IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 10, 2018, accepted June 25, 2018, date of publication July 4, 2018, date of current version July 25, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2852803

An Interactive Model-Driven Simulation Approach for Dynamic Behavior Analysis in Armed Conflicts

MINGHAO LI, JIANGUO XU, KEWEI YANG, AND BINGFENG GE, (Member, IEEE)
College of Systems Engineering, National University of Defense Technology, Changsha 410073, China

Corresponding author: Minghao Li (liminghao_nudt@foxmail.com)

This work was supported by the National Natural Science Foundation of China under Grants 71690233, 71501182, and 71571185.

ABSTRACT As system dynamic behaviors are difficult to capture and analyze in model-based systems engineering, executable architectures, and simulation methods are widely studied to understand the behaviors that result from interactions amongst system components. However, most current studies often strive for analyzing independently specific aspects of architecture rather than capturing simultaneously the structural, behavioral, and performance related features, leading to failures of understanding the whole system from a global perspective. Accordingly, an interactive model-driven simulation that includes meta-model mapping and collaborative engine capabilities specific to DoDAF, Rhapsody, and STK simulations is first proposed to integrate the full advantages of different analytical tools. Then, a synergic engine is developed to provide synchronized control between time-driven and event-driven execution through the dynamic analysis process. Two executable techniques communicate with each other throughout the entire execution cycle to provide complementary necessary process information. The events occurred in the STK simulation scenario are used as the trigger of the executable state machine model that describes only the logical procedure of system behaviors without time- and space-related constraints. Last, an illustrative example of the Russian–Turkish Plane Incident is carried out to demonstrate the feasibility of the foregoing approach.

INDEX TERMS Model-based systems engineering, multi-modeling, dynamic behavior analysis, systems simulation, interactive validation.

I. INTRODUCTION

With the development of information technologies and the ubiquity of interdependence and interaction amongst systems, most real-world systems continue to grow in scale, complexity, and uncertainty. Consequently, systems engineering has become a methodology that focuses on how to design and manage complex systems and procedures over their lifecycle to solve these challenges with widespread applications in defense industry, business management, and elsewhere. As a major cause of the complexity, dynamic interactions amongst systems make the behaviors and properties of each system associated together, which also increase the difficulties to understand global dynamic process of complex systems and lead to a series of functional logic verification issues. With the evolution of system engineering to a new fashion, research on dynamic behavior analysis in the context of model-based systems engineering (MBSE) [1], [2] have become more and more popular, many executable techniques [3]–[7] emerged in recent years for different analysis needs throughout the systems decision process. They execute models over time and space to understand the behaviors that result from interactions between system components, such as discrete event simulation for assessing operational factors of the system, and physics-based simulations that model the physics of system components, such as communications in electronics components or probability of detection for radar systems. However, as more factors are considered, traditional executable techniques cannot solve the complex behavioral logic problems independently, which calls for a collaborative multi-modeling approach to combine the strengths of these techniques together and make these analytical tools interoperate to achieve new analytical capabilities.

This paper focuses on the problem that complex system logic cannot be comprehensively analyzed in MBSE filed. In order to achieve unified understanding among stakeholders with domain-specific concepts, architecture frameworks are adopted to describe the structural and behavioral information with various models from different perspectives and organize the models into an effective structure [6]. Although SysML has been widely used in MBSE as its executable analysis ability to verify behaviors of systems, limitations still exist. On one hand, is for the domain specific concepts in military field, it show weak descriptive ability relative to DoDAF. On the other hand, it lies in the lack of consideration for reality constrains. In industrial and military field, many hardware and systems are included into the architecture, the performance indexes of each system and their interactions with operational environment have a great impact on the systems behaviors that SysML models cannot reflect. In order to analyze more details about system behaviors, professional executable platforms are obviously important part in this study, which calls for a model transformation research. There have already been many approaches working on solving multiple model transformation problems [9]–[11], whereas most of the transformation is manual process which lacks of consistency and accuracy, and the result is usually analyzed by visual inspection.

For dynamic behavior analysis purpose, there are various executable techniques such as SysML executable model, discrete event model, petri-net model, and simulation model. All these different executable analysis methods can describe and analyze narrow aspects of dynamic behaviors. Some of them focus on verifying temporal attributes or performance, while some of others focus on analyzing the rationality of the functional logic. Most of these tools work well in their own specific filed independently. However, there are few methods to combine their respective strengths together and it is difficult to implement synchronous execution and share internal details with each other to make a more comprehensive decision. For instance, SysML executable model [12] is designed to perform functional logic analysis focusing on how the components interacted in a specific context without considering the geographic information and specific parameters of some components, such as flight speed, reconnaissance range, landforms, etc. However, the simulation method [13] focuses on how the components respond to external environments when executing specific tasks. If a simulation runs out, we can only see the results and data produced, however, we do not know whether the behavior process is correctly executed relative to the intuitive graphical behavior model.

Considering the above issues, the motivation of this paper is to graphically demonstrate and analyze the complete behavioral process of a multi-system operational scenario through a collaborative multi-modeling approach automatically and precisely. The solution we choose is to design an interactive model-driven simulation framework which incorporates two executable techniques with physics based performance and circumstance parameters to address the concerns. However, engineering obstacles exists. One is that different techniques means different formalisms, we need to propose a method to ensure the consistency of models from a semantic perspective. Second, the integration of SysML and simulation means that we doesn't simply use these tools together, but execute them in parallel with synchronization and communication. As STK simulation is time-driven and SysML is event-driven, theoretical and practical barriers exist to make them work as complement to each other. We need to design programs to realize software integration and execution control. Because this field is mostly engineering field, so the innovations are also about engineering realization.

The innovative contribution of this paper is as follows. First, we propose a methodology framework to graphically conduct dynamic behavior analysis by an executable SysML model driven simulation with DoDAF specifications. Second, in order to ensure the consistency of semantics, a meta-model approach is studied to support model transformation. It is also a new kind of attempt. From this perspective, models can be transformed and generated by analyzing the concepts of core data elements and meaningful relationships among them instead of accommodating the graphical model information [14]–[16]. Finally, because STK holds powerful simulation capabilities for real physical environment and system operations, and is time-driven which is different from other simulation techniques, a new monitor-response service program is designed and developed to break down the barriers between time-driven and event-driven execution. Instead of simply sharing inputs and outputs, they communicate with each other throughout the entire execution cycle to obtain the necessary process information. Although the details about program implementation are not given in the text, it is still a major innovation in engineering.

The Russian-Turkish Plane Incident happened at the end of 2015 is a typical system dynamic process. It involved several weapon systems such as radars, missiles and planes to anticipate interactively in the conflict. The result is that Russian aircraft was shot down, but for the true process, both Russian and Turkish stick to their own arguments. Thus, this incident is chosen as an illustrated example to verify the applicability of the proposed approach and determine which version of the statements about this incident is reasonable based on the information published online.

The remainder of this paper is organized as follows. In section II, the DoDAF-based approaches, model-driven simulation frameworks and dynamic analysis techniques are briefly introduced. In section III, the model-based dynamic behavior analysis approach is proposed. A case study of Russian-Turkish Plane Incident is conducted to verify the rationality of two scenarios in section IV.

II. RELATED WORK

A. DODAF-BASED ANALYSIS

The DoDAF 2.0 is designed to be more flexible to provide the consistency of models [11]. "Data-centric" concept is first proposed in this version to replace the traditional "Product-centric" mode. The whole architecture framework works as a data template to collect the data needed to be considered in decision support system. Each artifact is allocated to collect only part of the whole dataset with fundamental data concrete, related associations, and attributes. All the information collected can be shared across the architecture and further support system engineering analysis needs.

