Doctrine-Based Multi-Resolution Conversion for Distributed Agent-Based Simulations

Raul Ceretta Nunes[1][a], Guilherme Miollo[1], Edison Pignaton de Freitas[2][b] and Luis Alvaro Lima Silva[1][c]

[1] *Graduate Program in Computer Science, Federal University of Santa Maria, Santa Maria, Brazil*

[2] *Institute of Informatics, Federal University of Rio Grande do Sul, Porto Alegre, Brazil*

Keywords: Distributed Simulation, Simulation Integration, Multi-Resolution Simulation, Agent-Based Simulation.

Abstract: The treatment of different levels of simulation detail is a relevant component for integrating simulators in various application domains. These levels vary in federated multi-resolution simulations, from low-information (low-resolution) to high-information (high-resolution). One of the challenges in this integration is the representation and conversion of simulation data exchanged between the simulators. This work explores the use of doctrine-based rules in the conversion to ensure correct simulation integration. These rules contain information on how the multi-resolution conversion handlers should operate. To avoid abrupt changes from one doctrine rule to another, this work also extends a doctrine description language to capture information for the smooth transition between these rules. Experimental results demonstrate that it is possible to achieve simulations that flexibly deal with the required dynamism of a multi-resolution simulation environment.

1 INTRODUCTION

The interoperability between different resolution simulators is a fundamental issue for distributed simulation applications. Such simulation systems' interoperability is usually based on the exploration of the *High-Level Architecture* (HLA) (IEEE, 2010), where the integrated simulators are built to meet different training objectives (Falcone et al., 2017). There are simulators for operational training with high information granularity models. For example, in (Zhou et al., 2019), the goal is to allow trainees to perform detailed operational activities in virtual environments. In contrast, simulation systems can focus on the execution of more abstract simulation-based training activities with low information granularity levels, such as the ones required for strategic and tactical training. In (Pozzer et al., 2022), for example, operational aspects of the simulated problem are abstracted because they have limited relevance for tactical and strategic training. The differences in these simulation goals most often observed when simulation systems with different resolution levels are used in the same integrated simulation setup, each implemented according to distinct information levels of detail. The problem with this approach is that simulated exercises involving simulators with different resolution levels will likely require customized integration and synchronization methods, and the standard HLA architecture only superficially/partially answers this problem.

The works presented in (Kong and Xing, 2013) and (Tolk, 2012) investigate multi-resolution conversion solutions where the proposals are independent of the communication architecture in the distributed simulation. From (Kong and Xing, 2013), a multi-resolution modeling can be built from a multi-resolution entity (MRE) method that maintains running information of different resolution models. The challenge of MRE is to maintain consistency of the interaction with different resolution models. The work in (Tolk, 2012) details a solution for multi-resolution conversion where agent-oriented aggregation/disaggregation methods are investigated. The authors in (Paul et al., 2017) also use aggregation/disaggregation methods but propose exploring user-written doctrine rules to detail the required information for multi-resolution conversions. These works are based on multi-resolution conversion handlers, which deal with the required conversion flexibility for distributed simulation architectures. However, the

[a] https://orcid.org/0000-0003-3228-4071
[b] https://orcid.org/0000-0003-4655-8889
[c] https://orcid.org/0000-0002-6025-5270

Nunes, R., Miollo, G., Pignaton de Freitas, E. and Silva, L.
Doctrine-Based Multi-Resolution Conversion for Distributed Agent-Based Simulations.
DOI: 10.5220/0012124700003546
In *Proceedings of the 13th International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH 2023)*, pages 377-384
ISBN: 978-989-758-668-2; ISSN: 2184-2841

literature still weakly addresses the dynamic multi-resolution conversions and the consequent multi-resolution mappings between low and high-resolution multi-agent simulation systems (Macal, 2016).

The main contribution of this work is to propose a solution to multi-resolution conversion problems. This solution is based on doctrine rules that specify how to make customized agent aggregation/disaggregation conversions between low and high-resolution agent-based simulators. These converters receive low-granularity information from a low-resolution level simulator (e.g., containing aggregate units such as battalions) and convert them to a high-granularity simulator (e.g., convert an aggregate unit into its associated physical entities). This work also proposes the exploration of dynamic multi-resolution conversion handlers, providing a solution to smooth the simulation changes between discrete states, allowing these changes to be realistically developed in simulators with different resolution levels.

