RULE BASED TARGET EVALUATION AND FIRE DOCTRINE

Mehmet Fatih Hocaoğlu

Department of Industrial Engineering, Faculty of Engineering and Natural Sciences
Istanbul Medeniyet University
Göztepe İstanbul, Turkey
mfatih.hocaoglu@medeniyet.edu.tr

ABSTRACT

Threat evaluation is the process in which threat values of targets are calculated, based on the inferred capabilities and intents of the targets to give damage to defended assets. The capability and the intent are measured by its capacity of giving damage and by its actions, respectively. While having warhead and its estimated power show a damage capability measure, being aimed a defended unit is a measure of hostile action. Target evaluation is proceeded in real time in a wargame and the aim is to compare weapon target pairs according to a set of criteria. The target evaluation cycle is repeated anytime a new detection received and when any change is happened in the target currently detected. In this paper, an agent based command and control entity, which is in charge of target evaluation and giving engagement decision, is designed and it is situated in an air defense simulation environment.

Keywords: Agent driven Simulation, Air defense simulation, Command & Control, Logic programming, Target evaluation

1 INTRODUCTION

The task of the air defense is to evaluate the tactical situation in real-time, and to protect the defended assets by allocating available weapon systems to threatening enemy targets (Roux and Vuuren 2007). In an air defense situation the threats are comprised of aircrafts and air to ground missiles. Weapon Target Assignment (WTA) problem is studied in the literature as given below as an optimization problem. In this study, a Greedy algorithm based WTA solution (Tin G. and P. Cutler 2006) is considered. This paper provides an agent based target evaluation and weapon selection. A target evaluation process is designed based on a combination of plugin architecture and rule based architecture and a rule based weapon selection is designed to select a weapon for the target aimed. The whole functionality is composed under an agent that is situated in an air defense simulation system. The agent is responsible for collecting detection from sensors it has, evaluating weapon and target pairs, and selecting the target-weapon pair with the highest score. The evaluation process is defined as a scoring process using weighted criteria set. A criteria may consist of parameters from target, from weapon and from the environment.

In this paper, an agent based target evaluation and weapon selection are studied and the idea is exemplified using an air defense simulation example. For this purpose, an agent named as “C2 Agent” is developed to evaluate targets detected by sensors and to give fire decision. The paper is organized as follows. Since the whole simulation and development including execution are developed based on AdSiF (Agent driven Simulation Framework), in Section 2, a brief explanation about AdSiF is given. In Section 3, a literature survey is given for the target evaluation concept. In Section 5, data fusion for target position and target identification is described and how it is used is explained. In Section 5, the algorithms for target evaluation and for weapon selection are considered. The agent architecture is presented in Section 6 and an air defense example is given in Section 7. The paper gives a discussion about the solution in Section 8.

2 ADSIF: AGENT DRIVEN SIMULATION FRAMEWORK

AdSiF (Hocaoglu 2005; Hocaoğlu 2018) is an agent and simulation development environment that enables continuous and discrete event simulation execution in an agent architecture. AdSiF combines multiple programming paradigms into state-based programming paradigm as an integrated paradigm. These paradigms have been defined as object-oriented programming, logic programming, and aspect oriented programming (Hocaoglu 2017; Hocaoğlu 2013). State-based programming diagrams provide a scripting language based on which all semantic structures of each entity developed are modeled by this scripting language. The atomic functions that the behaviors access are defined as functions developed in the C ++ language. There is no semantic flow in the C ++ class structure of the entity as whole semantic flow is kept in behavior declaration script. The simulation realizes the simulation execution by interpreting the kernel interpreter model scripts. Because of the agent architecture and event driven architecture that the entities developed in AdSiF are based on, the interactions between entities are provided as event transmissions.

Since AdSiF is also based on agent oriented programming, each model has its own dual world representation. A dual world representation is defined as an inner representation of an environment to be simulated, and is constructed of time-stamped sensory data (or beliefs) obtained from that environment, even when these data consist of errors. Time-stamped facts represent the models' own earlier values of state variables along with those of other models in the environment. In this way, dual world representation is a good candidate to represent conflicting behaviors and knowledge as is happened in the real world.

3 TARGET EVALUATION AND WEAPON SELECTION

Threat evaluation is the process in which threat values of targets are calculated, based on the inferred capabilities and intents of the targets to give damage to defended assets. In this sense, it is possible to talk about three types of target threating indexes. The first one is proximity parameters, which is an important class of parameters for assigning threat values to target-defended asset pairs measuring the target's proximity to the defended asset. The second class is capability parameters, which refers to the target's capability to threaten the defended asset. The last class is defined as intent parameters. The class of intent parameters is a broad category, containing parameters that can reveal something about the target's intent. An example of this is the target's kinematics (Johansson and Falkman 2008) according to (Oxenham 2003), the target's velocity (i.e. speed and heading) in combination with its altitude can be a good indicator of the target's intent to attack a defended asset. Another kinematics parameter that can be used is the number of recent maneuvers (Liebhaber 2002).