Each viewpoint and artifact is defined and developed for a specific analysis and management purpose in defense acquisition to help manage the risks of complex procurements. One of the applications in the acquisition is portfolio management which includes capability gap analysis, capability phasing, and impact analysis for options and disposal. Many approaches focusing on making use of the architecture data to fulfill the analysis needs in their own fields have been proposed in recent years, such as operation loop analysis [18], mission reliability assessment, systems portfolio, resource planning, and risk analysis. Different analysis methods like Markov Chains theory, Petri Net theory [6], complex network theory [19] and fuzzy theory were employed with professional domain tools and models to evaluate the system performance. Griendling *et al.* [20] used SV-1 combined with SV-2 to generate a network model to evaluate centrality in a weapon structure evaluation approach. Muller and Dugli [21] employed DoDAF models as the input and modeling language to explore the coevolution in a counter-trafficking system of systems. However, most of the methods mentioned above focused just on the usage of the content described in specific viewpoints without considering how the architecture data could be transformed correctly.

B. MULTI-MODELING INTEGRATION

With the increase in demand for solving complex new problems, more and more researchers begin to concentrate on multi-modeling approaches. Multi-modeling involves several techniques which are usually domain-specific and each of which offers unique insights and assumptions about the domain model. Two or more models run concurrently and supply a complementary part of a solution, and one model supplements computational or analysis tasks of another with results and parameter values. There have existed a lot of environments that support multi-modeling based modeling and simulation, such as the Command & Control Wind Tunnel (C2WT) [23] developed by Vanderbilt University, the Service Oriented Architecture for Socio-Cultural Systems (SORASCS) [24] developed by Carnegie Mellon University and NAOMI [25] developed by University of California Berkeley, including model data exchange, constraint management, change propagation, and model execution. These environments address the multi-modeling issues by defining a flexible framework and integration model to capture the essential flows of information and provide support for flexible orchestration, coordination, and transformation.

ModelCenter is a commercial integrated modeling and analysis platform of federating models and data from diverse ecosystems of modeling and simulation tools, enterprise applications, and data repositories, and weaving a digital connected graph. It integrates multiple domain specific software by formalizing the data interfaces of components. Each software shares inputs and outputs with each other so that the workflow can be executed subsequently. In this framework, software like CAD systems, Simulation tools (e.g. Mathematica and MATLAB/Simulink), SysML modeling tools (e.g. MagicDraw, Rhapsody), and other professional tools are integrated to address complex analysis concerns. Further, with the support of MBSEPak, STK can be integrated into the SysML model as a component to exchange parameters or results with other parts of a defined system in MagicDraw and Rhapsody. In addition, other kinds of multi-modeling based simulation techniques exist, such as Agent Modeling of Event-Based Architectures (AMoEBA), Syndeia from InterCAX which establish mappings between simulation elements, and a robust discrete-event simulation interpreter and execution environment of MagicDraw with the Cameo Simulation Toolkit. For the multi-modeling based simulation applications, Jbara *et al.* [26] proposed a methodology involving Social Networks, Time Influence Nets, Organization Structures, and Geospatial models to solve a class of problems in Drug Interdiction and Intelligence domain to identify best courses of action (COAs).

C. DYNAMIC BEHAVIOR ANALYSIS

Specific executable dynamic platforms are combined with architecture frameworks as executable architecture (EA). An important characteristic of EA is the ability to execute the model directly from architecture products with minimal additional system definition or manipulation. Various simulation techniques serving for the development of the executable architecture with DoDAF have been established by academia and commercial software companies. Discrete Event Simulation (DES) is a modeling concept executing events or processes in a discrete period of time [6]. It is usually utilized to model discrete system changes with statistical significances. Color Petri Net (CPN) extended from Petri Net with an expanded token is also a unique executable analysis method to capture behavior features in a network-based theory. There are also some other techniques such as multi-agent simulation system, Markov Chain, and Mathematical Graphs. However, most of them only support single directional transformation from static architecture products, once the error is discovered during the simulation process, it needs to be corrected in the initial architecture model and translated again. Moreover, in the transformation process, the graphic model has to be first translated to a dataset that can be recognized by the simulation platform and then generates the platform related model. Two phases of the data translation will cause additional errors. To consider this, Ge *et al.* [7] proposed a translation method from a high-level Data Meta Model perspective. The architecture data meta-model and executable formalism meta-model were first defined to guide modeling respectively and then related by mapping rules. The proposed method has been utilized to translate DoDAF described model to CPN [7] and DES [11] executable model successfully in his research. Accordingly, what the static model looks like and which language is used to represent architecture information are no longer important.

III. THE PROPOSED MODEL-BASED DYNAMIC BEHAVIOR ANALYSIS FRAMEWORK

DoDAF supports the application of MBSE in the military field and provides the description framework for acquisition projects based on the DoD tasks and principles and further the schema for the data repository and exchange. A complete MBSE framework usually consists of methodologies, modeling platforms, languages, and purposes oriented analysis functions. The methodology proposed in this paper employs the black box logic modeling process supported by IBM Rational Rhapsody with a SysML modeling language as a component to analyze the system behavior from the logical perspective. Although SysML has shown feasibility in system modeling and Rhapsody has been proven to be well suited for SysML executing, the SysML language does not natively include military-related semantics. Thus, SysML is not suitable for the military architecture modeling. In this paper, DoDAF is considered as the original architecture standard of interest because of its user-friendly interface to military stakeholders. Then, the DoDAF models will be translated to executable models. In order to overcome the limitations of the analytical functions provided by Rhapsody, STK is adopted as a supplement, so that any event happening in the simulation process can be extracted as the triggers of the execution process of the SysML model. Considering that STK cannot graphically describe the logical process of the behavior, and the executable SysML cannot execute the components' spatiotemporal and positional properties. These two platforms are integrated together as a whole executable analysis framework to complement each other.

The analysis process supported mainly by Rhapsody, STK and the Collaborative Engine (independently developed in our research) aims to check and find out the logical errors under a spatiotemporal scenario. A modeling workflow is created based on the analysis purpose to capture the interoperations between interconnected models of different platforms. The syntactic and semantic consistency will be guaranteed by a high-level meta-model mapping relation aligned with DM2. The model-based dynamic behavior analysis framework coping with DoDAF views and executable analysis techniques consist of three main phases.

More specifically, in the first phase, two viewpoints of DoDAF are first constructed to describe the interrelations among systems involved and behavior information of each system. In the second phase, two types of executable models are generated with additional information which is platform specific through meta-model level translation. In the last phase, the mapping relation between events in the simulation scenario and the trigger condition in SysML executable model will be defined and the dynamic behavior analysis process can be conducted. The expected output of this phase is a logical correctly sequence model describing the interactions among components through a period of time under a real spatiotemporal situation.

The model-based dynamic analysis framework is shown in Figure.1 and the analysis procedure is described in detail as follows.