2 A DOCTRINE-BASED MULTI-RESOLUTION CONVERSION SOLUTION

The work in (Pozzer et al., 2022) presents a concrete instance of integrated multi-resolution simulation. The SIS-ASTROS simulator develops a low-resolution simulation for recognition, choice, and occupation of tactical battery positions to promote tactical training. These virtual tactical activities are simulated in a high level of detail in this simulation system. However, the SIS-ASTROS simulator is a component of an *Integrated Simulation System*, where constructive, virtual tactical, and virtual technical (operational) simulators are interconnected to build a virtual real-world setting for the augmented training of personnel with distinct military echelons.

A common simulation scenario in the proposed SIS-ASTROS Integrated Simulation System occurs when the constructive simulator, for higher military echelons, controls agents inserted in the virtual tactical exercises. The multi-resolution conversion issues appear because the simulated agents (or aggregates) do not need to perform detailed tactical behaviors in the constructive simulations. When converting the simulation information from the constructive simulator to the virtual tactical, the constructive view only provides broad and discretized information to control the agents mapped in the other simulation system. As the constructive simulator keeps the property of these mapped agents, it cannot deal with the different agent behaviors that occur when these entities follow alternative simulation doctrines. These doctrines guide the execution of the required agent behaviors in the various tactical positions of the simulated exercises. However, the agent's doctrine behavior changes throughout the simulations are important elements for the fluency and realism of the more detailed virtual tactical simulations.

The solution for distributed multi-resolution simulation proposed in this article considers that at least two simulators are integrated, low and high-resolution ones. Figure 1 shows that unit X is aggregated in the low-resolution simulator. The division of unit X into its individual elements is performed according to the specified disaggregation rule. Unit X is detailed as circularly dispersed Y entities on the high-resolution simulator's terrain. Figure 1 also shows that the high-resolution simulator represents terrain elements in higher levels of detail (e.g., various vegetation types). These elements are taken into account at the time the disaggregation occurs.

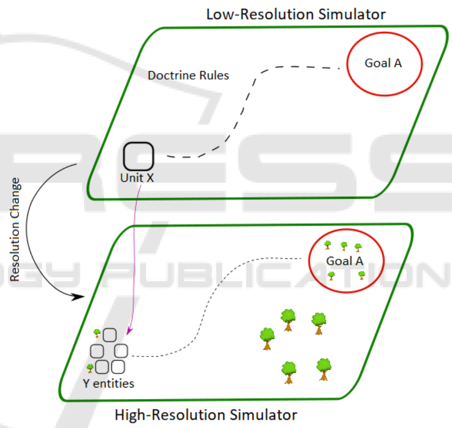


Figure 1: The disaggregation of simulated units from the low to the high-resolution simulator: the unity X is split into several entities Y.

In addition to the terrain differences between the involved simulators, the used military doctrines must be respected during the resolution exchange. As shown in Figure 1, the doctrine defines that the unit X disaggregation corresponds to five entities Y. The doctrine also defines that a circular displacement should be used when these Y entities are placed in the terrain managed by the high-resolution simulator.

Even if the simulation information is exchanged between simulators to ensure the correct resolution conversion, the problem of integrating the simulators when the simulation situation changes persist. In the high-resolution simulator, the Y entities must modify the situation at the simulation run time if the unit moves to a new area. This exchange also involves altering the doctrines that guide the simulation of Y entities. Although this exchange of doctrines to guide the Y entities' behaviors in the high-resolution simulator are relevant to produce faithful simulations, this exchange of doctrines presents an irrelevant level of detail to the low-resolution simulations.

An example of doctrine changes appearing in simulation exercises is presented in Figure 2, which shows that Y entities in the high-resolution simulator must move to goal A. As they are still under the control of the low-resolution simulator, these entities must perform the navigation behavior actions required by the X unit in the low-resolution simulator. To arrive at A, this unit follows a planned route (dashed line in Figure 2). However, according to the simulated doctrine in the low-resolution simulator, the movement of Y entities must follow a convoy formation along the dotted line path in Figure 2. When they arrive at goal A, their arrangement on that terrain location can also differ from the one used in the tactical position initially occupied by these entities. The result is the need to dynamically deal with the following simulation situation changes: the first occurs when the simulated entities leave the current tactical position, and the second when they arrive at a destination tactical position (goal A), in addition to executing doctrine-based convoy movement behaviors.