Data fusion is a process collecting and merging diverse piece of data from multiple sensors into a single whole to achieve refined positions and features of individual targets, to complete the tactical or strategic situation assessment and the hostile force threat assessment (Changwen and You 2002). Because of uncertain and blurred characteristics of the sensory data and the fused detection data, fuzzy theory has a widespread usage in target evaluation (Kumar and Dixit 2012). In navy, because of high variety of littoral environments, fuzzy knowledge based systems are used for target evaluation in operations including air contact (Liang 2007). An intuitionistic fuzzy inference (Atanassov 1986) is introduced to handle quantitative and qualitative factors together effectively (Dongfeng, Yu, and Yongxue 2012). Merging fuzzy theory and rule based systems, such as expert systems studies, which prepare a base for fuzzy knowledge base systems for target evaluation, are studied in (Chalmers 1988; Endsley 1995; Harris 1988; Negoita 1985).

Bayes theory and relatedly Bayesian networks are widely used in target evaluation. A Bayesian network framework for situation assessment is proposed by (Bladon, Hall, and Wright 2002) and a Bayesian network solution taking intend, capability, and proximity parameters into consideration is proposed by (Johansson and Falkman 2008).

As a decision making problem, it is seen that threat evaluation problem is modeled a single participant multiple attribute (SPMA) decision making problem. SPMA decision making is often referred in the literature as multiple attribute, multiple criteria or multiple objective decision making (Changwen and You 2002). In the formulation, discrete alternatives are evaluated against different kinds of attributes (criteria) in order to obtain a ranking of the alternatives.

As a general resource allocation problem, WTA problem focuses on defensive perspective, but some have considered offensive perspective (Sikanen 2008), wherein the objective is to maximize the probability of destroying enemy protected assets. WTA problem consists of optimally assigning *n* weapons to *m* targets so that the total expected survival value of the targets after all the engagements is minimum (Ahuja et al. 2007) or the target destroyed is maximum. Many different problem formulations are developed to maximize destruction level or minimize survival probability (Kline, Ahner, and Hill 2019). The problem can be categorized as a combinatorial optimization problem proved to be nondeterministic polynomial time complete (S. P. Lloyd and S.Witsenhausen 1986) and the problem is considered as a specific case of the more general resource allocation problem (Plamondon and Chaib-draa 2004; Zhang 2002). Since resources are usually constrained, the allocation of resources to one task restricts the options available for other tasks. The action space is exponential according to the number of resources, while the state space is exponential according to the number of resources and tasks. The very high number of states and actions in this type of problem coupled with the time constraint makes it very complex (Plamondon and Chaib-draa 2004). A closed loop WTA process is not only NP-Complete but also, is it discrete, dynamic, non-linear, stochastic, and of a large scale in terms of increased number of WS and targets (Allouche 2005) and it is difficult to solve it optimally when the number of threats and weapons is large, as computation time of the solution increases rapidly with the size of problem (Naeem, H., Masood 2010). An asset-based DWTA problem is formulated and a rule-based heuristic to solve it is proposed by (Xin, B., Peng 2011). Despite its study to decrease the computational time, the problem was still treated in exponential complexity (Cook and Skinner 2005). A nonlinear goal programming solution is developed by Hocaoğlu (Hocaoğlu 2015). The model optimizes fire doctrines selection for each target, depending on user choice either optimize missile types and numbers for a complete destruction or finds destruction level to be able to be got with limited number of missiles, and it also considers setup times between consecutive engagements. In this paper, a greedy weapon target assignment that is repeated at each target detection based on target score is studied, instead of trying to optimize weapon target assignment considering all targets at a time.

Some swarm optimization algorithms such as ant colonies (Hu et al. 2018) and hybrid algorithms consisting of improved artificial fish swarm algorithm (AFSA) and improved harmony search (HS) (Chang et al. 2017) are applied to WTA problem. Using ant colonies algorithm, a solution is developed considering WTA problem as a bi-objective WTA (BOWTA) optimization model which maximizes the expected damage of the enemy and minimizes the cost of missiles (Li et al. 2017).