A. PHASE 1: SCENARIO ANALYSIS WITH DODAF VIEWS

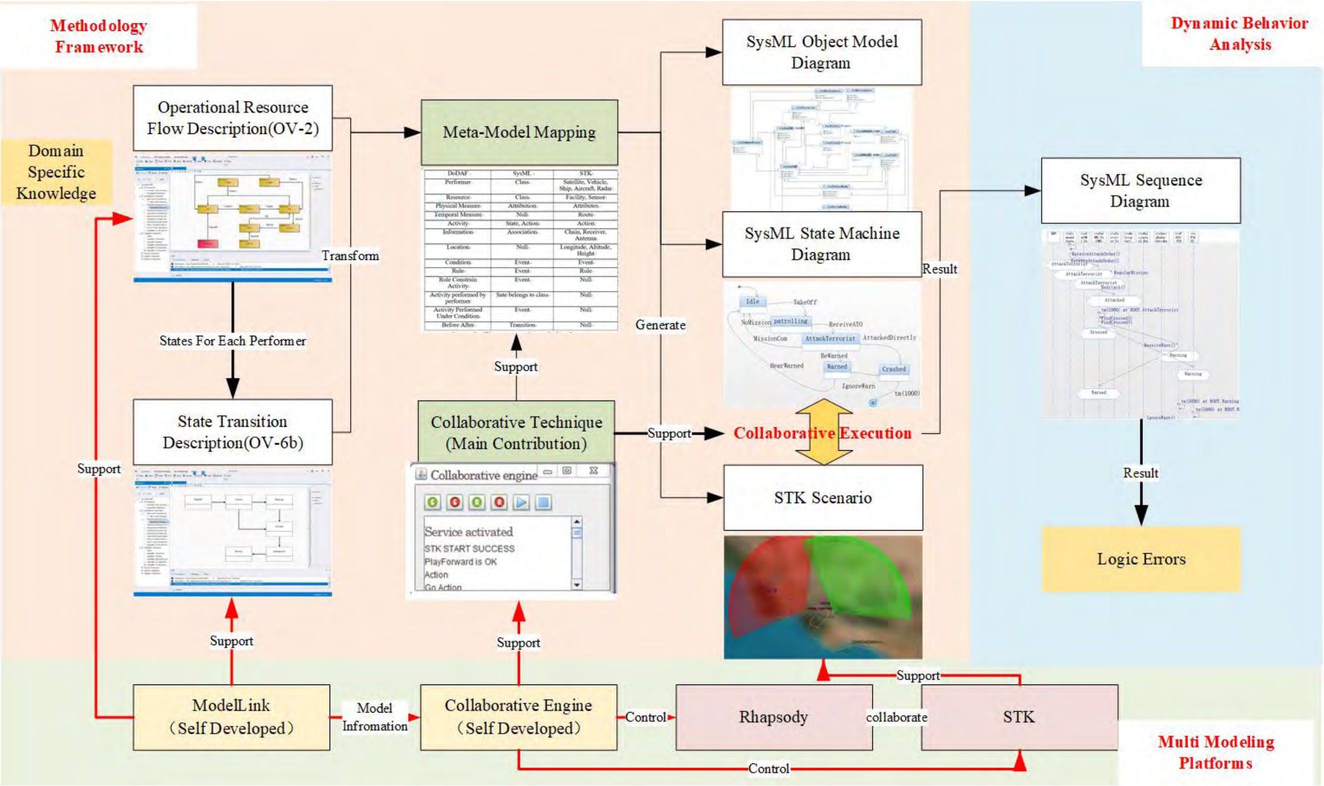
One of the main ideas of DoDAF is ''Fit-for-Purpose'', indicating that not all the artifacts defined have to be modeled. OV-2 and OV-6b were chosen in this scenario to describe the structural and behavioral aspects of the architecture. By analyzing a certain scenario, information about system configuration and system attributes can be collected as the input of DoDAF. In this study, the modeling phase is assisted by ModelLink developed by our research group owing to its flexibility in view customization and description, as well as the convenience in customizing meta-model' structure and properties.

1) STEP 1.1: OPERATIONAL RESOURCE FLOW DESCRIPTION

The OV-2 DoDAF-described Model is designed to describe the resource flows between anticipated systems within an operational context. Systems and resource flows are modeled as nodes and edges. Each system exchanges information, funding, personnel, or material necessary for mission completion with each other in a specific time step. This model plays mainly two roles: one is to provide constraints and rules for the state transition between systems, the other is to describe the detailed properties necessary for model translation to other platforms. In order to generate STK scenario automatically, extra properties like movement routes will be modeled as additional properties of the system.

2) Step 1.2: STATE TRANSITION DESCRIPTION

The OV-6b is a graphical method to describe how an operational activity responds to external and internal events by changing its state. The potential usage of OV-6b includes the analysis of business events, the identification of constraints, and the behavioral analysis. It is one of the products establishing dynamic issues defined in DoDAF 2.0. Each system accomplishes tasks with series of states and transitions. The internal or external actions can cause the transition of states. For all systems, states are always nested. The product provides a graphical way to analyze the completeness of the rule set, detections of dead ends and missing conditions. In order to describe the whole state transition process, several steps need to be executed. First, decompose the operational activity into a series of time-ordered steps. Second, for each step, analyze the states of each system that may exist and the operational events that may trigger the transition of states. Last, collect all the states belonging to a single system along with the whole operational activity and construct OV-6b products for each system. Note that the state transition between systems must conform to the system association rules in OV-2.



DoDAF	SysML	STK
Performer	Class	Satellite, Vehicle, Ship, Aircraft, Radar
Resource	Class	Facility, Sensor
Physical Measure	Attribution	Attributes
Temporal Measure	Null	Route
Activity	State, Action	Action
Information	Association	Chain, Receiver, Antenna
Location	Null	Longitude, Altitude, Height
Condition	Event	Event
Rule	Event	Rule
Rule Constrain Activity	Event	Null
Activity performed by performer	Sate belongs to class	Null
Activity Performed Under Condition	Event	Null
Before After	Transition	Null

```
Service activated
STK START SUCCESS
PlayForward is OK
Action
Go Action
```

FIGURE 1. Interactive model-driven simulation framework for dynamic behavior analysis.

B. PHASE 2: MODEL TRANSLATION

In order to conduct dynamic behavior analysis and capture the logical features of the operational activity completely, a multi-modeling technique is employed with two pieces of executable software.

Rhapsody provided by IBM is a visualization modeling platform for system engineers and software developers. It was originally designed for the purpose of more conveniently and systematically developing in software engineering. Programming code can be automatically generated by constructing structural and behavioral models with UML language. Nodes, attributes, and links in UML formation are translated into classes, relationships between classes, class properties, methods, and functions making it possible for a rapid prototyping and flexible to requirements changes. In accordance with the expansion of model-based concept in system engineering fields, SysML language, and Harmony SE methodology were integrated into this platform as a critical component with the function of the logical error detection and correction. Three diagrams are employed in our research, Object Model Diagram and State Machine Diagram will be generated from DoDAF model as the inputs of the dynamic behavior analysis process. The Sequence Diagram is the final result representing the logical issues of the input information, and it will be generated by the execution of the State Machine Diagram through transitions between states. There are three types of transition conditions: fixed time, transition of other states, and external events. Events like whether or when a missile can be detected by a radar under terrain restrictions cannot be given directly in Rhapsody. Therefore, other platforms need to be incorporated.

STK is a physics-based software package widely used in the aerospace and defense communities allowing engineers and scientists to perform the real-time visualization and complex analyses of ground, sea, air, and space assets, and share results in one integrated solution. STK provides the analysis ability of calculations for access, communications systems, radar, interplanetary missions and orbit collision avoidance. Within each scenario, any number of satellites, aircraft, targets, ships, communications systems or other objects can be created. All results that depict the key behavior interaction information can be adopted as the triggers for transition of states in Rhapsody. Thus, STK constitutes a key part of the whole behavior analysis process for the sequence diagram generating.

1) STEP 2.1: META-MODEL MAPPING

Based on the approach proposed by Ge *et.al* [11], a meta-model level mapping methodology is adopted as the guiding principle to conduct the model translation with semantic consistency. By constructing the mapping rules between meta-models of architectural data and executable formalism, the architecture instances can be transformed to a platform-specific data format by a series of operations automatically. In order to build a reasonable meta-model mapping dictionary, it is necessary to analyze the semantic concepts of the fundamental meta-model structure respectively and decide which meta-models are of the same meaning by considering what they are intended to express and how they are organized from a high-level perspective regardless of the graphical formalisms. Then a module to realize the transformation process is developed based on the interfaces provided by each software with a physical exchange specification. The related files include mainly meta-model schemas, project files, and mapping rule dictionaries.

As shown in Tables 1 and 2, mapping relations between core concepts and artifacts both in DoDAF and SysML are listed to provide a data-centric transformation vision in relative to product-centric habits. As the same concept may appear in different views, all the data instances can be converted to a consistent dataset without considering the contents of independent products. Additionally, DoDAF is more comprehensive for the description in military operations. It can provide the information which is necessary for SysML modeling and simulation modeling respectively.

TABLE 1. Mapping relation between DoDAF meta-model and products.

	OV-2	OV-6b
Performer	√	√
Resource		
Physical Measure	√	
Temporal Measure	√	
Activity		√
Information	√	√
Location	√	
Condition	√	√
Rule	√	
Rule Constrain Activity		√
Activity Performed by Performer		√
Activity Performed under Condition		√
Before After		√
Organization	√	

TABLE 2. Mapping relation between SysML meta-model and products.

	Object Model Diagram	State Machine Diagram
Class	√	√
Attribution	√	
Operation	√	
Association	√	
Generalization	√	
Dependency	√	
State		√
Transition		√
Event		√
Action		√
Sate Belongs to Class		√

TABLE 3. Meta-model mapping relation across three formalisms.