In summary, multi-resolution conversion depends on two issues: 1) the representation of additional information to regulate the simulation differences between the low- and high-resolution simulators; and 2) the mapping of simulation changes that occur at simulation run-time. This work proposes an XML-based representation language to express doctrine rules for mapping low and high-resolution distributed simulations to approach the first issue. It relies on ***dynamic multi-resolution conversion handlers*** to manage the simulation changes of the integrated agents to approach the second issue. Therefore, the research relies on the representation of doctrine rules to better specify multi-resolution conversions.

2.1 Multi-Resolution Aggregation/Disaggregation Conversion Rules

The multi-resolution conversion solution adopted in this work is based on previous advances found in the literature (Li et al., 2008) and (Paul et al., 2017). It explores the representation and computation of doctrine rules to guide the distributed simulations. The proposed solution extends the expressiveness of the doctrine rule description language detailed in (Paul et al., 2017). The multi-resolution doctrine rules aim to provide the information necessary for the high-resolution simulator to faithfully simulate real-world agent behaviors, where the simulated entities and their simulated behaviors are controlled by the low-resolution simulator. Although the low-resolution simulator owns these agents, and this ownership remains so during the course of the integrated simulations, these agents make a fundamental part of the simulations executed by the high-resolution simulator.

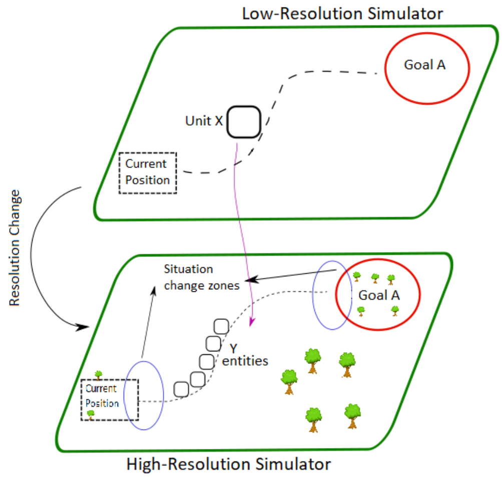


Figure 2: Low and high-resolution simulators showing different views of a simulated agent movement situation.

When implementing a new multi-resolution doctrine rule language, rule modeling concepts must be defined. In our work, these concepts were motivated by a case study developed in the SIS-ASTROS GMF project (Pozzer et al., 2022). One of the aims of this project is to integrate a high-resolution virtual tactical simulator with a constructive simulator while executing distributed simulation exercises. The integration rules rely on defining these domain concepts to detail different simulation situations. In this case study, the main attributes for specifying the multi-resolution conversion rules are the following: Object identification; Unity composition; Simulation situation; Formation organization; Agent position; Distance between agents; Distance between agents and terrain features; Central agent; and Agent orientation.

DoctrineRules: Fig. 3 shows that DoctrineRules is the root concept in the rule specification model. This concept is detailed by Entities and Units, organizing these elements in the doctrine rule representation.

Entities: Figure 4 shows that Entities are composed of one or more Entity elements. The attributes of these concepts are:

- name: identifies the simulated object;

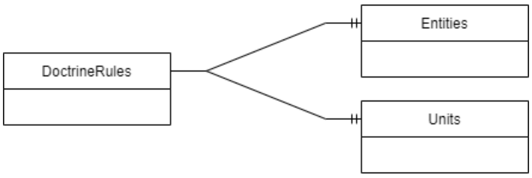


Figure 3: The representation of doctrine rules.

- FOMClass: describes the object type;
- enumeration: a unique numerical attribute for representing the simulated doctrines.

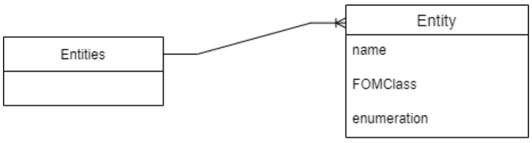


Figure 4: The Entity concept and its attributes.

Units: Figure 5 shows that the Units concept consists of one or more Unit elements. Each Unit is modeled by Composition and Behavior elements. The Unit attributes are the same of the Entity whenever the object is an entity and has the same purpose.