4 DATA FUSION

Data fusion is the process of integrating multiple data sources to produce more consistent, accurate, and useful information than that provided by any individual data source (Haghighat, Abdel-Mottaleb, and Alhalabi 2016). In this study, decision level data fusion is considered rather than sensory signal level and it is used for two purposes. The first purpose is to estimate the detected target position using position information collected from different sensor detections. Each sensory data may give different but close position information with mostly different time label. The idea is to combine sensory data that are received from the sensors and weighed by importance and reliability of the sensor they are received from. The combining function is selected as an weighted average function. The combined, in other words, fused position information is used as the target position information. The second purpose is to identify the target. A rule based identification reasoning mechanism based on first order reasoning is developed. The detection parameters such as target velocity, target altitude, target physical properties (detected using day cam or thermal cam) are used as input parameters for the predicate identification. The target detection parameters are stored in a knowledge base in a fact form and they are populated by the detection received from the sensors. The fused target position information is used in engagement as the target position for the target to be engaged and as a guidance information to a missile that has already been fired at a target. The C2 agent, which is in charge of target evaluation and weapon selection, is the owner of the knowledge base. The detection information and fusion algorithms are defined as facts and predicates, respectively as components of logic programming. The out-parameters predicate seen in Algorithm **1** are fused target position components. The predicate calculates the weighted average of each position subfield using detection fact parameters kept in the knowledge base. As seen in Algorithm 1, for a target that is represented by the variable A, the predicate calculates a detection time weighted position average using detections taken from radar (coded as 1) and laser range finders (coded as 4).

Identity fusion uses the detection information received from different types of sensors and fuses the information they provide using a set of rules. The whole process is called rule-based identification fusion and it is used to determine target types based on the information kept in the detection facts, such as target velocity, target position and target dimensions. The rules are a set of Prolog predicates and an example rule is shown in Algorithm **2**. The predicate is interpreted as If "(800 ≥ Target.Velocity ≥ 200 & 10 ≥ Target.Width≥ 3 & 15 ≥ target.Lenth ≥ 4 & Target.Height≥ 50)" Then Target is a plane.

```
list_sum([], 0).
list_sum([Head | Tail], TotalSum) :-list_sum(Tail, Sum1), TotalSum is Head + Sum1.
fusedPosition(A,Avg):-findall(X*T, (detection(S,A,X,T),member(S,[1, 4])),L), list_sum(L,Total),
                      findall(T, detection(S,A,X,T),L2), list_sum(L2,Total2), Avg is Total/Total2.
```

Algorithm 1: Position fusion predicate.

The target velocity comes from radar (coded as 1) and the size information comes from day cams (coded as 2). The predicate return parameter is associated with the C2 agent. Any time that the predicate turns to 'true' it makes the behavior to change the phase with which it is associated. Depending on how it is associated, it activates the behavior if it is an activation drive condition, cancels the behavior if it is a cancel drive condition, suspends the behavior if it is a suspension drive condition, or reactivates the behavior if it is a resume drive condition.

```
decision(Target, plane):-detection(Sensor, 1, Target, _,_,PosZ, Vel, _, _, _, _, Ac1, DistanceRatio0,_ ),
        detection(Sensor, 2, Target, _,_,_, _, _, _, Width, Length, Ac2, DistanceRatio1,_),
        Vel>=200, 800>=Vel, PosZ>=50, Width>=3, 10>=Width, Length>=4, 15>=Length,!.
```

Algorithm 2: An Example for identity fusion predicate.

After this point, we have a set of identified target with fused position information. The next step is to evaluate targets and select a weapon for the one with highest score.

5 TARGET EVALUATION AND WEAPON SELECTION

The first step is to prepare weapon and target pairs. Each identified target is matched with the weapons if there is no rule prohibiting in fire doctrine rule set. The fire doctrine declares what weapon is chosen for what target and what is their engagement rule. The engagement rule is given as a salvo fire pattern in the form of a fact.

fireDoctrine(trgtType, wpnType, M, N, K, T0, T1); M: number of fires, N: Number of fire repeat, T0: duration between two successive firings, K: number of cycles (millisecond), T1: duration between cycles (millisecond). N defines how many missiles to be fired in T0 time intervals, while K represents how many times this cycle to be repeated in T1 time interval. For example, a single shot fire doctrine is defined as *fireDoctrine(targetType, weaponType, 1,0,0,0,0)* and no engagement is defined as *fireDoctrine(targetType, weaponType, 0, 0, 0, 0, 0)*. The rule base looks as below for the target f16;

fireDoctrine(f16, patriot, 1,0,0,0,0)
fireDoctrine(f16, harpoon, 0,0,0,0,0)
fireDoctrine(f16, ciws, 10,3,1,2,3)

Target evaluation is to calculate a score according to a set of weighted criteria to measure suitability of a weapon for a given target. The criteria is defined as predicates and plugin functions to make the evaluation process flexible and reusable. The selected criteria set is given below and it is not limited by them. The term target is defined as a platform such as a missile, a plane, or any movable platform that is considered as a target.

- Target arriving duration; the smaller duration is the higher score. All opponent entities such as missiles or fighters aim an entity, a point, or a specific region. Their calculated arriving duration that is based on sensory data to their targets are normalized and the target score is increased by its normalized duration.
- Target priority; The higher priority is the higher score. The target priority is determined by the component with the highest priority the target has or carries.
- Target damage level; The lower damage is the higher score.
- The Probability of Kill; The higher probability of kill is the higher score.
- Number of ammunition; The tendency to choose is the abundant weapon.
- Atmospheric effects; Some weapons are not proper to use in specific weather conditions. Depending of weather condition level, a weapon target pair is scored to give a measure for the suitability level.
- Engagement duration; it is not expected all weapons have the same engagement duration. Fire doctrine is effective on determining engagement duration. The shorter engagement duration is a perfectible property for a weapon target pair because of the fast response.

