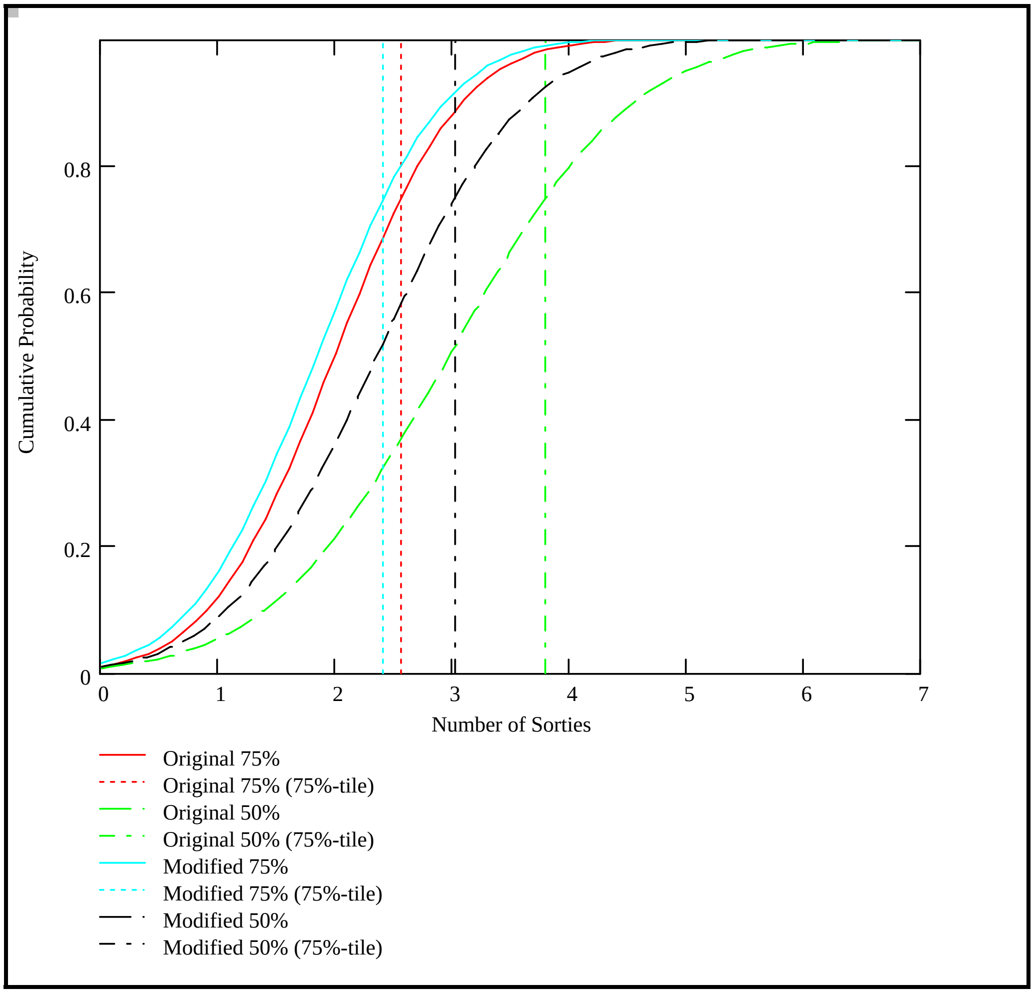

Figure 69: Cumulative probability

WBBDA. If we take a look at the number of sortie distributions, Figure 70 to Figure 73, we can see that the high lethality warhead reduced its variance with the modified system. In general the high lethality warhead did not benefit significantly when modified for WBBDA. For an average lethality warhead there is significant variance present in both the systems. The mode for all four Monte Carlo simulations was two sorties per kill.