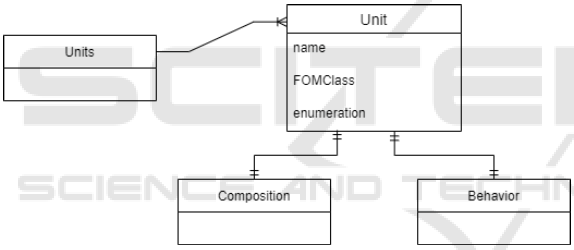


Figure 5: The Units concept, its subconcepts and attributes.

Composition: Figure 6 shows that the Composition concept indicates which entities, among those declared, are part of a Unit. It consists of one or more Entity elements. Its attributes are:

- type: identifies the entity type; it should correspond to a previous defined name of an Entity;
- id: a unique numerical ID attribute identifying an Entity of a Unit. When multiple entity types exist, this attribute distinguishes them.

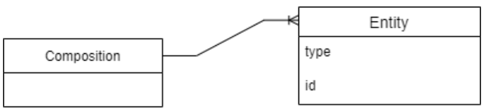


Figure 6: The Composition concept, its subconcepts.

Behavior: The Behavior concept indicates how units should behave in each simulated doctrine. This concept consists of one or more Situation elements and assumes an entity corresponds to a simulation agent. Its attributes are:

- name: identifies the doctrine situation;
- formation: formation organization adopted by the Entities of a Unit in the simulated situation;
- entityDist: if necessary, indicates the distance between entities in the simulated situation;
- terrainFeatureDist: indicates the distance the entities should keep from the terrain features observed in the simulated situation;
- entityRef: indicates the coordinate of the reference entity, which is used as reference for organizing other entities in the simulated terrain. This reference should be the related Unit on low-resolution simulator.

The Entity concept associated with the Situation concept is used to describe information that is specific to the entity within the simulated situation. The attributes that describe these concepts are:

- id: a unique numerical attribute identifying an Entity in the simulated situation. This numerical id is related to elements of the Composition concepts;
- position: indicated the terrain position the Entity is located in the simulated situation (and the terrain formation organization adopted by them);
- orientation: if necessary, it indicates the agent orientation in the simulated situation, e.g., North, South, etc.

2.2 The Representation of Dynamic Multi-Resolution Conversion Rules

The ***dynamic multi-resolution conversion rules*** are related to agent behavior changes occurring at simulation run-time. The dynamism concerns the smoothing that must occur whenever the doctrine rules guiding the agents' simulations change. To approach this problem, this work proposes determining ***situation change zones*** to address these events. When instantiating these zones in a simulated terrain, the determined zones indicate the areas where the simulated entity behaviors change from one doctrine to another.

In this doctrine-based multi-resolution conversion solution the situation change zones correspond to ***doctrine transition zones*** that are theoretical areas mapped into the simulated terrain at the simulation run-time. Figure 7 illustrates this zone from a parking situation to a movement situation in response to a movement order. When low and high-resolution simulators are integrated, these zones are created whenever agent movement orders are issued in the low-resolution simulator. When this order is executed, the agent moves from one terrain location to another.

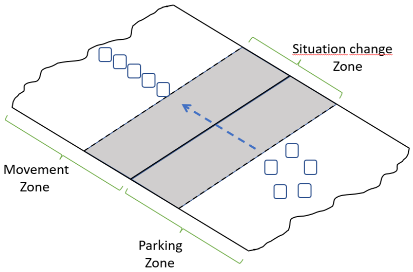


Figure 7: A situation change zone where occur a parking to movement doctrine transition.

Fig. 8 illustrates a unit movement in a simulation in response to a movement order in the low-resolution simulator. When the movement command is emitted, the first terrain zone is automatically determined in the high-resolution simulator. In the start zone, the disaggregated entities are under the doctrine rule behavior that must be used in that simulated tactical position (parking). When moving to leave this position, the unit executes a dynamic convoy movement behavior to change from parking to movement doctrine rule behavior. Consequently, the agents start following a different doctrine rule, a doctrine transition rule that activates an autonomous behavior. In the end zone, a similar behavior appears. The entities must change the executed doctrine rule (movement to parking).