For each target-weapon pairs, a score is calculated using the criteria set by weighting each. The weapon-target pair is chosen for the engagement.

6 THE AGENT ARCHITECTURE AND BEHAVIOR DIAGRAM

The C2 agent has a set of relations with the entities in the scenario. These are weapon systems to give fire order, to match weapons with targets to evaluate weapon target pair scores and sensors to collect detections. The relations are defined in scenario design time and they provide a media for event interaction, in other words, communication between entities (Hocaoğlu 2018). In addition to this, relations force entities that a relation is established between, a set of specific action in both phases of the relation establishment and in the relation breaking. In our case, beyond the relation is a communication channel, a relation such as "*Fires*" established between a launcher and a commander agent, forces commander to activate the behaviors related with target evaluation and engagement. In other word, the relation activates a set of behaviors representing an agent behavioral aspect. For example, the second commander in the scenario has no launcher related with it, because of that, it deactivate target evaluation and engagement behavior sets and activates the detection reporting behaviors. In this behavior set, the commander sends the detection it receives from the sensors to the commander it connected with. The two important architectural characteristics are event driven architecture and plugin architecture. The entities in the simulation such as sensors, weapon systems, and C2 agents communicates each other sending events. Each event trigs a set of behavior on the receiver side. The agent changes the target evaluation algorithm depending on the satisfied condition, this is modeled as different aspect of the agent model and this is named as aspect based agent modeling (Mehmet Fatih Hocaoğlu 2018).

As seen in Figure **1**, the C2 agent starts by its initial behavior *Fsa_Alive.* the behaviors of *Fsa_UpdateTarget* and *Fsa_TargetEvaluation* are activated by the event *detection* that is sent by the sensors related with the

C2 agent. The behavior *Fsa_UpdateTarget* adds the target to the target list in the C2 agent or if it is already in the list the target information is updated in the state *Update Detection List* entry phase function *UpdateDetectionList*. At the same phase, the state also updates the agent knowledge base by creating a detection fact for each detection received.

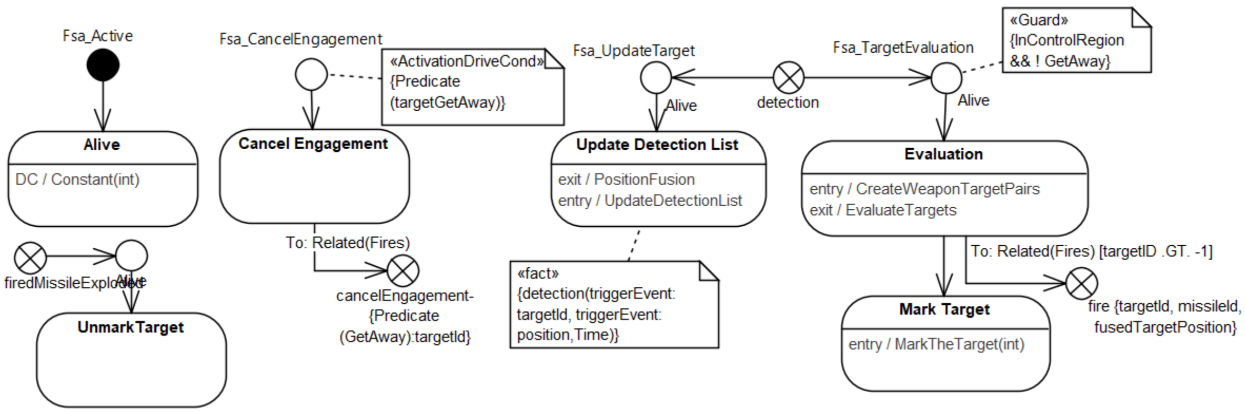

Figure 1: The C2 Agent Behavior Diagram.

The behavior *Fsa_TargetEvaluate* activates the state Evaluation first. In the state, weapon target pairs are created at the state entry phase (*CreateWeaponTargetPairs*) and weapon target pair score calculation at the exit phase (*EvaluateTargets*). At the exit phase, if there is a weapon target pair with the highest score is selected as the target to be engaged and the event fire is sent to the launcher. The event parameter is shown in the figure in cruelly parenthesis. The values to be bonded to the parameters are taken from the weapon target pair selected. For the target position of the event *fire*, the fused position is used (*fusedTargetPosition*).