With the 75% lethality modified system having an average of nineteen occurrences higher when compared to the other three data points. It is of note to mention the irregularity of the normal fit based on the data from the baseline average lethality warhead. This irregularity can be attributed to a number of outlier data (7 sorties). This phenomenon was not present on the high lethality warhead and the modified system with the average lethality warhead.

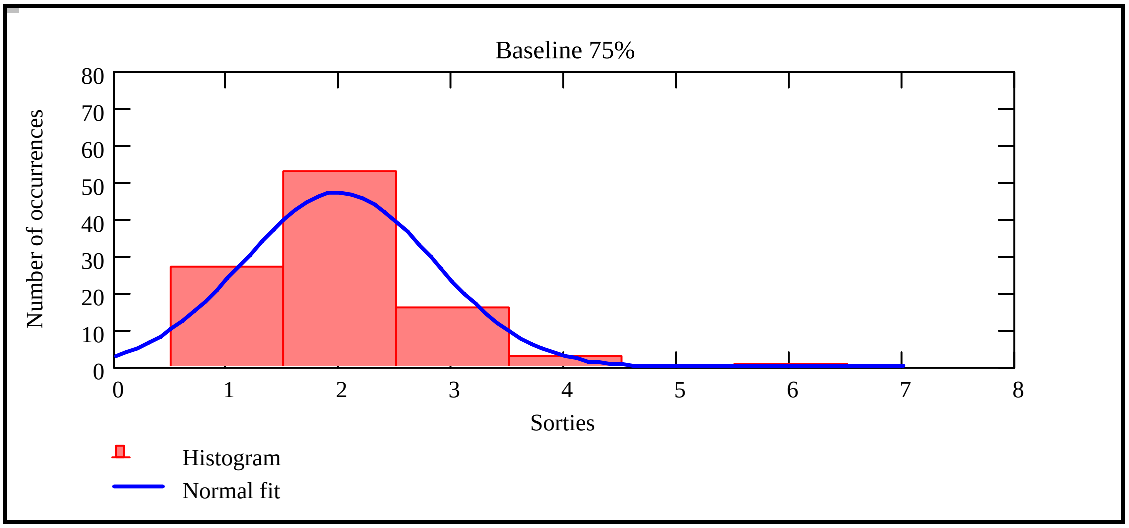

Figure 70: Baseline 75% lethality

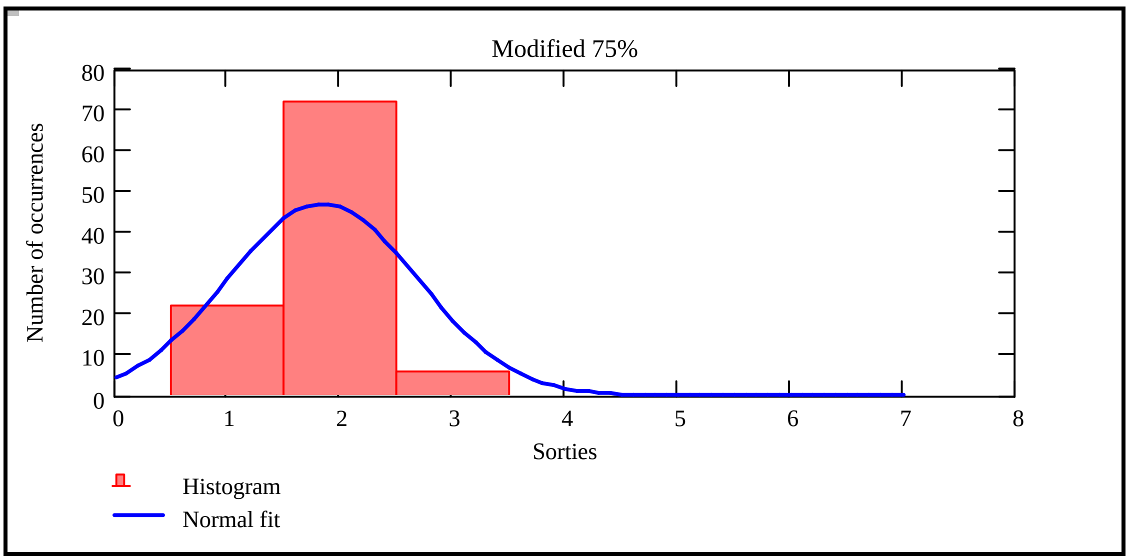

Figure 71: Modified 75% lethality

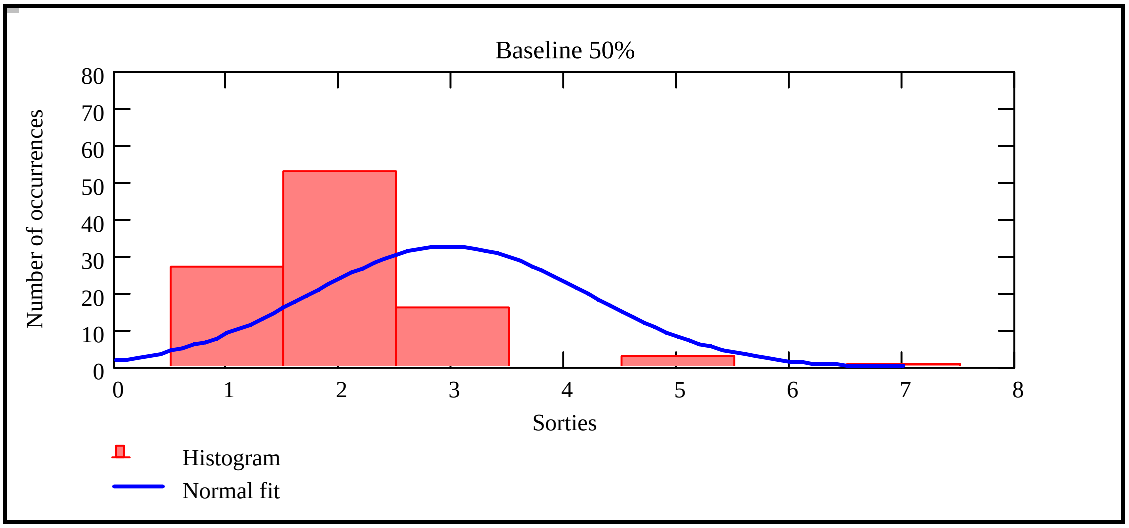

Figure 72: Baseline 50% lethality

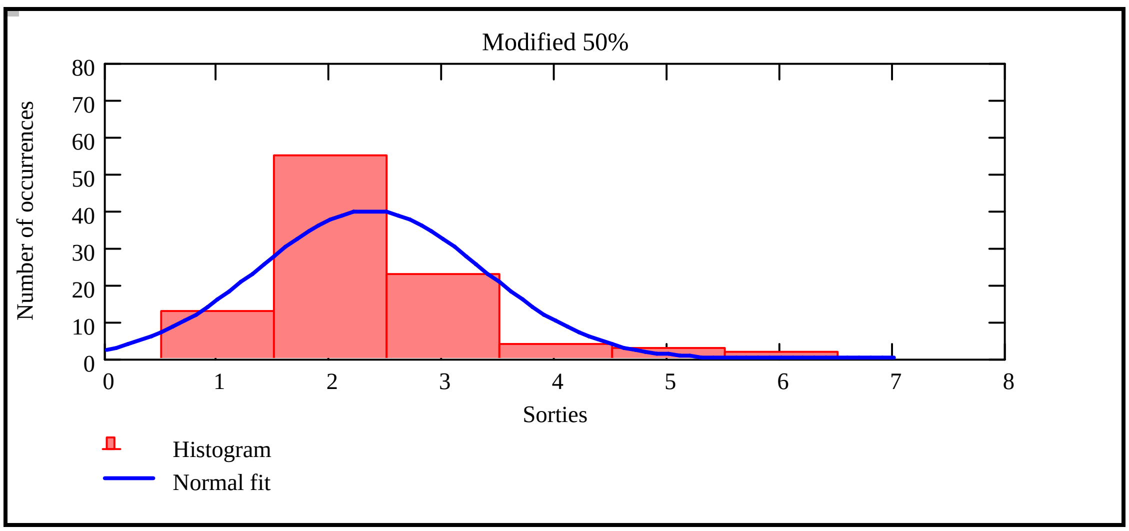

Figure 73: Modified 50% lethality

Architectural Analysis with Executable Models

The analysis compared the sorties per kill distribution of the Monte Carlo runs. It was assumed that the WBBDA had a 100% accurate detection capability and functions were given a normal time delay distribution that didn't take into account any breakdown in communications.

The executable model was stable within the constraints of our analysis. Changes to the model could be performed with the executable GUI open, making for fast turn around and debugging. There is room for improvement as more representative time intervals could be obtained for each function.