DoDAF	SysML	STK
Performer	Class	Satellite, Vehicle, Ship, Aircraft, Radar
Resource	Class	Facility, Sensor
Physical Measure	Attribution	Attributes
Temporal Measure	Null	Route
Activity	State, Action	Action
Information	Association	Chain, Receiver, Antenna
Location	Null	Longitude, Altitude, Height
Condition	Event	Event
Rule	Event	Rule
Rule Constrain Activity	Event	Null
Activity performed by performer	Sate belongs to class	Null
Activity Performed Under Condition	Event	Null
Before After	Transition	Null

Table 3 shows the meta-model mapping dictionary for these three formalisms. Performers in DoDAF have the same meaning with Class in SysML and System Components in STK. Physical Measures will be converted into Class Attributes, and STK system components' properties, such as radar detection range and aircraft speed. The resource flows will be transformed into Associations and Chains. For the core concepts defined in DoDAF, some of them are dedicated to SysML and some are for STK only. For example, location data is not required in SysML, whereas it can correspond to Latitude, Longitude, and Height in STK. Relationships between activities can be converted into transitions in SysML, whereas, the data is not necessary for STK.

2) STEP 2.2: EXECUTABLE SYSML MODEL GENERATION

Static structure and behavior information are two core elements in the description of a military activity and also the critical components for the executable dynamic logic analysis. Once the DoDAF model is constructed, essential elements will be collected as data repository in accordance with the meta-model template. Through the mapping rules and specialized codecs, the key information can directly generate the relevant models in Rhapsody. Herein, the Object Model Diagram will be generated with the data of performers, resources, and information flows collected mainly from OV-2, and the State Machine Diagram will be generated with the data of performers, activities and so on.

3) STEP 2.3: SIMULATION SCENARIO GENERATION

With the information provided by DoDAF model and additional attributes described in it, scenarios can be generated for the simulation and events extraction purpose. Performers with labels describing which kind of weapon system they belong to and temporal properties will be collected and transformed to instances of predefined modules in STK. So far, STK has supported a variety of modules including ships, satellites, aircrafts, missiles, vehicles, facilities and radars, each of which has its specific behavioral characteristics and attributes. From the DM2 perspective, resource flows between performers in an activity which is expressed as associations between systems in the graphic level will be mapped to senders and receivers in a scenario describing the behavior of communication or detection. The spatiotemporal and positional information like movement routes and module specific attributes like radar detection range attached to performers will also be used to supplement the module information to make the modeling complete and executable. Through these two steps, a basic dynamic behavior analysis framework is constructed.

C. PHASE 3: DYNAMIC BEHAVIOR ANALYSIS

The last but also the most important phase leverages a Collaborative Engine module to integrate two platforms to gain new analytical capabilities. For the traditional simulation method, STK is usually employed as an independent platform to provide visualization and analysis for a specific purpose. In addition to its powerful analytical capabilities, its openness and external controllability for developers are also features of concerns in this paper. STK can be embedded within another application (as an ActiveX component) or controlled from an external application (through TCP/IP or Component Object Model (COM)). The Collaborative Engine is developed based on this feature. It uses TCP port to monitor the simulation process and delivery the event message to Rhapsody. Rhapsody, STK and the Collaborative Engine in this part constitute the execution toolchain as shown in Figures 2 and 3.

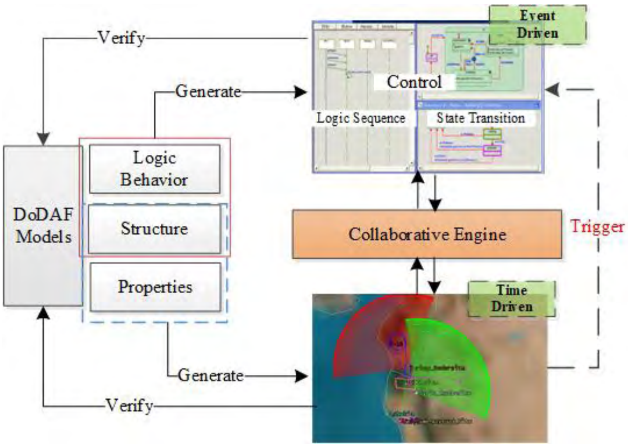
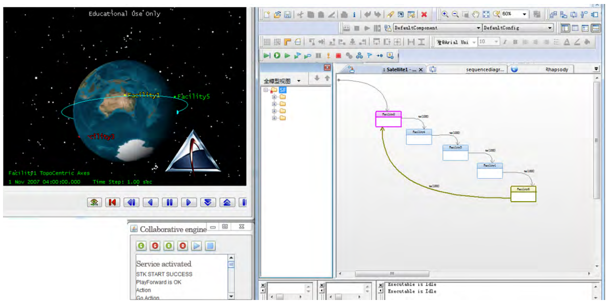


FIGURE 2. The relationship among the DoDAF models, executable SysML, and simulation.

1) STEP 3.1 EVENTS SYNCHRONIZATION

The Collaborative Engine consists of three parts including translator, detector, and communicator as shown in Figure 4. Based on the interfaces provided by Rhapsody and STK, these three modules provide the abilities of data translation between platforms and real-time communication. The translator is a component developed based on a meta-model



```
Service activated
STK START SUCCESS
PlayForward is OK
Action
Go Action
```

```
Executable is Idle
Executable is Idle
```

FIGURE 3. User interfaces for collaborative engine.

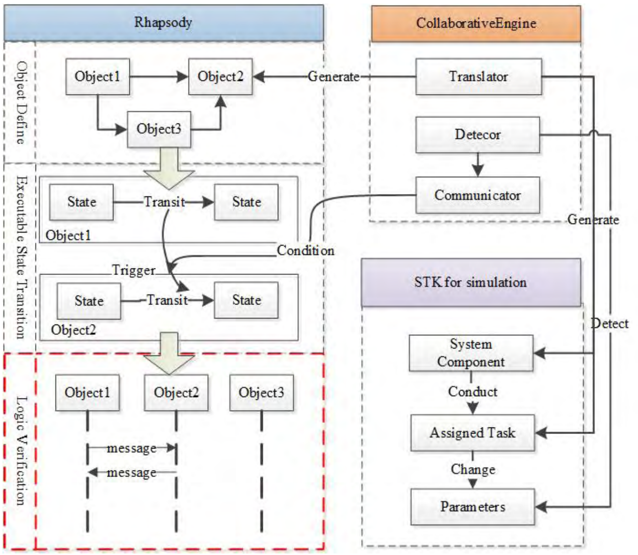


FIGURE 4. Internal work details of toolchain execution.

mapping theory for the model translation between formalisms. The detector is designed as a monitor to run along with the simulation process and identify whether some events concerned can happen or not. Since all the entities in STK perform assigned tasks independently, the software cannot tell which events will occur within a time period. Essential events need to be predefined like whether and when a vehicle moves to the specific location so that the detector knows which information should be identified. When a specific event happens, a message will be generated in communicator component and delivered to Rhapsody as the trigger of transition between states. In the meantime, a sequence diagram can be generated subsequently.

With this integrated executable toolchain, a specific scenario can be analyzed considering the influence of system parameters and environmental parameters on the execution results. Assume that city A is targeted by missile 1, and radar A is used to detect the coming missile. Once missile 1 is detected, a message will be sent to the command center and the interceptor will be activated. Considering missile trajectory influence and the radar detection range, whether city A will be destroyed or not needs validation. By changing parameters of scenario, different sequence diagrams can be generated through the executable toolchain to support decision-making activities.

2) STEP 3.2 EXECUTION

As shown in Figure 4, the Sequence Diagram is the final result establishing the interactive behaviors of systems in a period of time. It can be used to verify that the design is reasonable, the process is complete and the conclusion is in line with expectations. Models in Rhapsody provide a graphical way to describe the logical process intuitively and rigorously showing weaknesses in the STK simulation method. The Sequence Diagram can accurately describe the direction, time-consuming, content, and order of the information flow in the process of running. By collaboratively executing these two kinds of models, a dynamic sequence diagram under the influence of simulation process can be generated. Through the analysis of final sequence diagram, logic errors can be detected.

