DEVS-BASED SIMULATION FOR SEARCH AND RESCUE MISSIONS INVOLVING MULTIPLE UAVS

Juan B. Bordón-Ruiz
Eva Besada-Portas
José A. López-Orozco
José L. Risco-Martín

Department of Computer Architecture and Automation
Universidad Complutense de Madrid
Calle Prof. José García Santesmases, 9
28040 Madrid, SPAIN
{jubordon,ebesada,jalo,jlrisco}@ucm.es

ABSTRACT

Using Unmanned Aerial Vehicles (UAVs) for searching for targets is a complex problem whose applications range from military operations (e.g. targets localization or border surveillance) to civil missions (e.g. mountain or sea rescue). Simulations of real-world operations play an important role in determining the best search strategies to be used. The high number of elements involved in the problem makes real-time simulations an useful tool for system elements coordination, data handling confluence, etc. Model-Based Systems Engineering (MBSE) principles can also be exploited to define the behavior of the whole system from the behaviors of its elements. This paper presents a Discrete Event System Specification (DEVS) modular and hierarchical architecture for simulating real-world target-search scenarios that enhances the adaptability, reusability and scalability properties of our system. The simulations show the benefits of our architecture, which allows to automatically profile and straightforward evaluate different models to select the best alternative.

Keywords: Model-Based Systems Engineering, Discrete Event Systems, Multi-Agent Systems, Bayesian Search.

1 INTRODUCTION

Searching for targets placed at unknown positions within a given area using Unmanned Aerial Vehicles (UAVs) is a problem with a strong research interest and applicability in a wide variety of missions, such as military target detection, wildlife monitoring, search for survivals, or search and rescue missions (Koopman 1980, Frost and Stone 2001). Depending on the type of mission, the objectives to fulfill are different. For instance, military missions can be carried out in hostile environments where it is crucial to avoid threats to preserve the assets (Moreno et al. 2011), while in search and rescue missions, the target detection time is critical (Lanillos et al. 2014, Pérez-Carabaza et al. 2016). Within Bayes theory, mission objectives can be used to formulate the target search as an optimization problem whose utility functions are used to determine the best UAVs trajectories and sensor poses strategies (Grocholsky et al. 2003, Bourgault et al. 2004).

Although this approach has been applied in numerous search applications, new studies are continuously emerging as the ones reviewed in Raap et al. (2019), Pérez-Carabaza et al. (2019) and Ivić et al. (2020).

Modeling and simulation of the real-world scenario before going to life usually result in multiple benefits such as a better understating of the problem, better performance of the mission objectives, and costs reduction. Simulating these scenarios can be a complex task due to the high number of elements and the mission aspects to be considered. In particular, for each UAV participating in the mission, the simulation has to take into account its flying dynamics, the specific dynamics and likelihoods of its onboard sensors, and the uncertainty of the target location and movements. Besides, depending on the final objectives of the search mission, the simulation process should be able to calculate different utility functions related to the previous elements, such as the probability of detecting the target (Bourgault et al. 2004) or the expected time of detection- to increment the chances of locating it as soon as possible (Lanillos et al. 2014, Pérez-Carabaza et al. 2016). Other mission aspects could also be included in the simulation, like environmental conditions (e.g., winds and ocean currents), terrain topology or mission constraints (e.g., non-flying zones). In the end, this results in a complex problem with a large set of interacting elements to be handled as a system itself.

To manage this complex system, Model-Based Systems Engineering (MBSE, Wymore 2018) and Discrete Event Systems Specification (DEVS, Zeigler et al. 2018) can play an important role during the model definition. On the one hand, MBSE facilitates a better understanding of the system behavior by using an abstraction of reality and by eliminating unnecessary components. This is usually represented by a graphical, mathematical, or physical specification. Having a high-level system specification, clearly separated from the final chosen implementation, facilitates system refinement, decision making and incremental design. On the other hand, DEVS enables system verification, validation and testing through a mathematical formalism and different software solutions. Integrating different models (e.g., probabilistic and deterministic, continuous and discrete, multi-resolution, synchronous and asynchronous) is a straightforward process within DEVS. Moreover, as complexity usually grows when increasing the scenario realism, DEVS can explore parallel and distributed ways of simulation (Mittal and Risco-Martín 2017).

DEVS has been used in multiple applications within the target-search problem scope and/or involving UAVs. To name a few, Holman et al. (2010) introduce a UAV path-planner application based on Cell-DEVS; Moreno et al. (2011) present a DEVS-based model to evaluate UAV trajectories (generated by an external application) and determine if these trajectories avoid dangerous or prohibited zones; and Happe and Berger (2010) introduce a multi-UAV cooperative search path planner focused on coordination strategies and information-sharing policies. Our approach is different to the previous, as it is intended to provide a common framework for simulating and evaluating different types of target-search problems involving UAVs.

This paper continues with the work we have presented in Bordón-Ruiz et al. (2021), which introduced a DEVS-based specification to systematize *the evaluation process* of UAV target-search strategies, and which incorporates improved versions of the models presented by Pérez-Carabaza et al. (2016) and Pérez-Carabaza et al. (2019). In this work, we are focused on the design of the architecture of the evaluation process and provide two valid DEVS specifications for that purpose, which have been designed to be part of a future UAV path-planner and to simulate real-world target-search scenarios. Additionally, the underlying MBSE and DEVS support of both specifications guarantees an incremental, scalable, reusable, and adaptable design that can be used to integrate different types of models (for the UAVs, sensors and targets) and multiple mission objectives. Finally, this paper also uses a simulation profiling method to straightforwardly evaluate the performance of the different models, demonstrating the versatility and flexibility of our architectures.