The level of detail of the executable model is an important decision. The complexity of the executable model increases dramatically as we decide to model increased numbers of sublevels of the system. There are also practical reasons not to model to the lowest level available. The fact is that an incomplete or an erroneous rules model will limit the amount of detail that can be introduced into the executable model. You can not accurately model what you are not aware of or don't know. It was found desirable to understand one level below from the lowest level implemented into the executable model. The extra level may not be complete; it just needs to be enough information to ensure an understanding of the level above it.

The top level executable provided a good platform to compare the effects from changes in the system structure. For this study we can only compare trends generated by the different executables generated. Even at this level of detail the executable is a valid tool when performing preliminary investigations of top level trade studies. This statement relies in the ability to generate or obtain a simple and valid rules model for the system. As a preliminary top level trade study you are not that concerned with specific detail data, you are more concerned with capturing a representative behavior of the system. This representative behavior can be compared to the behavior of other permutations, obtaining insight into which alternative deserves future investigation. More accurate distributions are needed to obtain realistic behavior from the executable. The extra burden required to obtain the better simulation may not be worth the effort at the early concept analysis stage of the lifecycle.

Summary

By observing the cumulative probability distribution we can infer that high accuracy/lethality warheads combined with WBBDA will have minimal improvement over just better munitions. This can be attributed to the high damage these warheads can inflict to the stationary target on the first strike. If we look at the variance we see a significant improvement as depicted by a 94.47% percent difference in their variance, implying a more repeatable process for WBBDA-supported strike. We could infer because of their mean, 1.99 strikes for the original and 1.84 strikes for the modified, a strike on this type of structure should involve two warheads. If we look at the average lethality warhead case, we see a significant improvement in modifying the system with WBBDA. The percent difference of the mean and variance, was 23.31% and 40.39% respectively (see Table 16). If we look at Figure 69, we see a tendency of the modified

Table 16: Comparison of results

	High lethality warhead		Average Lethality Warhead	
	Mean (Sorties)	Variance	Mean (Sorties)	Variance
Baseline System	1.99	0.717	2.97	1.504
Modified System	1.84	0.257	2.35	0.997
Percent Difference	7.83%	50.21%	23.31%	20.47%

system with WBBDA and average lethality warheads resembles the behavior of a system using high accuracy warheads without WBBDA. The observed effect could be attributed to the lower damage accumulation of the average warhead.