With the definition of the simulation situation transition zones mapped to the simulated terrain, the proposed dynamic multi-resolution conversion follows the activities presented in Figure 9. Notice that the high-resolution simulator is responsible for executing the movement behaviors of the simulated unit. The low-resolution simulator controls the agents' movement positions in this distributed simulation scenario. When the unit moves in the low-resolution simulator, all entities mapped into the high-resolution simulator must move too. The low-resolution simulator also determines the target position. Thus, the

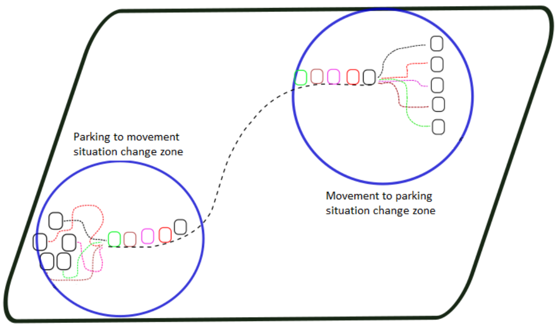


Figure 8: Start and end movement zones determination and their corresponding behaviors.

unit's movement should be mapped from the low to the high-resolution simulator. To approach this problem, the units' terrain coordinates are continuously updated and mapped from one simulator to another. According to the doctrine rules used by the high-resolution simulator, the unit terrain coordinates are used to determine whether any of the simulated units entered a ***situation transition zone***.

The low- and high-resolution simulator agent movement algorithms are executed when agents are within the simulation situation change zone. However, they cooperate to achieve the required realistic movement behavior in the high-resolution simulator. Then the ***multi-resolution doctrine conversion rules*** proposed in this work detail which terrain positions the units should be. When the simulated units enter the change zone determined in the simulated terrain, a doctrine rule relevant to that simulation situation change is selected, determining the new simulated doctrine these agents should start executing.

3 A CASE STUDY

A case study was developed to assess the flexibility and dynamism of the proposed multi-resolution conversion solution. This case study involves integrating simulated exercises developed in two distinct simulators: a low and a high-resolution simulator. Importantly, the low-resolution simulator contains a simulation scenario that should be realistically mapped into the high-resolution simulator. To do so, implementations of ***multi-resolution conversion handlers*** are used so that agents' aggregations are converted (via the RTI structure) from the low- to the high-resolution simulator, reaching the high-resolution simulator in a proper doctrine-based disaggregated format.

In the developed case study, the low-resolution simulator contains an aggregation of vehicles comprising a military battery. This aggregation involves multiple instances of three types of elements (i.e., military vehicles) named in this case study as Unid1, Unid2, and Unid3. Unlike the low-resolution simulator, the high-resolution one must simulate these entities individually. The implemented multi-resolution conversion handlers execute the specified multi-resolution doctrine rules to properly split the aggregated objects into their individual components.

When the units are disaggregated and represented in the high-resolution simulator, they must occupy the same tactical position as in the low-resolution simulated terrain. In this tactical position, agents must be displaced in the terrain according to a particular static formation organization detailed by the military doc-

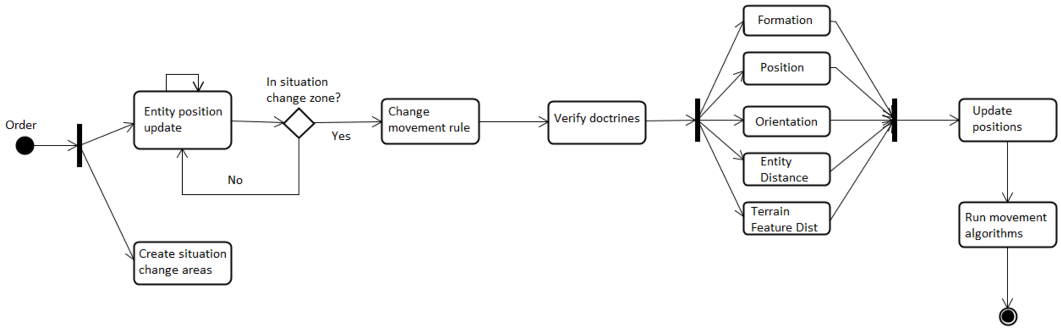


Figure 9: Activity diagram explored ***dynamic multi-resolution conversion handlers***.

trine. Therefore, the different disaggregated units involved in this case study, named U1EntA, U1EntB, U1EntC, U2EntA, U2EntB, and U3EntA, are positioned on the simulated terrain accordingly.