2 PROBLEM FORMULATION

This section introduces the target search problem formulation under the Bayesian perspective. This information is needed to 1) understand the structure of our system specification, and 2) verify and validate the DEVS-models and system behavior.

In the target-search problem of this paper, there is a group of U UAVS looking for a single target within a search area Ω (which is a rectangular region, parallel for simplicity to the (x, y) axes, and defined by its south-east corner and lateral dimensions). Each UAV can be equipped with S sensors that make observations to find the target. The target, the UAVs, and their sensors are the main elements of the problem and directly represented as DEVS models. The mission objectives are also modeled as utility functions and evaluated during the simulation. In the following, we briefly describe these elements. For further details, the reader can refer to Pérez-Carabaza et al. (2019) and Bordón-Ruiz et al. (2021).

2.1 UAV Definition and Modeling

Each UAV is modeled with a deterministic motion function, whose inputs are (according to Eq. (1) of Table 1) the previous UAV state $s_u^{t-T_u}$, possible control actions $a_u^{t-T_u}$, environmental conditions ε^{t-T_u} and time lag T_u. For the UAV motion model used in this paper, s_u^t includes the u-th UAV (x, y) position, its height, its heading, its speed, and its fuel consumption. Besides, ${a_u}^{t-T_u}$ comprises the desired UAV heading, height and speed; is definable with absolute, incremental or rate values; and can be updated periodically or at user-defined time moments. Additionally, ε^{t-T_u} considers the wind speed and angle. Moreover, the model is based on the set of differential equations presented in (Pérez-Carabaza et al. 2019), which only consider the UAV behavior in 3D and in the yaw body-axis. Finally, the model is simulated within DEVS, using the 4th order Runge-Kutta integration method and outputting s_u^t periodically (every T_u seconds).

2.2 Target Definition and Modeling

Due to the target position uncertainty, Ω is discretized into a grid G (of $N_G = N_x$ x N_y rectangular cells each of size w_x x w_y) to represent the target probability distribution as a probability map or belief $b(c^t)$ over each cell $c^t \in G$. The initial target belief $b(c^0)$ is known. For computation purposes, we introduce $p(c^t)$ to store, for each cell in G and time step t, the "remaining or unobserved probability". At any time step, $b(c^t)$ can be obtained as $p(c^t)/\sum_{g \in G} p(c^t = g)$. Besides, we initialize $p(c^0) = b(c^0)$.

To model the behavior of a moving target, we use a Markovian target motion probability $p(c^t|c^{t-T_\tau})$, which states how probable is that the target at cell c^{t-T_τ} at time step $t - T_\tau$ arrives at cell c^t at time t. Moreover, we apply the prediction step of the Recursive Bayesian Filter (RBF, Bourgault et al. 2004) to redistribute the unobserved probability over the map, using Eq. (2) of Table 1. From DEVS perspective, this prediction step is executed periodically every T_τ seconds.

In addition, the unobserved probability $p(c^t)$ shall also be updated with the information provided by the sensors onboard the UAVs. This operation, implemented as Eq. (3) of Table 1 and performed whenever a new observation is available, is similar to the assimilation step of the RBF and uses $p(\overline{D}|c^t, s_{u,k}^{t_m})$, which is the complementary of the likelihood sensor model that will be explained in the following section.

2.3 Sensor Definition and Modeling

Onboard detection sensors are used to make observations within Ω to find the target. To model the sensor uncertainty, we have to define, for each sensor k in UAV u, its likelihood or its complementary function (i.e.

Table 1: Model equations.

Element	Type	Expression	Eq. #
UAV	Deterministic Motion	$s_u^t = f(s_u^{t-T_u}, a_u^{t-T_u}, \varepsilon^{t-T_u}, T_u)$	(1)
Target	Probabilistic Motion	$p(c^t) \leftarrow \sum_{c^{t-T} \in G} p(c^t \vert c^{t-T_\tau}) p(c^{t-T_\tau})$	(2)
Target & Sensor	Probabilistic Observation	$p(c^t) \leftarrow p(\overline{D} \vert c^t, s_{u,k}^{t_m}) \cdot p(c^t)$	(3)
Sensor	Deterministic Motion	$s_{u,k}^t = g(s_u^t, s_{u,k}^{t-T_{u,k}}, a_{u,k}^{t-T_{u,k}}, \varepsilon^{t-T_{u,k}}, T_{u,k})$	(4)
Evaluation	Objective	$P_d(t) = 1 - \sum_{c^t \in G} p(c^t)$	(5)
		$ET(t) = \sum_{l=1:1:t/T_\tau} (1 - P_d(l \cdot T_\tau)) T_\tau$	(6)
	Constraint	$NFZs = \sum_{t=1}^{T} \sum_{u=1}^{U} WithinNFZ(s_u^t)$	(7)
		$COL = \sum_{t=1}^{T} \sum_{u=1}^{U} \sum_{l=k+1}^{U} Collision(s_u^t, s_l^t, dCOL)$	(8)

the probability of not detecting the target $p(\overline{D}|c^t, s_{u,k}^{t_m})$ placed at cell c^t from the location and pose $s_{u,k}^{t_m}$ of that sensor at timestamp $t_m \in [t, t+T_\tau]$)).