3) STEP 3.3 ADJUSTMENT AND COMPARISON

With more detailed information being added, agile changes can be achieved by modifying model parameters in DoDAF models. Several alternative solutions having structural, logical, and parametric differences can be quickly generated and compared.

IV. THE CASE STUDY OF RUSSIAN-TURKISH PLANE INCIDENT

Armed conflicts in recent years have represented a different facet of the irregular warfare. On November 24, 2015, a Russian Su-24M fighter was shot down by two Turkish Ari Force F-16 fighters near the Syria-Turkey border after an airstrike operation against ISIS targets in Syria. Both sides of governments have their own views about this incident.

(1) The Russian government believes that the attack operation is premeditated.

(2) The Turkish government claimed that the cause of the attack was a border transgression of the Russian fighter which lasts for 17 seconds, while the Russian side denied it.

(3) The Turkish government said that its F-16 had repeatedly warned the Russian jet ten times in five minutes before opening fire. However, the other side claimed that there had been no warnings at all.

The Russian-Turkish Plane Incident involves several weapon systems with interactive behaviors, and it shows features of uncertainty, systematism and suddenness which bring a great challenge for the international emergency management. This kind of armed conflict often causes an extensive concern of the international community. With some relative information and radar detection data about the incident being disclosed, some dynamic behavior analysis techniques are able to be conducted. Since the authorities need to respond rapidly to what happened and give a reasonable statement about the incident, an agile and adaptive modeling and validation framework show great significances.

A. SCENARIO DESCRIPTION

Based on the information published by both sides, two kinds of scenarios could be modeled. The major concern of this incident is whether Russian fighters have crossed the boundary and which kind of scenario is more logical and convincible. The two scenarios involve the same systems, and the difference lies in the specific behaviors and parameters of systems. By analyzing the constituent of this conflict incident, an object set of 12 entities involving Russian SU24 fighters, two Turkish F16 fighters, two Russian pilots, two MiG-8 helicopters, Hmeimin Air Force Base, Syrian terrorists, Syria radar and Turkey radar can be extracted as shown in Table 4. The behavior and performance information can be converted to attribute value of entities. Table 5 lists the attributes and corresponding formulas of the SU24 fighter. Some of these attributes are functions of time, and values will change over time to describe the dynamic behavior in terms of time and location according to radar data. The relationship between the fighter and the external environment (boundary line) and the behavioral interactions with other systems can be obtained through simulation.

TABLE 4. Objects list of the Russian-Turkish plane incident.

Name	Relative Objects	Description
SU24	Hmeimin Base, F16, pilot, terrorists, Turkey radar, Syria radar	Russian fighter in Syria
Hmeimin Base	SU24, MiG8	Russian Air Force Base
F16	SU24, Turkey radar, Syria radar	Turkey fighter
MiG8	Hmeimin Base, pilot, terrorists	Russian helicopter
Pilots	MiG8, terrorists	Pilots of SU24
Terrorists	SU24, MiG8, pilot	Terrorists in Syria
Turkey radar	SU24, F16	Detection radar
Syria radar	SU24, F16	Detection radar

TABLE 5. Attributes list of SU24.

Name	Initial value	Measurement	Formula
Speed (v)	810	km/h	-
Crew	2	person	Null
Flight radius	1200	km	Null
Missile capacity	4	AA-8 missile	Null
Height (h)	6000	m	H(t)
Longitude (lo)	29.384	Degree	Lo(t)
Latitude (la)	38.273	Degree	La(t)
Flight time (t)	0	s	Null
Flight mileage (s)	0	m	s=v*t

B. STRUCTURAL AND BEHAVIORAL MODELING

Since the dynamic behavior analysis framework contains multiple platforms, information will be transformed several times among those formalisms with data consistency. In order to distinguish the notation, three concepts are employed. Participants in this armed conflict incident are defined as performers in DoDAF modeling based on the concept from DM2. In SysML executable platform, it is defined as an object which is an instance of its class type expressing which kind of system it belongs to and what attributes it has. The concept of entity is employed in the STK simulation scenario (specific components such as vehicles, aircraft, radars, etc.). All these concepts refer to the same thing in reality. From the data perspective, several types of data need to be extracted including performers, attributes of performer, association, association properties (such as strike relationships, reconnaissance relationships, communication relationships with related parameters and constraints), performers' possible states, and state transition conditions. From model perspective, model types of different software include OV-2, OV-6b, Object Model Diagram, State Machine Diagram, STK simulation scenario and Sequence Diagram need to be constructed or generated. The transition conditions used to execute the model can be triggered automatically in a fixed time or by an external message from simulation. To simplify the modeling process in the case study, the generated executable models based on DoDAF models will be described directly.

1) STRUCTURAL MODELING

In this conflict incident, in order to describe the process that F16 shot SU24 effectively, F16 and its ARMAAM missiles can be decomposed into two types of objects. A total of 14 objects can be collected as shown in Figure 5. The objects from left to right are Hmeimin Air Force Base which controls the Russia battlefield by sending operational orders and receiving battlefield information, two SU24 fighters which perform attack mission against terrorists, armed terrorists in Syria who killed the pilot and shot down Russian helicopters, armed MiG-8 helicopters which perform rescue mission, two pilots that one was shot and the other was rescued, two F16 fighters which perform air defense mission, ARMAAM air missiles, Syria detection radar and Turkey detection radar. Events like "SU24 have taken off", "receive striking command", and "be warned" will trigger the state transition between objects.

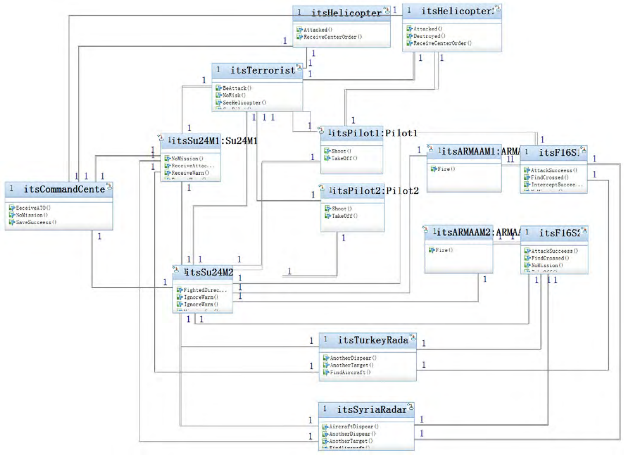


FIGURE 5. The object model diagram of the incident.

2) BEHAVIOR MODELING

The State Machine Diagram defines the objects' behavior by specifying their reaction to operational events. The reaction may be performed by the transition between objects' states. Figure 6 shows the states of SU24, F16, Russian pilots and Syrian radar. The state transition process of SU24 is executing its patrol mission after receiving the takeoff command, and arriving cruise area 20 minutes later. They attacked terrorists at 10:12 and completed the first bombing mission at 10:16. Then, they turned around for a second bombing. The mission was completed at about 10:24. If SU24 was in the cross-border state, it may receive a warning trigger and transfer to the "be warned" state. Whether a cross-border event of SU24 takes place requires being simulated from STK scenarios with more detailed geographic information.

The state transition process of F16 is also shown in Figure 6. F16 performed cruise mission after receiving take-off command. According to the data from Syrian radar, F16 got into the Syrian radar surveillance range at 9:11. Then F16 warned Russian fighter after finding that it had crossed the boundary and shot it down after five minutes' warning, or shot SU24 fighter directly.