The launcher that is related with the C2 agent by the relation "*Fires"* receives the event and selects the missile given by the event (by *missileId* parameter) and fires it to the target given by the parameter *targetID*. At the state following (Mark Target), the C2 agent marks the target as fired. When the missile explodes, it sends the event *firedMissileExploded*. The event trigs *Fsa_UnmarkTheTarget* and the behavior drops the target from the target list if it is dead otherwise it unmarks the target in *UnmarkTarget* state and it is put in the target list again. All of the behaviors except *Fsa_Active* are parallel, that means, the behaviors process all event received at the same time in parallel.

The behavior *Fsa_CancelEngagement* is activated by a predicate named as *GetAway*. If the predicates returns a *true* truth value, the behavior is activated and cancels the engagement for the target given by the predicate return value (cancelEngagement-{Predicate(GetAway):targetId}). The predicate is given in Algorithm **3**. The behavior sends an event named as *cancelEngagement* to the weapon system to terminate engagement or even it is not considered as a target as is done in *Fsa_TargetEvaluation* guard condition.

gettingFarAway(Target,true):-detectionDistance(Target, Dist0, Time0),
detectionDistance(Target, Dist1, Time1), detectionDistance(Target, Dist2, Time2),
Dist0>Dist1, Dist1>Dist2, Time0>Time1, Time1>Time2.
gettingFarAway(_,false).

Algorithm 3: Engagement Termination Predicate.

As seen from the execution semantic, the whole simulation is driven by state automaton based behavior interpretations. A rule based reasoning is used to drive the agent behaviors.

7 EXAMPLE: AIR DEFENSE SCENARIO

In the scenario given in Figure **2** and its assets in Table **1**, The red fighters aim the blue air defense systems. Both blue air defense system have launchers, missiles, and radars. Long range and medium range missiles have seekers and warheads. In this sense, they are guided, thrusted, multi-phase missiles.

Table 1: The Scenario Assets.

Entity Type		Side	# of Entity
Fighter (F16)	F16-1, F16-2, F16-3, F16-4	Red	4
Choppers	Hel-1, Hel-2	Red	2
Defense site (Long Range)	FFS-1, FFS-2, FFS-3	Blue	3
Defense site (Medium Range)	KMS-1, KMS-2, KMS-3, KMS-4	Blue	4
Defense site (Short Range)	AAA-1, AAA-2	Blue	2
Missiles		Blue	28
Radar		Blue	4
Seeker		Blue	28
Warhead		Blue	28

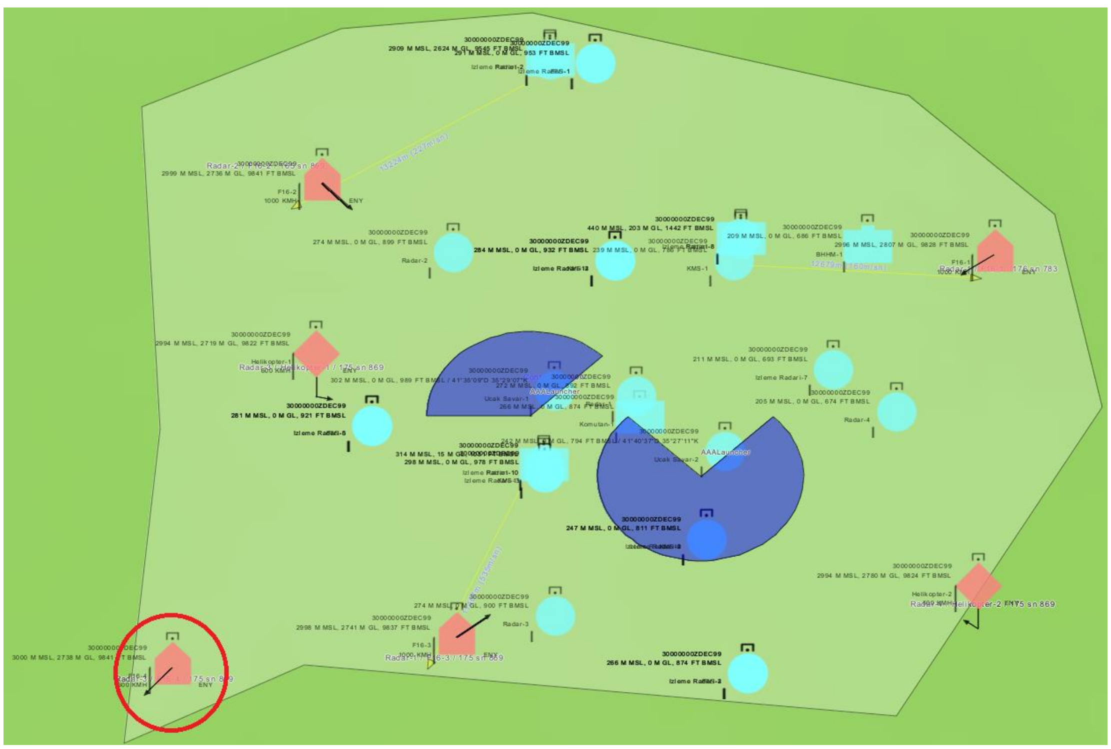

Figure 2: Air Defense Scenario.