The sensor likelihoods used in this paper model the behavior of a downward-looking primary radar and of high resolution rotating cameras. Although both models are respectively based on the likelihoods used in Pérez-Carabaza et al. (2016) and Pérez-Carabaza et al. (2019), they have been improved for grids of different resolutions by averaging the likelihoods over a set of N_c equally spaced points within each cell c^t.

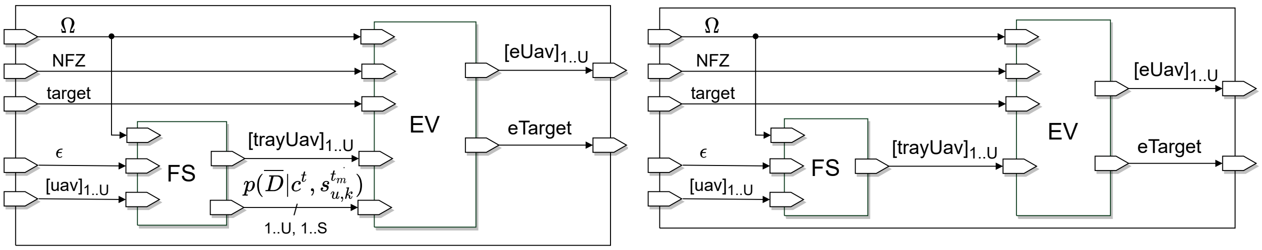
Finally, for moving sensors, DEVS must also model the sensor dynamics. These are conceptually similar to the UAV motion models, but, as Eq. (4) of Table 1 shows, require also the sensor state $s_{u,k}^*$ and its own input signal $a_{u,k}^*$. In particular, in this paper we use the model of a gimballed camera whose input actions are the azimuth and elevation, and whose differential equations are presented in Pérez-Carabaza et al. (2019). We have incorporated them into a DEVS model that cyclically integrates them using a 4th order Runge-Kutta.

2.4 Mission Objectives and Constraints Definition

The mission objectives and constraints represent the utility functions. They are used to determine the grade of fulfillment of the proposed UAV trajectories and sensor poses for a given scenario. The functions used in this paper are the probability of target detection (P_d, Eq. (5) of Table 1), the Expected Time of detection (ET, Eq. (6)), the number of Non-Flying Zones overflights (NFZ, Eq. (7)) and the number of UAV collisions (COL, Eq. (8)).

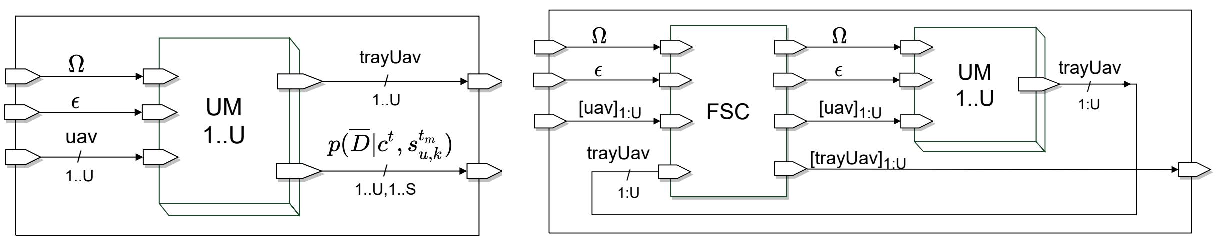
3 MODEL SPECIFICATIONS

This section presents our DEVS-based specification for simulating real-world target-search scenarios. Following MBSE principles, two alternative architectures have been designed in order to analyze the trade-off between them. The first specification (named SPEC1 here after and already presented in Bordón-Ruiz et al. 2021) performs the evaluation incrementally as in the real-world: while the UAVs are flying over the search area, the sensors are making the observations, and the target probabilities and objective functions are consequently updated. The second alternative (SPEC2) is based on how many target-search path planners (e.g. Pérez-Carabaza et al. 2016 and Pérez-Carabaza et al. 2019) decide which are the best solutions: at each evaluation step they initially simulate the UAVs trajectories for a given set of control actions, and afterwards they simulate the target behavior and evaluate the resulting UAVs trajectories and sensor poses. The following sections describe both specifications from a top-down perspective.



(a) SIM model for SPEC1 (b) SIM model for SPEC2

Figure 1: Simulator SIM coupled models.



(a) FS model for SPEC1 (b) FS model for SPEC2

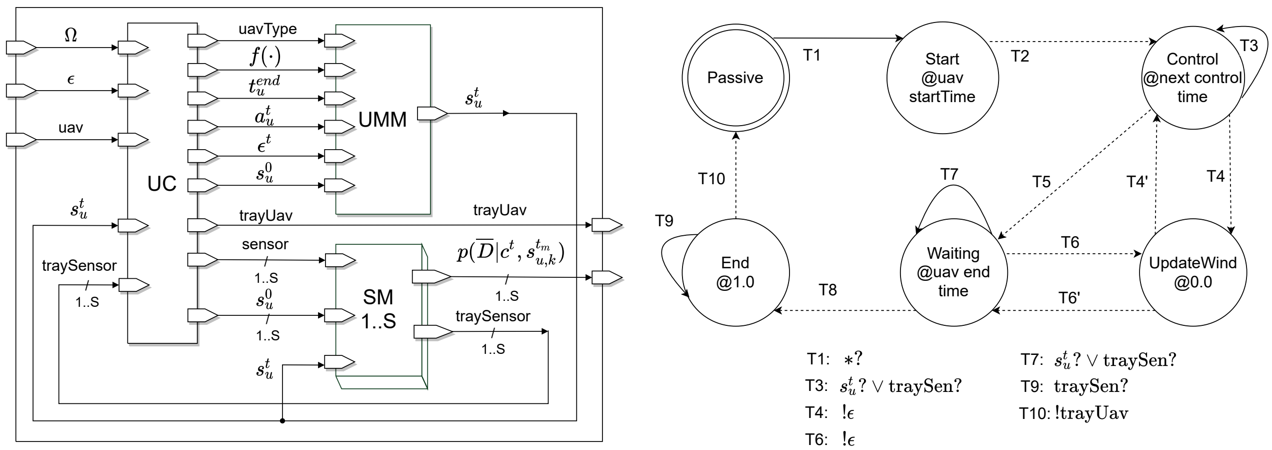
Figure 2: Flight Simulator FS coupled models.