VI. Conclusions and Recommendations

Conclusions

1) It is possible and useful to directly model an architecture, and use this model to perform behavior and performance analysis for the selected concepts.

    a) The architecture must be developed to support this type of analysis, and then it is critical that the key products be complete, correct, and consistent. At a minimum the OV-1, OV-5, OV-6, and SV-7 are necessary to obtain representative behavior from the architecture. Additional products may be needed to conduct performance analysis, as the current investigation was limited to behavioral analysis.

    b) When developing an executable model, complexity increases dramatically as you incorporate lower levels of decomposition. The model should be developed to the point where a good representation of the architectural behavior can be obtained. It is the author's impression that modeling to a level 2 decomposition would have been too much work for little return. No matter the level selected for modeling, it is beneficial to have available a decomposition at one level below that of the model. The extra level can be used to clarify any ambiguities that may be present in higher levels.

    c) Architectures can be used effectively to analyze and evaluate new weapon system concepts and modification to existing concepts.

    d) The CORE^{TM} environment has promise as an architectural analysis tool. It is one of very few software tools that directly model architectures, thus facilitating verification, validation and analysis. The introduction of a more comprehensive stochastic capability is essential for the development of more complex executable models.

2) WBBDA has promise as a future weapon system concept. This analysis is preliminary and insufficient; however, some conclusions can be drawn.
   a) The low lethality warhead benefited more from the introduction of the WBBDA in the architecture. At it has a lower rate of damage accumulation, it is more sensitive to the time elapsed between attacks. Further, WBBDA may allow for greater sortie efficiency by reducing the average number of warheads needed per target.
   b) To obtain the maximum benefit, communication needs to be reliable and timely and the ability to accurately detect WBBDA information in a timely fashion may be the limiting factor. The current investigation did not take into account any of these issues and the capability of WBBDA will be degraded due to them.

Recommendations for future research

1) To obtain a more accurate representation of system behavior and better understand the impact of WBBDA on such behavior, it is necessary to make some changes to the assumptions and approach for development of the executable model.
   a) Simulate more targets per sortie. The engagement of multiple targets per flight sortie is a more representative wartime scenario.
   b) Allow multiple warheads per target. As no conventional warhead is 100% lethal, it is common practice to deliver multiple warheads to a target. By allowing

multiple warheads per target it is now possible to investigate how WBBDA affects the efficiency (target kills per sortie) of different aircraft payload configurations.

c) Consistent with the discussion, a more meaningful metric would be targets kill per aircraft sortie. This metric can better quantify the effect that WBBDA has on the efficient use of battlefield resources.

d) Implement more representative time intervals. Time interactions between functions can heavily influence the behavior of the architecture, and this should be addressed in future investigation. It is recommended that the different function time intervals be prescribed by stochastic modeling.

Bibliography

[DAUSEF01] Department of Defense Systems Management College. "Systems Engineering Fundamentals," Fort Belvoir, VA: Defense Acquisition University Press, January, 2001.

[DITIC02] DITIC Joint Vision. http://www.dtic.mil/jointvision/ Accessed 2002.

[DoDAF02] Download DoDAF Documents. http://aitc.aitcnet.org/dodfw/ Accessed 2002.

[DoDAFD03] Department of Defense Systems Management College. "Department of Defense Architectural Framework ver.1 Deskbook." 2003

[DoDAFI03] Department of Defense Systems Management College. "Department of Defense Architectural Framework ver.1 Volume 1." 2003.

[DoDAFII03] Department of Defense Systems Management College. "Department of Defense Architectural Framework ver.1 Volume 2." 2003

[D&S04] C. E. Dickerson, S. M. Soules; M. R. Sabins; and P. H. Charles. "Using Architectures for Research Development and Acquisition," October, 2004.

[ESC03] ESC Chief Architect's Offices. https://cao.hanscom.af.mil/ Accessed Dec 2003.

[HQAF02] HQ Air Force Doctrine Center. https://www.doctrine.af.mil/ Accessed Dec 2002.

[Levis03] Alexander H. Levis. "SENG 640 Systems Architecture." Class Notes, AFIT, WPAFB, Winter 2003.

[Levis04] Alexander H. Levis. "Modeling and Simulation for Architecture Assessment,"

[SMP06] Strategic Master Plan FY06 and Beyond. http://www.cdi.org/news/space-security/afspc-strategic-master-plan-06-beyond.pdf Accessed 2002.