When agents' aggregations receive a command to occupy a new tactical position at run-time, these agents must perform a doctrine-based motion behavior. The transition from a static position occupation doctrine to a dynamic movement one is simplified in the low-resolution simulator. This simulator just needs to simulate an aggregate of agents (i.e. the whole battery) moving from one place to another. In this simulation level, the movement details of each unit in the aggregated entity are irrelevant.

As the low-resolution simulator controls the aggregate entity, the problem is that the transition between these static and dynamic doctrines must occur in an organized and realistic manner in the high-resolution simulator. So the proposed multi-resolution conversion handler should detail how to smoothly move between these doctrines while mapping these aggregate entities into individual units in the high-resolution simulation.

3.1 The Multi-Resolution Aggregation/Disaggregation Conversion

All the multi-resolution doctrine rules are checked during the simulation exercise's representation. Initially, declarations of the involved simulation entities and units are analyzed.

We highlight the doctrine rules' modeling allows unique identifiers for each entity within the simulated situations. It also allows handling doctrine rules individually. The optional attributes added in the multi-resolution doctrine rule representation favor capturing the specifics of simulated agent behaviors. The following attributes present examples of these simulation situations:

- *entityDist:* determines the distance between the vehicles in a convoy movement formation.
- *terrainFeatureDist:* determines the distance convoy vehicles must keep from obstacles and other simulated terrain features;
- *entityRef:* describes a central agent (e.g., a command vehicle) used by simulated doctrines, which are static and dynamic agent organizations involving a central element;
- *orientation:* determines the geographic direction in which the simulated agents are presented in 3D terrain scenarios.

The proposed multi-resolution doctrine rules represent the simulation information using the XML language. Starting from the proposals presented in (Paul et al., 2017), the XML representation allows multi-resolution conversion handlers to better process the data required to process the multi-resolution doctrine rules. As experienced in our project, such XML format is simple to understand, favoring the doctrine rules' modeling, revision, and updating.

3.2 The Dynamic Multi-Resolution Conversion

In the developed case study, the integration of simulators was tested with and without the use of dynamic multi-resolution conversion implementations. These handlers map run-time simulation situations, where agent behavior information is mapped from the low to the high-resolution simulator.

The ***dynamic multi-resolution conversion*** is based on the following simulation information: the convoy formation organization must be adopted by the simulated agents; there is a simulation command indicating the agents should act in such a way in the simulations; the agents are geographically oriented in the simulated terrain; the agents must consider the terrain elements when executing the simulated doctrine rules; and the agents must keep minimum and maximum distances between them.

In the performed implementations, an *Integration* module is used in the our case study. This module uses a target coordinate, received from low-resolution simulator via HLA, and calculates the movement parameters required by the agents of high-resolution simulator. These parameters are stored in the *Vehicle-Behaviour* handler, responsible for moving the agent in the simulated environment. These parameters are called *steeringParameters* (implemented as Steering Behavior algorithms (Reynolds et al., 1999)). They are necessary for the agent to transit through the simulated terrain to destinations avoiding collisions with static and dynamic obstacles. The algorithm for moving agents in the high-resolution simulator is called *VehicleBehaviour*. The setup parameters for these modules are the following:

- *Move* defines whether the agent should move. If false, the agent stops the movement;
- *MoveForward* defines whether the agent moves forward or backward;
- *DesiredSpeed* required movement speed in km/h.
- *SteeringCoefficient* defined the movement direction, varying from -1 to 1. Negative values make the agent turn left, and positives right. If the value is zero, the agent moves in a straight line.

An usual example of a dynamic simulation situation occurs when the aggregated Unid1 has to move from one tactical position to another. The Unid1 vehicles are in a marching column formation to simulate the desired doctrine movement behavior. When the vehicles reach the destination, they occupy the new tactical position accordingly. It means these simulated vehicles are displaced in another terrain formation, a ready line (side-by-side) formation.