3.1 Simulator (SIM)

The Simulator Models (SIM) represent the high-level abstraction of the target-search problem. Their input ports are: the search area Ω, the NFZs defined as the lists of cells of $c_l^{NFZ} \in G$, the encapsulated definition of the `target`, the information of the environment ε, and the encapsulated definition of all the UAVs involved in the mission `[uav]`$_{1:U}$. Their outputs are the encapsulated definition of the UAVs evaluations `[eUav]`$_{1:U}$ and the encapsulated definition of the target evaluation `eTarget`. The structure of this model in both specifications is composed, as Fig.1 shows, of the Flight Simulator (FS) and the Evaluator (EV) coupled models. Their internal and external couplings are almost the same. The only difference is that, due to each specification philosophy, the FS model for SPEC1 outputs the sensors likelihoods $P(\overline{D}|c^t, s_u, k^{t^m})$, while the FS model for SPEC2 does not.

3.2 Flight Simulator (FS)

FS structures, respectively represented in Fig.2, are slightly different for both specifications. Although both aggregate a group U of UAV coupled Models (UMs), SPEC2 needs an additional atomic model named Flight Simulator Core (FSC). This is a consequence of how each alternative handles the simulation and the scenario times. In SPEC1, the simulation time should be equal to the scenario time to synchronize FS and EV models, as both are running in parallel (as in the real-world). In SPEC2, the simulated UAVs trajectories `trayUAV` are generated before the rest of the evaluation process starts. Hence, the FSC model of SPEC2 is needed for receiving the whole set of `trayUAV` before sending them as a unique set of data.



(a) UM model for SPEC1 (b) UC states for SPEC1

Figure 3: UM model and UC states diagrams.

3.2.1 UAV Model (UM)

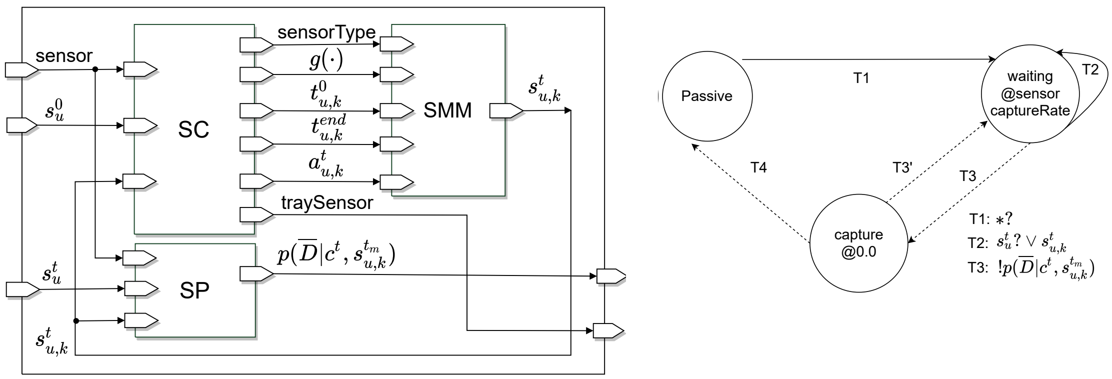
As stated above, each instance of UM coupled model represents a different UAV. Besides the encapsulated UAV definition `uav`, the search area Ω and the environment information ε are the other inputs of the model. Its structure for SPEC1 is shown in Fig. 3(a), and defined in a group of S Sensor coupled Models (SM, which represent the onboard sensors of the UAV), and in the UAV Control (UC) and UAV Motion (UMM) atomic models. The structure for SPEC2 is almost the same. However its SM models do not output the sensor likelihood (since this functionality is carried out in the EV model for SPEC2). Consequently, the SM models of SPEC2 do not need either the UAV start time t_u^0 or the UAV state s_u^t coupling to generate (only) the corresponding `traySensor`.

Within UM, the main functionality of UC is to handle the lists of UAV setpoints/control actions. The UC model behavior for SPEC1 is described in the Fig. 3(b). It differs from the UC behavior for SPEC2 in the value of the DEVS σ parameter that triggers the internal transition T2. In SPEC1, T2 is triggered setting σ to the UAV start time t_u^0. This guarantees that the simulation time matches the scenario time, and therefore that the synchronization is achieved. On the other hand, in SPEC2, there is no benefit in doing this, so σ is directly set to the difference of time between the UAV start time t_u^0 and the scenario time of the first setpoint a_u^t in the control list, to let all UM model run simultaneous. Besides, UC is also in charge of receiving the new states of s_u^t from UMM (T3, T7), wrapping them into `trayUAV`, updating the environmental information if needed (T4/T4'), and receiving the sensor simulated trajectories `traySen` (T3, T7). Finally, the result is the simulated `trayUAV` outputted via DEVS λ function before T10 happens.