The red fighters from three different directions and the red choppers from two opposite directions start to follow their routes and enter to the blue defense system control zone. The fighters depart to their destination almost 2 minutes earlier than the choppers. The red forces head to the directly to the blue defense unit located at the center of defense zone. The defense system radars detect the red fighters. The radars send their detections to the command and control system they are related with. The relation "Sends Detection" is established between the sensors and the commanders and the relation "Fires" is established between the command and control system and launchers. Each launcher is connected with a C2 system. Until the red forces enter to the defense zone, the C2 agent does not consider them as targets. The C2 agent creates weapon target pairs, evaluates and give a score for each of them. After scoring, it chooses the pair with the highest score for each defense site. There is no engagement started with the platform shown in the circle. Since it is getting far away, it is not evaluated as a target.

Since AAALauncher-1 is closer to Chopper-1 than AAALauncher-2, it has higher score and it is chosen to engage. Similarly, AAALauncher-2 is chosen for Chopper-2. For each target, the most proper missile is selected using target – weapon pair scores. In selection process, round wheel algorithm that the higher score gives the higher selection probability is used. The function is defined as a plugin to be able to change with any another function depending on the conditions defined.

The fired missiles are seen in Figure 2. A yellow line from a missile to a target represents which target is aimed by which missile. A layered defense architecture is applied. The long range missiles are deployed at the outer layer. The middle range missiles are deployed at the behind of the long ranges. The short range anti-aircraft artilleries (AAA) surround the command and control center. Because of the fire doctrine rules being applied in the scenario, it is not allowed to engage the choppers with missiles. The only way is to engage by AAAs and because of the same reason, the single shot fire doctrine is applied for the F16s.

The scenario is run for two different configurations. In the first configuration, the red forces fly at higher altitude (3000 m) and the second case, they follow the lower routes (~2000 m).

At the end of scenario, the blue land forces defend its area successfully for two cases. How long the red fighters fly in two scenarios is shown in Table **2**.

Table 2: Fly Durations.

Red Fighters & Blue Missiles	**High Altitude Scenario**	**Low Altitude Scenario**
FFSMissile-1	41,359	39,519
FFSMissile-3	35,433	34,347
FFSMissile-5	41,182	39,130
F16-1	79,518	72,932
F16-2	57,444	50,465
F16-3	86,353	80,215
Chopper-1	43,219	37,001
Chopper-2	176,238	172,506
Mean	70,09325	65,76438

According to the Chi-Squire test, the difference between the two scenarios are randomly different at %95 level. The P value is equal to 0,999475.

The engagement times with the related targets are given in Table **3**.

Table 3: Target Engagement Times.

	High Altitude Scenario		**Low Altitude Scenario**	
Defense Site	Target	Engagement Time	Target	Engagement Time
FFS-1	F16-2	136,086	F16-2	130,947
FFS-2	F16-1	164,086	F16-1	158,086
FFS-3	F16-3	165,172	F16-3	161,086
AAA-1	Helikopter-1	82,204	Helikopter-1	76,204
AAA-2	Helikopter-2	94,095	Helikopter-2	88,095
Mean		128,3286		122,8836

According to Chi-Squire test results flight durations are not different randomly. There is a meaningful difference between the two scenarios. The calculated p value is 5,79837E-18. The engagements are succeeded in a short time in “Low Altitude Scenario” (Table **3**).

In Table **4**, the distances between target red air platform and missiles aimed them are shown with the first distance when they are fired. In collision calculation, the physical dimension of the entities are taken into account. In this sense it is seen the targets are eliminated by hitting.

Table 4: Target Elimination Distances.

Missiles	**High Altitude Scenario**		**Low Altitude Scenario (unit meter)**	
	Explosion Distance	Engagement Distance	Explosion Distance	Engagement Distance
FFSMissile-1	20,7444	16307,5	12,0526	16534,5
FFSMissile-3	10,2273	19013,2	8,45495	19480,7
FFSMissile-5	20,5852	12007	17,2139	11938,1

Detections of sensor systems are given in Table **5**. The table consists of the first detection times and number of detections. According to the statistic test, the difference between two scenarios are not random (P= 3,06635E-14). The targets at low altitude are detected first.

Table 5: Target Detection Times and Detection Numbers.