C. LOGIC ANALYSIS OF THE RUSSIAN-TURKISH PLANE INCIDENT

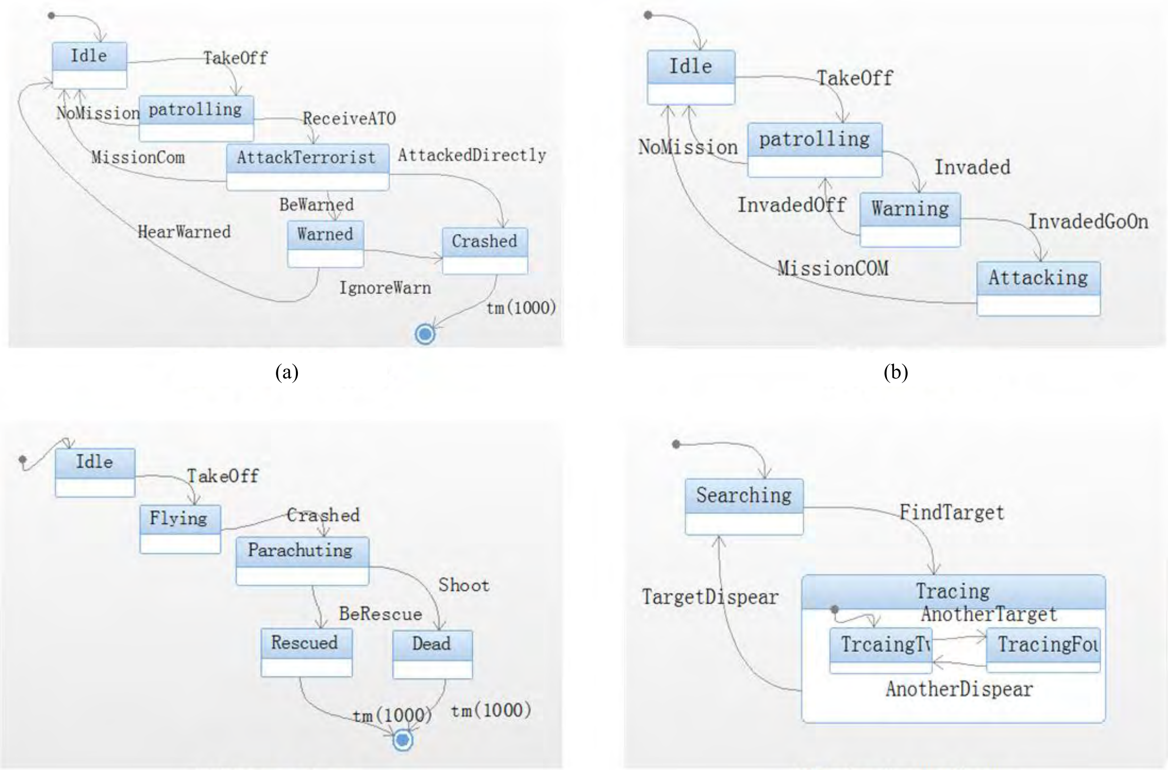
Sequence Diagram is the main form of dynamic behavior analysis results. Logical errors can be detected by analyzing the sequence of events and information exchange. By collaborating with simulation, the logical process considering time, geographic information and weapon performance can be obtained. According to the information published online, four points about this incident need to be verified as shown in Table 6.

TABLE 6. Doubts list of Russian-Turkish plane incident.

Doubts	Standpoint of Russian	Standpoint of Turkey
F16 take off time	Premeditated	null
SU24 transboundary	No	Crossed the boundary 1.36 km and 1.15 km
F16 warning	No	10 times warning in 5 minutes
F16 trans boundary and shot	Crossed the boundary 2 km and shot	Shot without crossing

1) SCENARIO 1: MODELING AND ANALYSIS BASED ON TURKEY'S REPORT

After the incident, Turkey claimed that these attack operations were in conformity with the international rules. When the invasion was detected, F16 continued to warn the cross-border fighters for about five minutes. However, there was



(a) (b)

(c) (d)

FIGURE 6. The state machine diagrams of objects. (a) Su 24M state diagram. (b) F16S state diagram. (b) Pilot state diagram. (b) Syria radar state diagram.

no response, so they took actions to shoot the cross-border fighters down to safeguard their homeland security. F16' operational process was divided into four steps. First, after detecting the Russian fighters' activities on the Turkey border, Turkey sent F16 fighters to take off from the nearest base to intercept. Second, F16 warned repeatedly the Russian fighters to confirm their identity and demanded them to leave the violated airspace. Finally, F16 took actions to launch missiles and shot Russian fighters down. By setting the takeoff time, flight speed and other parameters of F16, the simulation scenario of the Turkish report can be generated. Events like taking off, warning and attacking are defined as the transition conditions of states in State Machine Diagram. As shown in Figure 7, the red sector is the reconnaissance range of the Turkish radar and the green one is the range Syrian radar, the SU24' flight track is generated from the radar data. Blue curve stands for the flight track that F16 took off from the air force base directly to the border to intercept. The red curve is the track that Russian fighters performed bombing mission against terrorists near the border after receiving attack command. Whether the F16 can successfully intercept the Russian fighters within the territory of Turkey needs to be analyzed. The results of the execution will lead to differences in the final sequence diagram.

By changing the parameters of the F-16 fighter, the STK simulation result can help determine whether the F-16 fighter is premeditated in the air waiting for SU24 and shoot it down. The collaboration process of Rhapsody and STK is shown in Figure 8 meaning that when F16 and Su24 achieve a visible relationship in the simulation process, the state transition of warning will be triggered. Through all the steps above, models and data for the dynamic behavior analysis are ready, and the executable toolchain can be activated. During the execution, the Collaboration Engine component will continually monitor the simulation process and send event information about whether this event happens or not to Rhapsody, so that the State Machine Diagram can choose

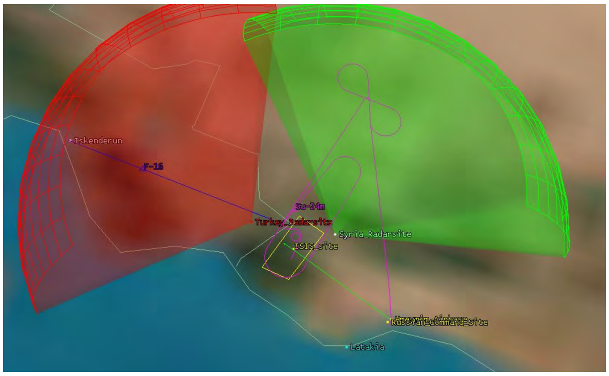


FIGURE 7. Scenario-based on Turkey report.

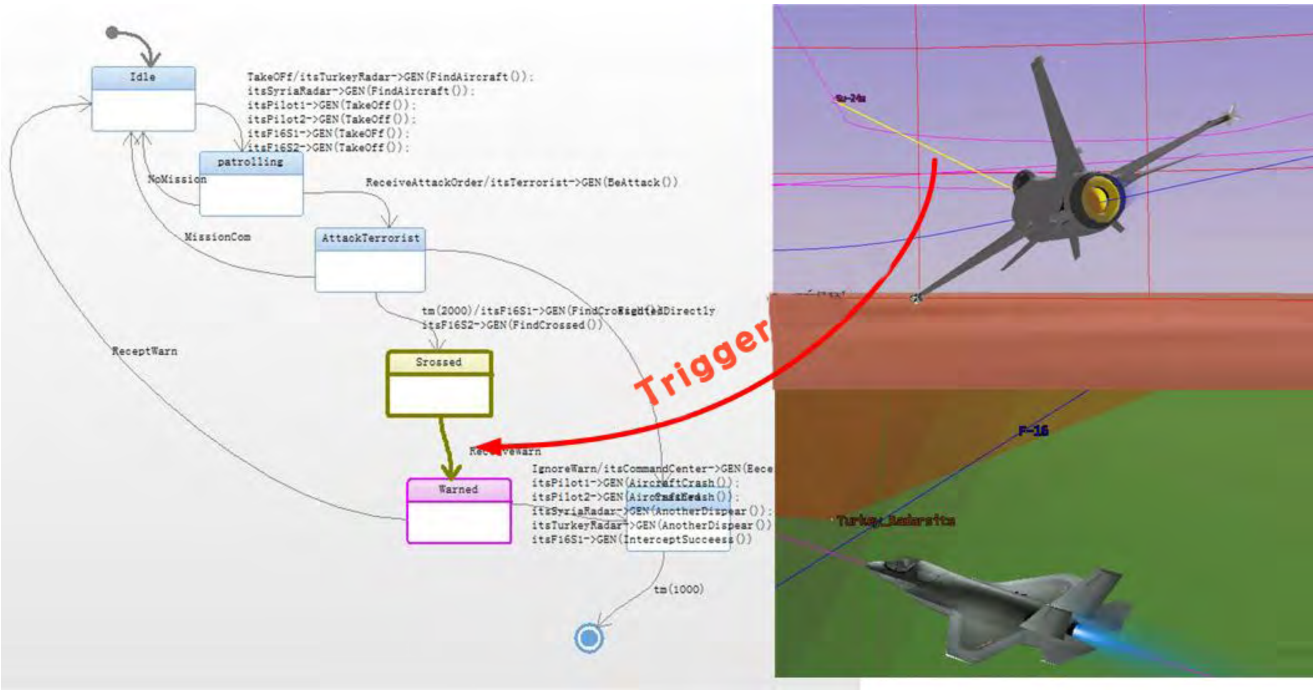


FIGURE 8. The interaction between executable state machine diagram and simulation.