The functionality of UMM consists in executing the UAV motion model $f(\cdot)$ by programming a δ_{int} setting σ to the UAV motion rate. Therefore, a new state s_u^t is periodically sent to the UC model via the λ function, a behavior that is repeated until this model time exceeds t_u^{end}.

3.2.2 Sensor Model (SM)

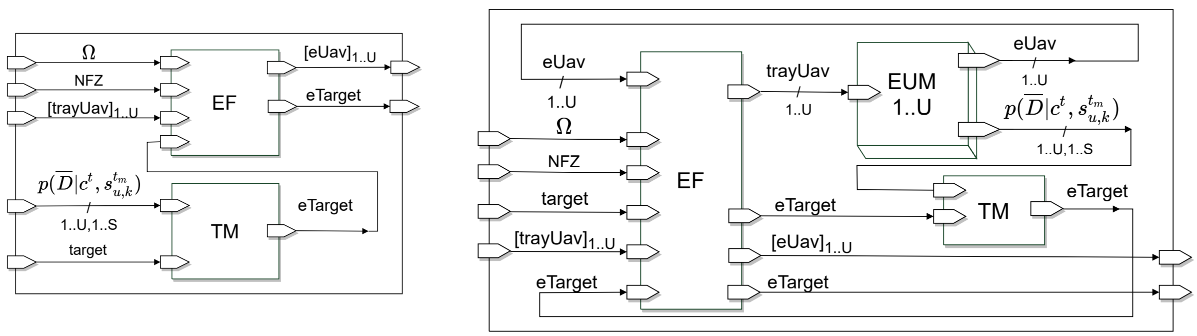
SM represents a sensor k mounted on the UAV u. Its structure, for SPEC1, is depicted in Fig. 4(a). Again, there are some differences with the structure for SPEC2: SM of SPEC1 aggregates the Sensor Control (SC), Sensor Motion (SMM) and Sensor Payload (SP) atomic models; while SM of SPEC2 does not aggregate



(a) SM model for SPEC1

(b) SP states diagram for SPEC1

Figure 4: SM model and SP states diagrams for SPEC1.



(a) EV model for SPEC1

(b) EV model for SPEC2

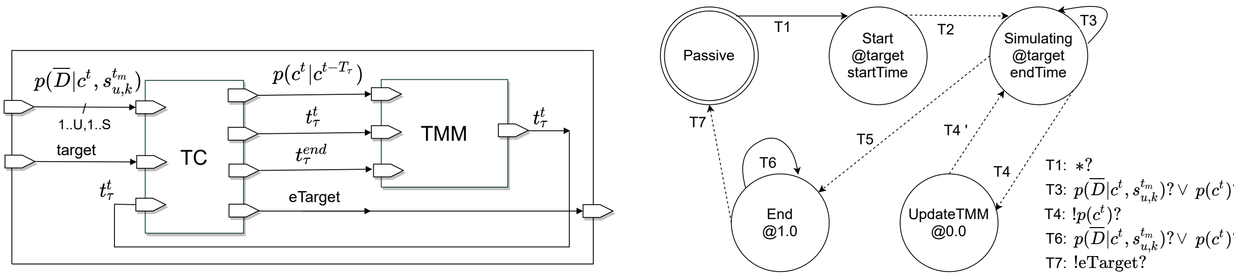
Figure 5: Evaluator (EV) coupled models.

SP (because the likelihoods are calculated within the EV module) and its associated couplings do not apply. Eventually, if sensor k is static, the SM model is simplified to a unique SP atomic model in SPEC1 or just eliminated in SPEC2, as these types of sensors do not have a motion model. This consideration also affects SP model, activating/deactivating the coupling for receiving $s_{u,k}^t$ from SMM for dynamic/static sensors.

The behaviors of SC and SMM are similar to the behaviors of UC and UMM. They differ in the motion model they integrate (i.e., the sensor $g(\cdot)$ vs. the UAV $f(\cdot)$), and in the input/output signals (related to sensors vs. UAVs). SP behavior, shown in Fig. 4(b) for SPEC1, consists in outputting the sensor likelihood.

3.3 Evaluator (EV)

The Evaluator (EV) coupled model is linked to the target-search evaluation process. Conceptually, the main difference between both alternatives resides in where the sensor likelihoods are calculated: they are inputs of EV for SPEC1 and produced inside EV for SPEC2. This results in two different model structures, represented in Fig. 5. In more detail, besides the common Target coupled Model (TM) and their respective Evaluation Function models (EF), SPEC2 needs to aggregate a group of U Evaluator UAV coupled Models (EUM) to simulate the sensor observations resulting from the generated `trayUAV` and `traySensor`.



(a) TM model for SPEC1 (b) TC states diagram for SPEC1

Figure 6: TM model and TC states diagrams for SPEC1.

3.3.1 Target Model (TM)

This coupled model represents the target-related processes, including the operations that predict (Eq. (2)) and update (Eq. (3)) the unobserved probability $p(c^t)$. TM structure can be adapted for static and dynamic targets, and Fig. 6(a) depicts the one with more complex dynamics. The structure and behavior of the aggregated atomic models do not change for both specifications, as they carry out the same processes.