	High Altitude Scenario		**Low Altitude Scenario**	
	First Detection Time	# of Detection	First Detection Time	# of Detection
Chopper-1	82,000	8	76,000	8
Chopper -2	97,000	15	91,895	10
F16-1	158,000	54	154,000	54
F16-2	161,000	10	130,861	15
F16-3	165,086	57	161,000	52
Mean		28,8		27,8

8 DISCUSSIONS

In this study, an agent based means is shown for target evaluation and target weapon assignment with reasoning. The target evaluation criteria set is selected among the parameters that of belonging the target and that of belonging the weapon system. For each criteria, a predicate based on logic programming or a plugin function based on plugin architecture is defined. A total score is calculated by adding the score calculated for each criteria for each target weapon pair. The target weapon pair is selected to engage. During target weapon pair preparation, fire doctrines are considered. For example, if it is not allowed to fire a weapon to a specific target type, the weapon and target is not paired. Similarly, the salvo patter of the weapon for a specific target is also considered to calculate total engagement duration and resultant probability of kill.

The agent has a reasoning mechanism with a knowledge base that is equipped with a set of facts and predicates. The facts are populated or updated depending on its knowledge type by the information received from the sensors the agent connected with. The sensors are the only way to pick information from the environment. The predicates allow the agent to judge about a target or a situation by making a decision considering environmental conditions, parameters of target and weapon systems it has to take an action and they return a truth value and a set of return parameters.

The agenthood nature bring some important flexibility to the simulation such as adding or removing new evaluation criteria, extending new selection functions, and reasoning based decision making. Since the weapon selection and target evaluation rules and calculation procedures are defined as logic programming predicates and plugin functions, the solution allows modelers to be able to extend rule base and plugins without changing the simulation model source code and it is possible to do that even in run time. During simulation execution, a new predicate can be sent to the model as a message interaction and it populates its knowledge base and uses it to evaluate weapon target pairs and/or uses it as a weapon selection rule.

A rule based target evaluation and weapon assignment gives results faster than non-linear optimization models, and for a simulation environment, a fast solution is one of the most important criteria. Possibly it does not find the best weapon assignment if it is compared with non-linear models studied earlier (Hocaoğlu 2015), but according the selected criteria set it still finds acceptable and applicable results.

The study advances the state of the art with regards agent-based modeling in accommodating an agent design in military decision making domain, bringing analytic methods with the first order logic together, and combine aspect orientation with agent design. The study also improves dynamic aspect management in agent programming using the relation concept.

ACKNOWLEDGMENTS

The study is supported by *Agena Defense & Information Technologies Ltd.* www.agenabst.com.

REFERENCES

Ahuja, R. K., a. Kumar, K. C. Jha, and J. B. Orlin. 2007. "Exact and Heuristic Algorithms for the Weapon-Target Assignment Problem." *Operations Research* 55(6):1136–46.

Allouche, M. K. 2005. "Real-Time Use of Kohonen's Self-Organizing Maps for Threat Stabilization." Pp. 153–63 in *Proceedings of the 8th International Conference on Information Fusion*.

Atanassov, K. 1986. "Intuitionistic Fuzzy Sets." *Fuzzy Sets and Systems* 20:87–96.

Bladon, Peter, Richard J. Hall, and W. Andy Wright. 2002. "Situation Assessment Using Graphical Models." Pp. 886–93 in *Proceedings of the 14th International Symposium on Integrated Ferroelectrics ISIF 2002*.

Chalmers, B. A. 1988. "On the Design of Computer-Based C2 Decision Support for a Modern Frigate." in *In Proceedings of the 4th International Command and Control Research and Technology Symposium, 14-16 September, 1998.*

Chang, Yi-zhe, Zhan-wu Li, Ying-xin Kou, Qing-peng Sun, Hai-yan Yang, and Zheng-yan Zhao. 2017. "A New Approach to Weapon-Target Assignment in Cooperative Air Combat." *Mathematical Problems in Engineering* 2017:1–17.

Changwen, Q. and H. You. 2002. "A Method of Threat Assessment Using Multiple Attribute Decision Making." *6th International Conference on Signal Processing, 2002.* 2:1091–95.

Cook, David A. and James M. Skinner. 2005. "How to Perform Credible Verification, Validation, and Accreditation for Modeling and Simulation." Pp. 20–23 in *CrossTalk*.

Dongfeng, Chen, Feng Yu, and Liu Yongxue. 2012. "Threat Assessment for Air Defense Operations Based on Intuitionistic Fuzzy Logic." *Procedia Engineering* 29:3302–6.

Endsley, M. R. 1995. "Toward a Theory of Situation Awareness in Dynamic Systems." *Human Factors* 37:32–64.

Haghighat, M., M. Abdel-Mottaleb, and W. Alhalabi. 2016. "Discriminant Correlation Analysis: Real-Time Feature Level Fusion for Multimodal Biometric Recognition." *IEEE Transactions on Information Forensics and Security* 11(9):1984-1996.

Harris, C. J. 1988. *Application of Artificial Intelligence to Command and Control Systems*. Peter Peregrinus

Ltd, 1988.

Hocaoğlu, M. Fatih. 2013. “Aspect Oriented Programming Perspective in Agent Programming.” *USMOS 2013- National Defense Application and Modeling & Simulation Conference -(in Turkish)* 8(3).