correct transition options based on the specific event information. Figure 9 shows the state transition results in the form of a sequence diagram, the Russian SU24 fighters received the mission order and attacked the terrorist, then crossed the border. Turkish F16 fighter attacked Russian fighters after warning.

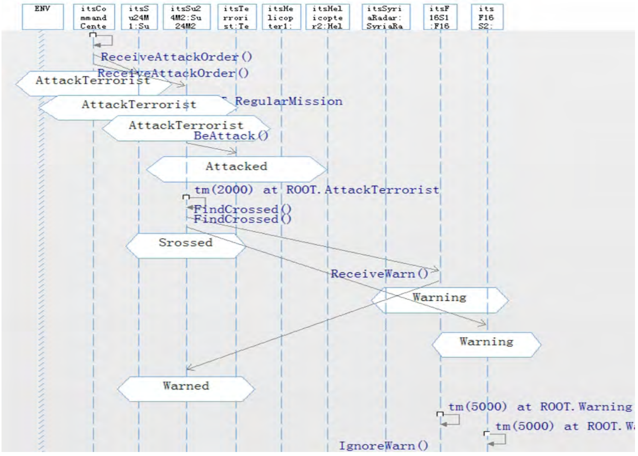


FIGURE 9. The sequence diagram of Turkish report.

2) SCENARIO 2: MODELING AND ANALYSIS BASED ON RUSSIA'S REPORT

On 24th November, Russia published the Su-24's roadmap to supplement the Russian fighter' cross-border issues. According to Syrian radar data, F16 fighters stayed in Syrian airspace about 40 seconds and 2 kilometers in depth, while the Russian fighters have never crossed the Turkish border. On the Turkey side, the F16 fighter took off from the 8th Air Force Base and shot SU24 down by air to air missile at 10:24 Moscow time. After the missile was launched, F16 speeded up and left below the height of the radar detection range. Moreover, there were no warning records at all.

According to the described situation above, the generated scenario is shown in Figure 10. It shows that the Russian fighters have never crossed the boundary. Moreover, according to the sequence diagram generated shown in Figure 11, there have been no warning actions activated.

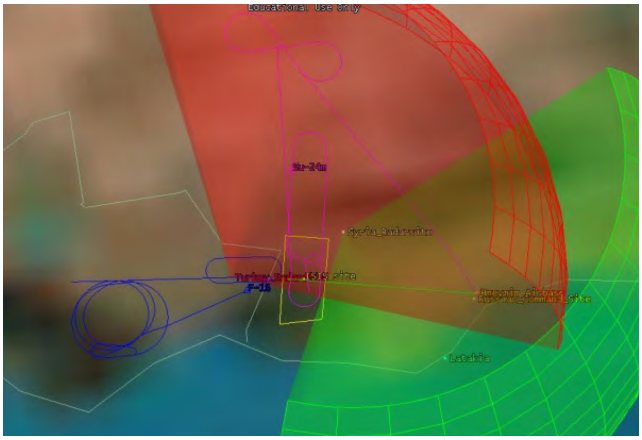


FIGURE 10. Scenario based on Russian report.

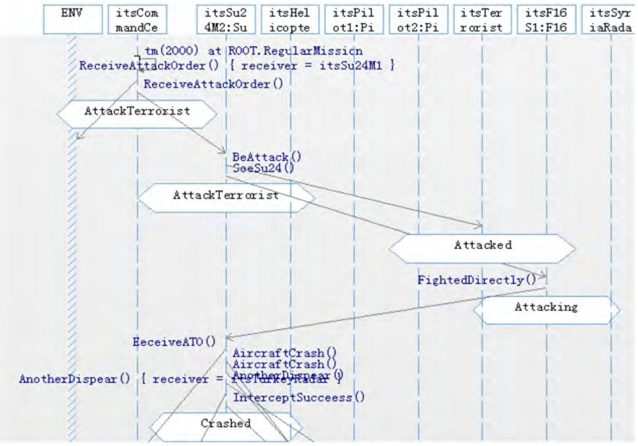


FIGURE 11. The sequence diagram of Russia report Comparison analyses.

By implementing the dynamic behavior analysis with the proposed approach, it is able to put forward some reasonable results and simple conclusions in Table 7 to verify the above doubts:

TABLE 7. Results of dynamic behavior analysis.

Main concerns	Turkish scenario	Russian scenario
F16 fly duration	20min(take off temporarily)	75min（patrolling）
SU24 fly duration	34min	34min
Transboundary time	20s(Su24 crossed)	40s(F16 crossed)
Warning time	10:20	No warning
Location when warning	26 kilometers away from the boundary	No warning
Height when attacking	5650m	5650m
Attack time	Cannot attack	10:25
Crash location	Cannot attack	4 kilometers from the boundary
Analysis result	Cannot reach the warning state	Logic well pass

First, by comparing these two scenarios, it is shown that if Turkey F16 took off from the nearest air force base after receiving an interception command, it cannot reach the conflict area in consideration of time and distance. As a consequence, it is more reasonable that the Turkey F16 fighter is patrolling nearby in advance rather than taking off temporarily based on the established data and the dynamic behavior analysis method. The results do not represent any official conclusion, only to verify the applicability of the proposed method.

Second, Turkey claimed that F16 fighters have warned SU24 of its invasion 10 times within 5 minutes and finally took attacking action without crossing the boundary. However, according to the real-time speed and position, it is not possible for F16 to complete the whole action process within the boundary.

In addition, if the SU24 transboundary flight path is generated based on the Turkish radar data, the execution result shows that the transboundary time is too short that F16 cannot complete the warning and attacking action neither.

V. CONCLUSION

Focusing on the logical validations in armed conflicts and the weakness of considering the logic process and scenario parameters simultaneously in traditional executable analysis methods, this paper proposes a novel dynamic behavior analysis methodology which intends to achieve new analytical capabilities by collaboratively integrating different techniques together. First, DoDAF framework and its relative artifacts (e.g., OV-2, OV-6b) are employed to describe the structural and behavioral information of the armed conflicts from various viewpoints. Then, the DoDAF models are transformed to SysML executable models and simulation scenarios for dynamic analysis from a data-centric perspective which ensures the automation of the conversion process and the semantic correctness. Next, an execution process is implemented with the Collaborative Engine developed based on a service-oriented and interfaces integrated technique to parallel the timeline and event information of Rhapsody and STK. A sequence diagram is generated to describe the interactions of systems under a predefined scenario. Finally, this paper takes a typical armed conflict incident with several logic doubts as a case study to verify the validity of the proposed approach and draw some meaningful conclusions.

In addition, more specialized analytical tools and big-data analysis techniques can be integrated to the collaborative multi-modeling dynamic behavior analysis framework. Then, the framework is also a meaningful way to rapid modeling, real-time analysis for different specific purposes or other systematic issues of the armed conflicts.

REFERENCES

[1] B. Beihoff *et al.*, "A world in motion: SE vision 2025," in *Proc. Incose*, 2014, pp. 1–9.