The Target Control (TC) atomic model contained in TM model (see Fig. 6(b)) addresses the reception of the likelihoods $p(\overline{D}|c^t, s_{u,k}^{t_m})$ via external transitions T3 and T6, and updates the $p(c^t)$ with Eq. (3). For dynamic targets, additional functionalities are to sent the new $p(c^t)$ to the Target Motion Model (TMM) by programming internal transition T4 and T4', and processing prediction updates Eq. (2) received via T3 and T6. These processes are performed until the target end time t_τ^{end}, when `eTarget` is finally sent.

The Target Motion atomic Model (TMM) is in charge of handling the target movement predictions operations. These predictions are operated periodically (at target motion rate) by applying $p(c^t|c^{t-T_\tau})$ to obtain a new $p(c^t)$ that is output to TC. This is repeated until the next prediction time is bigger than t_τ^{end}.

3.3.2 Evaluator Function (EF)

The Evaluator Function (EF) atomic model manages the calculation of the utility functions (i.e. the mission objectives and constraints). In addition, in SPEC2 EV works as a data concentrator to receive the whole set of `trayUAV` before starting the evaluation process. This is necessary as in the evaluation process all the collaborating models must be synchronized, matching the simulation time to the scenario time. In SPEC1, as the synchronization is done at the SIM model level, there is no need to use EF as a concentrator. EF behavior is elementary, acting as a passive model waiting for the input data to calculate the mission objectives and send the final evaluation data.

3.3.3 Evaluator UAV Model (EUM)

This coupled model only applies to SPEC2. It is composed by one Evaluator UAV Control (EAC) atomic model and a group of S Evaluator Sensor Payload (ESP) atomic models. Its functionality consists in going through `trayUAV` and `traySensor` (for dynamic sensors) to produce the sensor likelihoods $p(\overline{D}|c^t, s_{u,k}^{t_m})$.

Table 2: Characterization of the two UAVs.

Decription	UAV1	UAV2
Flying height range (ft)	$[500, 23000]$	$[500, 13100]$
Flying speed range (kts)	$[60, 110]$	$[115, 190]$
Operating range (nm)	135	81
Flying autonomy (hours)	20	7
Payload	Radar	Camera(s)

4 RESULTS

SPEC1 and SPEC2, which represent a high-level abstraction of the target-search problem, have been designed modular and hierarchically to make them reusable, scalable, and adaptable to further requirements. Their independence provides multiple benefits for users unfamiliar or expert in coding, as future developers might extend/modify them without understanding the low-level details of their implementation.

Nowadays, there are multiple DEVS M&S engines. Among them, xDEVS (Risco-Martín et al. 2017) is a programmer-friendly and well-defined Application Programming Interface (API) to implement DEVS applications. It is also independent of any vendor, product, or technology (Mittal and Risco-Martín 2017). Additionally, the xDEVS profiling tool lets users analyze and detect the most resource-consuming models.

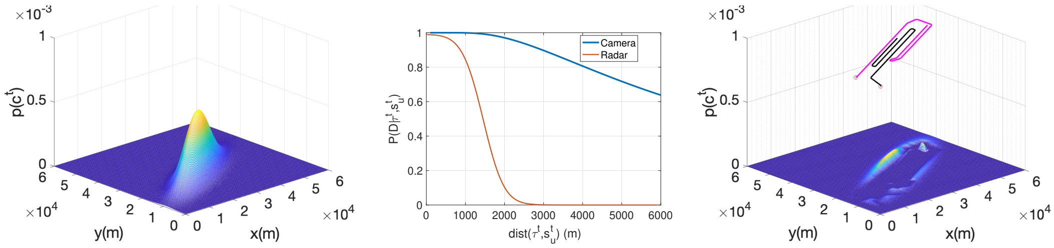
The following section introduces a real-word-inspired scenario to validate both specifications and conduct a profile analysis to compare the computational efficiency of both specifications under the same scenario and search strategy. The scenario is defined in a JSON file that includes all the mission aspects (e.g., UAVs definition, search area location, dimensions, target initial probability map, etc.). Additional configuration files are used to load specific parameters needed to define the behaviors of some models. Therefore, the scenario setup is a straightforward process because once the behavior and ports of all the models are defined, their connection and configuration can be performed through these files. This is facilitated by the integral separation between the modeling and simulation layers in the proposed MBSE methodology.

4.1 Simulation Scenario

The selected real-world scenario is based on a maritime Search And Rescue (SAR) mission, where it is necessary to find a group of shipwreck survivors in a drifting rubber boat (of 5 m). The search area Ω is defined as a square of 32x32 nm^2, discretized into a grid of 200x200 cells. The initial target probability map $b(\tau^0)$, shown in Fig 7(a), is defined based on the ship course before the shipwreck and its estimated position by the call time. The boat is drifting due to the wind and sea tides. We model this behavior with a $p(c^t|c^{t-T\tau})$ that is used in Eq. (2) every $T_\tau = 150$s and that makes the target advance at most 2 m/s northeast.

Available resources to perform the SAR mission are two UAVs inspired in the tactical Spanish Searcher MK-III and in the INTA SIVA, respectively, whose main characteristics are summarized in Table 2. The motion model for both UAVs is detailed in Pérez-Carabaza et al. (2019) and integrated every 1s. The input control signals to the UAVs' motion models are defined with variable time duration in a file that stores a list of absolute heading actions for UAV1 and a list of absolute headings and rate heights for UAV2.