Hocaoglu, Mehmet F. 2017. “Aspect Oriented Programming Perspective in Software Agents and Simulation.” *International Journal of Advancements in Technology* 8(3).

Hocaoğlu, Mehmet F. 2018. “AdSiF: Agent Driven Simulation Framework Paradigm and Ontological View.” *Science of Computer Programming* 167:70–90.

Hocaoglu, Mehmet Fatih. 2005. “AdSiF : Agent Driven Simulation Framework.” in *Hunstville Simulation Conference -HSC2005,* edited by Joseph S. Gauthier. Huntsville, Alabama.

Hocaoğlu, Mehmet Fatih. 2015. *Target Evaluation and Weapon Assesment Optimization*. Technical Report No: FBA-2014-327, İstanbul.

Hu, Xinwu, Pengcheng Luo, Xiaonan Zhang, and Jun Wang. 2018. “Improved Ant Colony Optimization for Weapon-Target Assignment.” *Mathematical Problems in Engineering* 2018:1–14.

Johansson, F. and G. Falkman. 2008. “A Bayesian Network Approach to Threat Evaluation with Application to an Air Defense Scenario.” Pp. 1–7 in *Proceedings of 11th International Conference on Information Fusion*.

Kline, Alexander, Darryl Ahner, and Raymond Hill. 2019. “The Weapon-Target Assignment Problem.” *Computers and Operations Research* 105:226–36.

Kumar, Sushil and Arati M. Dixit. 2012. “Threat Evaluation Modelling for Dynamic Targets Using Fuzzy Logic Approach.” Pp. 143–49 in *International Conference on Computer Science and Engineering*.

Li, You, Yingxin Kou, Zhanwu Li, An Xu, and Yizhe Chang. 2017. “A Modified Pareto Ant Colony Optimization Approach to Solve Biobjective Weapon-Target Assignment Problem.” *International Journal of Aerospace Engineering* 2017:1–14.

Liang, Yawei. 2007. “An Approximate Reasoning Model for Situation and Threat Assessment.” Pp. 246–50 in *Proceedings - Fourth International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2007*. Vol. 4.

Liebhaber, M. J. and B. Feher. 2002. *Air Threat Assessment: Research, Model, and Display Guidelines*.

Mehmet Fatih Hocaoğlu. 2018. “AdSiF: Agent Driven Simulation Framework Paradigm and Ontological View.” *Science of Computer Programming* 167:70–90.

Naeem, H., Masood, A. 2010. “An Optimal Dynamic Threat Evaluation and Weapon Scheduling Technique.” *Knowledge-Based Systems* 23:337–342.

Negoita, C. V. 1985. *Expert Systems and Fuzzy Systems*. The Benjamin/Cummings Publishing Company, Inc.

Oxenham, M. G. 2003. “Enhancing Situation Awareness for Air Defence via Automated Threat Analysis.” in *in Information Fusion, 2003. Proceedings of the Sixth International Conference of. 2003.*

Plamondon, Pierrick and Brahim Chaib-draa. 2004. “A Multiagent Task Associated MDP (MTAMDP) Approach to Resource Allocation.” *Decision Support Systems* (Zhang 2002).

Roux, J. N. and J. H. van Vuuren. 2007. “Threat Evaluation and Weapon Assignment Decision Support: A Review of the State-of-the-Art.” *OriON, J. ORSSA* 23(2):151–187.

S. P. Lloyd, H. and S.Witsenhausen. 1986. “Weapons Allocation Is NP-Complete.” Pp. 1054–1058 in *Proc.*

IEEE Summer Simul. Conf. Reno, NV.

Sikanen, T. 2008. "Solving Weapon Target Assignment Problem with Dynamic Programming." *Independent Res.Projects Appl.Math.* 32.

Tin G. and P. Cutler. 2006. "Accomodating Obstacle Avoidance in the Weapon Allocation Problem for Tactical Air Defense." Pp. 1–8 in *9th International Conference on Information Fusion*.

Xin, B., Peng, Z. 2011. "An Efficient Rule-Based Constructive Heuristic to Solve Dynamic Weapon-Target Assignment Problem." *IEEE Transactions On Systems, Man, And Cybernetics—Part A: Systems And Humans* 41(3).

Zhang, W. 2002. "Modeling and Solving a Resource Allocation Problem with Soft Constraint Techniques, Report: WUCSE-2002-13."

AUTHOR BIOGRAPHY

MEHMET FATIH HOCAOĞLU is an Assistant Professor of Industrial Engineering at Istanbul Medeniyet University. He holds a PhD in Industrial Engineering and Operations Research from Sakarya University. His research interests lie in simulation modeling, reasoning technologies, and languages especially in defense technologies. His email addresses are mfatih.hocaoglu@medeniyet.edu.tr; hocaoglumf@gmail.com; hocaoglu@agenabst.com.