[2] A. L. Ramos, J. V. Ferreira, and J. Barceló, "Model-based systems engineering: An emerging approach for modern systems," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 1, pp. 101–111, Jan. 2012.

[3] L. W. Wagenhals and A. H. Levis, "Service oriented architectures, the DoD architecture framework 1.5, and executable architectures," *Syst. Eng.*, vol. 12, no. 4, pp. 312–343, 2009.

[4] P. Helle, I. Giblett, and P. Levier, "An integrated executable architecture framework for system of systems development," *J. Defense Model. Simul. Appl. Methodol. Technol.*, vol. 10, no. 4, pp. 435–448, 2013.

[5] A. Bueno, L. T. Carreño, D. J. Delgado, and R. Llamosa-Villalba, "Executable architecture based on system dynamics: An integrated methodology composed by standard system dynamics modelling and DoDAF operational view models," *Procedia Comput. Sci.*, vol. 36, pp. 87–92, 2014.

[6] C. Piaszczyk, "Model based systems engineering with Department of Defense Architectural Framework," *Syst. Eng.*, vol. 14, no. 3, pp. 305–326, 2011.

[7] B. Ge, K. W. Hipel, K. Yang, and Y. Chen, "A novel executable modeling approach for system-of-systems architecture," *IEEE Syst. J.*, vol. 8, no. 1, pp. 4–13, Mar. 2014.

[8] K. Griendling and D. N. Mavris, "Development of a DoDAF-based executable architecting approach to analyze system-of-systems alternatives," in *Proc. Aerosp. Conf.*, 2011, pp. 1–15.

[9] W.-C. Xie, X.-H. Yu, and P.-F. Zhu, "Model-transformation based executable model's construction of C4ISR architecture," *Syst. Eng. Electron.*, vol. 36, no. 8, pp. 1537–1543, 2014.

[10] G. Zhan, B. Ge, M. Li, and K. Yang, "A data-centric approach for model-based systems engineering," *J. Syst. Sci. Inf.*, vol. 3, no. 6, pp. 549–560, 2015.

[11] B. Ge, K. W. Hipel, K. Yang, and Y. Chen, "A data-centric capability-focused approach for system-of-systems architecture modeling and analysis," *Syst. Eng.*, vol. 16, no. 3, pp. 363–375, 2013.

[12] J. A. Lane and T. Bohn, "Using SysML modeling to understand and evolve systems of systems," *Syst. Eng.*, vol. 16, no. 1, pp. 87–98, 2013.

[13] W. C. Baldwin, B. Sauser, and R. Cloutier, "Simulation approaches for system of systems: Events-based versus agent based modeling," *Procedia Comput. Sci.*, vol. 44, pp. 363–372, 2015.

[14] R. E. Giachetti, "Evaluation of the DoDAF meta-model's support of systems engineering," *Procedia Comput. Sci*, vol. 61, pp. 254–260, 2015.

[15] K. Li, X.-M. Li, and D. Liu, "Modeling method for weapon system of systems structure based on semantic meta-model," *Syst. Eng. Electron.*, vol. 37, no. 11, pp. 2503–2512, 2015.

[16] F. Pfister, V. Chapurlat, M. Huchard, C. Nebut, and J.-L. Wippler, "A proposed meta-model for formalizing systems engineering knowledge, based on functional architectural patterns," *Syst. Eng.*, vol. 15, no. 3, pp. 321–332, 2012.

[17] E. Garcia and P. J. Antsaklis, "Model-based event-triggered control for systems with quantization and time-varying network delays," *IEEE Trans. Autom. Control*, vol. 58, no. 2, pp. 422–434, Feb. 2013.

[18] J. Li, Y. Tan, K. Yang, X. Zhang, and B. Ge, "Structural robustness of combat networks of weapon system-of-systems based on the operation loop," *Int. J. Syst. Sci.*, vol. 48, no. 3, pp. 659–674, 2017.

[19] J. Sun, B. Ge, J. Li, and K. Yang, "Operation network modeling with degenerate causal strengths for missile defense systems," *IEEE Syst. J.*, vol. 12, no. 1, pp. 274–284, Mar. 2018.

[20] K. A. Griendling, S. Balestrini-Robinson, and D. N. Mavris, "DoDAF-based system architecture selection using a comprehensive modeling process and multi-criteria decision making," in *Proc. 12th AIAA/ISSMO Multidisciplinary Anal. Optim. Conf. (MAO)*, 2008, pp. 10–12.

[21] G. Muller and C. Dagli, "Simulation for a coevolved system-of-systems meta-architecture," in *Proc. 11th Syst. Syst. Eng. Conf.*, 2016, pp. 1–6.

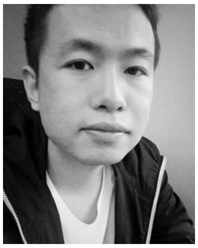
[22] D. R. Wibben and R. Furfaro, "Model-based systems engineering approach for the development of the science processing and operations center of the NASA OSIRIS-REx asteroid sample return mission," *Acta Astronaut.*, vol. 115, pp. 147–159, Oct./Nov. 2015.

[23] G. Hemingway, H. Neema, H. Nine, J. Sztipanovits, and G. Karsai, "Rapid synthesis of high-level architecture-based heterogeneous simulation: A model-based integration approach," *Simulation*, vol. 88, no. 2, pp. 217–232, 2012.

[24] D. Garlan, K. M. Carley, B. Schmerl, M. Bigrigg, and O. Celiku, "Using service-oriented architectures for socio-cultural analysis," in *Proc. 21st Int. Conf. Softw. Eng. Knowl. Eng. (SEKE)*, 2009, pp. 370–375.

[25] T. Denton, E. Jones, S. Srinivasan, K. Owens, and R. W. Buskens, "NAOMI—An experimental platform for multi–modeling," in *Model Driven Engineering Languages and Systems* (Lecture Notes in Computer Science and Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Berlin, Germany: Springer, 2008, pp. 143–157.

[26] A. A. Jbara, A. H. Levis, and A. K. Zaidi, "On using multiple interoperating models to address complex problems," *Procedia Comput. Sci.*, vol. 16, pp. 363–372, 2013.



MINGHAO LI received the M.S. degree in management science and engineering from the National University of Defense Technology, Changsha, China, in 2014.

He is currently pursuing the Ph.D. degree with the College of Systems Engineering, National University of Defense Technology. In 2016, he was a Visiting Scholar with Leiden University, Leiden, The Netherlands. His current research interests include model-based system engineering, defense acquisition, system-of-systems modeling, and optimization.

JIANGUO XU received the M.Tech. degree in economics and management from the National University of Defense Technology, Changsha, China, in 2016, where he is currently pursuing the Ph.D. degree of management science and engineering with the College of Systems Engineering. He has extensive systems and software engineering experience in domains of Weaponry and Aerospace. His main research interests include model-based system engineering, architectures, and their application to technological system.

KEWEI YANG received the B.E. degree in systems engineering and the Ph.D. degree in management science and engineering from the National University of Defense Technology, Changsha, Hunan, China, in1999 and 2004, respectively. From 2011 to 2012, he was a Visiting Scholar with the Department of Computer Science, University of York, York, U.K. He is currently a Professor of management science and engineering and the Director of the System-of-Systems Engineering Laboratory. His research interests focus on intelligent agent simulation, defense acquisition, and system-of-systems requirement modeling.

BINGFENG GE (S'11–M'14) received the B.E. degree in systems engineering, the M.E. and Ph.D. degrees in management science and engineering from the National University of Defense Technology, Changsha, Hunan, China, in 2006, 2008, and 2014, respectively.

He was a Visiting Scholar with the Conflict Analysis Group, Department of Systems Design Engineering, University of Waterloo, Waterloo, ON, Canada. He is currently a Lecturer of management science and engineering with the National University of Defense Technology. His research interests include system-of-systems architecting and engineering management, portfolio decision analysis, and conflict resolution.

Dr. Ge is a Technical Committee Member of Conflict Resolution of the IEEE Systems, Man, and Cybernetics Society and a member of the IEEE Internet of Things Technical Community and the International Council on Systems Engineering.

• • •