Regarding the sensors, UAV1 is equipped with a continuous wave radar that takes measurements every 4s, while UAV2 has two cameras. The first one is a high-resolution video camera, whose Line of Sight (LoS) points in the same direction as the UAV heading and whose elevation is fixed at 30 deg (from the aircraft longitudinal axis). The horizontal and vertical Field of View (FoV) angles of its lens is set to 40 deg. Finally, its likelihood function is executed every 1s.The second camera is mounted on a gimbal, with a maximum slew rate of 5 deg/s, an azimuth range of 360 deg, and an elevation range from 0 to 90 deg. Its FoVs are also



(a) Initial belief $b(c^0)$ (b) Sensor likelihood (c) Final $p(c^t)$ at $t = 50$ minutes

Figure 7: Simulation Scenario.

fixed to 40 degrees. Its motion model and likelihood are respectively executed every 2s and 4s. Finally, all likelihoods are shown at Fig. 7(b) as a function of the distance between sensors and cells.

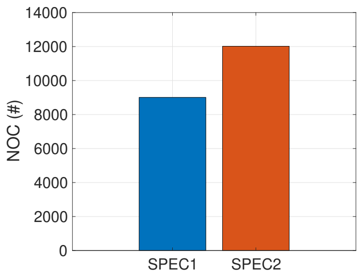
4.2 Analysis

For the simulation (run in an Intel 4-Cores i7 at 2,5 GHz with a RAM of 16 GB 1600 MHz DDR3), UAV1 arrives at Ω at t_1^{start} = 0s from the southwest corner, with a cruise speed of 77.7 kt and a height of 3000 ft. To take advantage of the radar, UAV1 initially flies straight forward to the shipwreck's reported position. Afterward, it follows the rubber boat's possible trace with a typical lawn-mower until $t_{mission}^{end}$ = 3000s. Besides, UAV2 arrives at Ω at t_2^{start} = 0s, at a position slewed to the north to perform a wider lawn-mower pattern that increases the target detection probability. Its cruise speed is 87.4 kt and its height starts at 10000 ft, and descends, with a height rate of 5 m/s, from $t_2^t = 10s$ to $t_2^t = 420s$. This increases the likelihood of both cameras but reduces their footprints. The UAV trajectories (without considering their height) and final $p(c^t)$ are displayed at Fig. 7(c). From an evaluation point of view, the simulated UAV and sensor behavior are adequate for this scenario, as they reach a probability of detection of 91% (as $P_d(t_{mission}^{end}) = 0.91$) and allow to detect the target, in average, at the 37% of the mission time (since ET=1113s and $t_{mission}^{end}$=3000s).

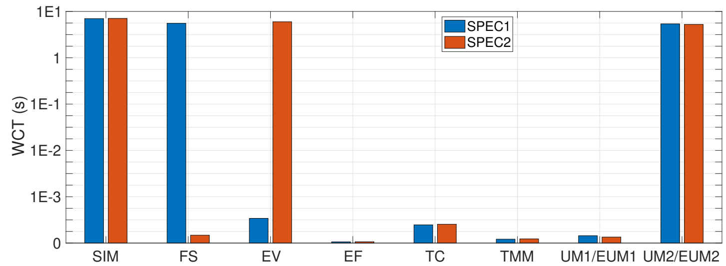
For the profile analysis, we run each specification 10 times over the selected scenario. Next, we obtain for any of the runs and from xDEVS profiling tool, the total Number-Of-Calls (NOC) of the models for the time advance *ta* function, which gives an idea about the granularity of the simulation loops. As Fig. 8(a) shows, SPEC2 NOC is 23% higher due to the higher number of DEVS modules in SPEC2, which overloads the underlying xDEVS coordinator. The Wall-Clock Time (WCT) of the top-level SIM model can also be measured for each run and averaged (since WCT can change slightly at each run). In this case, SPEC2 slightly increases SIM WCT in 0.21 ± 0.2s. Moreover, The Wilcoxon test over the WCT of SIM for SPEC1 and SPEC2 shows that they are statistically different. In more detail, Fig. 8(b) shows the WCT of several models for both specifications, representing, for each model and specification, the accumulated times of the transition and output functions. It is worth noting the non-linear scaling of the y-axis, and that the last two pairs of bars compare models that are equivalent, from the functional point of view, in both specifications. The graphic shows how the computational resources are swapped between the EV and FS modules for each specification. This happens due to the overload associated with the sensor payload models and where this functionality is carried out in SPEC1 (FS) and SPEC2 (EV). Finally, both specs obtain the same P_d and ET.

5 CONCLUSIONS

This paper presents two DEVS specifications for simulating and evaluating UAV strategies in search and rescue missions. Both specifications provide a solid framework to integrate different models (for UAVs, sensors, and targets) and mission objectives. We can easily do this by adjusting them to the interfaces (ports and couplings) provided by the DEVS formalism. Additionally, our approach guarantees an incremental,



(a) NOCs



(b) WCT

Figure 8: Profile analysis.

scalable, reusable, and adaptable design validated with the simulations conducted against real-world inspired scenarios. Finally, the profile analysis shows that, at present, SPEC1 is more straightforward and computationally efficient than SPEC2.

As future work, we plan to integrate other types of UAV trajectories (e.g., Dubin curves or splines) and target probability models (e.g., particle filters). We also want to include SPEC1 in an optimizer/planner that will automatically obtain the best search strategies for the UAVs. Besides, as some planners (as the ones presented in Pérez-Carabaza et al. 2016 and Pérez-Carabaza et al. 2019) require many evaluations and computational time, the profile analysis can be useful and set the basis to decide how to distribute better the existing modules in a parallelized simulation (Mittal and Risco-Martín 2017).