[Vitech02] Vitech Corp. http://www.vitechcorp.com/ Accessed 2002.

[Vittori03] Jay Vittori. "Time Sensitive Effects Operation (TSEO): How TCT will evolve by 2012."

[ZinnA04] Capt Andrew Zinn. "The use of Integrated Architectures to Support Agent Based Simulation: An initial Investigation." INCOSE paper, 2004.

[ZinnB04] Capt Adrew Zinn. "The use of Integrated Architectures to support agent based simulation an initial investigation." AFIT Thesis AFIT/GSE/ENY/04-M01

Vita

Mr. Luis M. Díaz Rodríguez graduated from the University of Puerto Rico, Mayagüez Campus, in 1999 with a B.S. in Mechanical Engineering. While at the UPR Mr. Díaz Rodríguez participated in the Formula SAE competition and participated in a CO-OP program at Pratt & Whitney, CT. Upon graduation he started working at the Aeronautical Systems Center at Wright Patterson Air Force Base (WPAFB) as a systems engineer for the Flight Systems Engineering Division Propulsion Branch (ASC/ENFP). While at ASC, he provided flight structures support to the Propulsion System Program Office (LP). After the dissolution of the propulsion branch in June 2002, he joined the Flight Systems Engineering Division Structures Branch (ASC/ENFS) where he continued supporting propulsion structures development for different aircrafts. In January 2001, he entered the Graduate School of Engineering and Management, Air Force Institute of Technology as a part-time student. Later on he applied and was accepted for Long Time Fulltime Training (LTFT) at AFIT. Upon graduation, he will be assigned to the Mobility System Wing C-17 Group (ASC/YCEF) at WPAFB.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. REPORT DATE *(DD-MM-YYYY)*
29-04-2005

2. REPORT TYPE
Master's Thesis

3. DATES COVERED *(From – To)*
January 2001 – April 2005

4. TITLE AND SUBTITLE
EXECUTABLE MODEL DEVELOPMENT FROM ARCHITECTURAL DESCRIPTION WITH APPLICATION TO THE TIME SENSITIVE TARGET PROBLEM

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Díaz Rodríguez, Luis M., Aerospace Engineer, USAF

5d. PROJECT NUMBER
If funded, enter ENR #

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)
Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way, Building 640
WPAFB OH 45433-8865

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GSE/ENY/05-M04

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFIT/SY

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
As the Department of Defense (DoD) moves to a capability based approach for requirements definition and system development, it has become necessary to conceptualize and evaluate our needs at the System of System (SoS) level. Desired capabilities are often achievable only through seamless integration of many different systems. As the classical system engineering approaches are not suited to effectively handle the complexity of SoS level concepts, an architectures-driven approach has emerged as a way of defining and evaluating these new concepts. While the use of architectures for documenting and tracking interfaces and interoperability concerns is generally understood, architectural analysis and the use of executable models for evaluation of architectures remain an open area of research. With this purpose in mind, this thesis will apply architectural-based analysis to the proposed Time Sensitive Effect Operation (TSEO2012) scenario. This scenario will become the baseline for architectural analysis, and an excursion to this baseline will add a Weapon Born Battle Damage Assessment (WBBDA) capability. By creating an executable model, the two architectural concepts can be compared against each other. The addition of a WBBDA capability to the TSEO architecture improves the efficiency of the time sensitive target operations by shortening the decision cycle for target re-strike. While this effort was successful in obtaining an executable model directly from the architecture description, it highlighted the importance of having sufficient and correct information contained in the architecture products.

15. SUBJECT TERMS
Architectural analysis, Architectural products, Executable models, System engineering analysis

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
U	U	U	UU	109	David R Jacques, PhD, USAF
					19b. TELEPHONE NUMBER *(Include area code)* (937) 255-3355, ext 3329 (david.jacques@afit.edu)