ACKNOWLEDGMENTS

This work is supported by the Spanish Ministry of Science and Innovation (MICINN) under AMPBAS (RTI2018-098962- B-C21), and by the Madrid Regional Government under CABAHLA-CM (S2018/TCS-4423).

REFERENCES

Bordón-Ruiz, J. B., E. Besada-Portas, J. L. Risco-Martín, and J. A. López-Orozco. 2021. "DEVS-Based Evaluation of UAVs-Based Target-Search Strategies in Realistically-Modeled Missions". In *ACM SIGSIM Conference on Principles of Advanced Discrete Simulation (PADS)*. Suffolk, Virginia, USA.

Bourgault, F., T. Furukawa, and H. F. Durrant-Whyte. 2004. "Decentralized Bayesian Negotiation for Cooperative Search". In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Sendai, Japan.

Frost, J., and L. Stone. 2001. "Review of Search Theory: Advances and Applications to Search and Rescue Decision Support". Technical Report CG-D-15-01, US Coast Guard Research and Development Center, Groton, CT, USA.

Grocholsky, B., A. Makarenko, and H. Durrant-Whyte. 2003. "Information-Theoretic Coordinated Control of Multiple Sensor Platforms". In *IEEE International Conference on Robotics and Automation (ICRA)*. Taipei, Taiwan.

Happe, J., and J. Berger. 2010. "CoUAV: A Multi-UAV Cooperative Search Path Planning Simulation Environment". In *Summer Computer Simulation Conference*, pp. 86–93. Ottawa, ON, Canada.

Holman, K., J. Kuzub, and G. Wainer. 2010. "UAV Search Strategies Using Cell-DEVS". In *Annual Simulation Symposium*, pp. 192–199. Orlando, Florida, USA.

Ivić, S., B. Crnković, H. Arbabi, S. Loire, P. Clary, and I. Mezić. 2020. "Search Strategy in a Complex and Dynamic Environment: the MH370 Case". *Sci Rep* vol. 10.

Koopman, B. 1980. *Search and Screening: General Principles with Historical Applications*. Oxford, UK, Pergamon Press.

Lanillos, P., E. Besada-Portas, J. A. Lopez-Orozco, and J. M. De la Cruz. 2014. "Minimum Time Search in Uncertain Dynamic Domains with Complex Sensorial Platforms". *Sensors* vol. 14 (8), pp. 14131–14179.

Mittal, S., and J. L. Risco-Martín. 2017. "DEVSML 3.0 Stack: Rapid Deployment of DEVS Farm in Distributed Cloud Environment Using Microservices and Containers". In *Proceedings of the 2017 Spring Simulation Multiconference*. Virginia Beach, VA, USA.

Moreno, A., L. la Torre, J. L. Risco-Martín, E. Besada-Portas, and J. Aranda. 2011. "DEVS-Based Validation of UAV Path Planning in Hostile Environments". In *The International Defense and Homeland Security Simulation Workshop*, pp. 135–140. Rome, Italy.

Pérez-Carabaza, S., E. Besada-Portas, J. A. López-Orozco, and J. M. de la Cruz. 2016. "A Real World Multi-UAV Evolutionary Planner for Minimum Time Target Detection". In *The Genetic and Evolutionary Computation Conference*, pp. 981–988. Denver, Colorado, USA.

Pérez-Carabaza, S., E. Besada-Portas, J. A. López-Orozco, and G. Pajares. 2019. "Minimum Time Search in Real-World Scenarios Using Multiple UAVs with Onboard Orientable Cameras". *Journal of Sensors* vol. 2019, pp. 22.

Raap, M., M. Preuß, and S. Meyer-Nieberg. 2019. "Moving Target Search Optimization — A Literature Review". *Computers & Operations Research* vol. 105, pp. 132 – 140.

Risco-Martín, J. L., S. Mittal, J. C. Fabero, M. Zapater, and R. Hermida. 2017. "Reconsidering the Performance of DEVS Modeling and Simulation Environments Using the DEVStone Benchmark". *SIMULATION* vol. 93 (6), pp. 459–476.

Wymore, A. W. 2018. *Model-Based Systems Engineering*, Volume 3. CRC press.

Zeigler, B. P., A. Muzy, and E. Kofman. 2018. *Theory of Modeling and Simulation: Discrete Event & Iterative System Computational Foundations*. Academic Press.

AUTHOR BIOGRAPHIES

JUAN B. BORDÓN RUIZ is a Ph.D. student from Universidad Complutense de Madrid (UCM), currently working for a private company of the aerospace sector. His research interest include modeling and simulation, control of autonomous vehicles and optimization of target-search problems. His email address is jubordon@ucm.es.

EVA BESADA-PORTAS is an Associate Professor of Systems Engineering and Automation at UCM. She holds a PhD in Computer Systems from the same university. Her research interests include uncertainty modeling and simulation, optimal control and planning of unmanned vehicles, and optimization of complex systems. Her email address is ebesada@ucm.es.

JOSÉ A. LOPEZ-OROZCO is a Full Professor in the University Complutense of Madrid. He holds a Ph.D. in Physics from the same University. His research interests include multisensor data fusion, control and planning of unmanned vehicles, and robotics. His email address is jalo@ucm.es.

JOSÉ L. RISCO-MARTÍN received his PhD from UCM, and currently is Full Professor in the Department of Computer Architecture and Automation at the same university. His research interests include computer aided design, and modeling, simulation and optimization of complex systems. His email address is jlrisco@ucm.es.