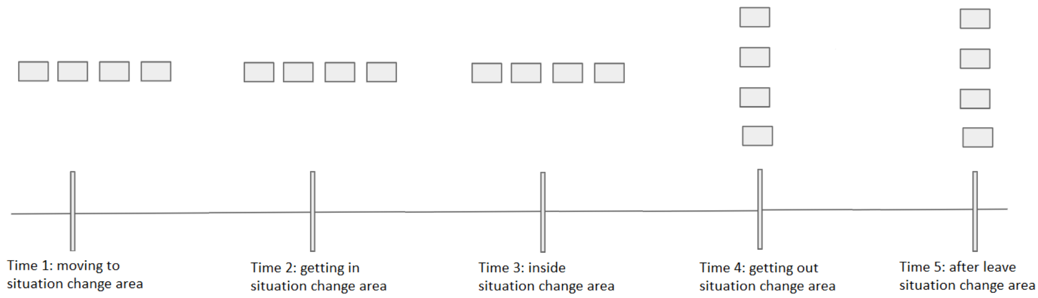
This simulation scenario can be detailed in five-time frames (i.e., simulation states). First, the Unid1 vehicles are out of a situation change zone and moving to it according a doctrine rule that specify column formation. Second, the Unid1 vehicles arrive in the situation change zone and getting in it. Third, the vehicles are inside the situation change zone and move in it. Fourth, the Unid1 vehicles arrive at the situation change zone border and cross it. Finally, the Unid1 vehicles arrive a new tactical position where the formation doctrine rule specifies an in line formation (side-by-side). As presented by the high-resolution simulator, Figure 10 details how these simulated vehicles (agents) behaviors can be mapped from the low to the high-resolution simulation scenario.

Fig. 10(a) presents a 2D projection of the high-resolution simulator without executing this dynamic multi-resolution conversion. Fig. 10(b) presents the same simulation situation where the high-resolution simulator uses the simulation information received from the dynamic multi-resolution conversions.

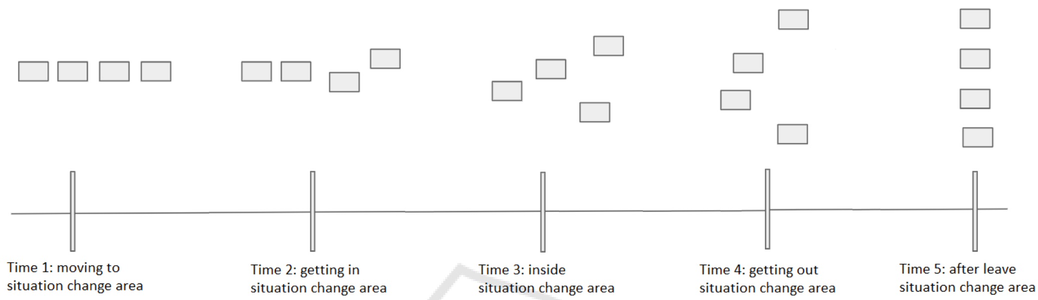
The four agents are organized in a linear formation in the first simulation time step. They are moving toward a simulation situation change zone. There is no difference between the simulated scenarios with and without the dynamic simulation treatment in this simulation period of time. The agents adopt a new formation organization in the second simulation time step. However, there is a gradual transition from one formation organization to another, controlled by the dynamic multi-resolution handlers. It is possible to observe the agents continue to move in the same ways when the dynamic simulation conversion is not executing. It means they do not change the doctrine movement behavior they are executing. The agents almost reach their destination positions in the third simulation time step. These positions are located inside the tactical position they should occupy. When the dynamic simulation treatment is not explored, the simulated agents keep the same movement behavior even when they almost reach their destination. In the fourth (and first) simulation time steps, the simulated scenarios with and without dynamic multi-resolution simulation handlers are identical in low and high-resolution simulators. However, there is a smoother and more realistic transition in the executed agents' navigation behaviors with the execution of the proposed dynamic multi-resolution handlers.

4 FINAL REMARKS

This work proposes a doctrine-based approach to solve the multi-resolution conversion problem in distributed simulation systems. To integrate simulators with different resolution levels, the proposed doctrine rule-based solution allows the representation of a greater variety of scenarios in which the transition between different simulators is required. This allows the representation of more specific simulation situations involving the transition of military doctrines into simulated terrain situations with different levels of detail in each integrated simulator. The multi-resolution solution proposed in this work allows more realistic integrated simulations, further increasing the role of doctrine-based rules in providing precise simulation data for resolution conversions. Directions for future work are focused on enhancing doctrine-based integration and, consequently, producing smoother and more realistic integrated simulations.



(a) High-resolution simulator where the dynamic multi-resolution conversion is not executed.



(b) High-resolution simulator where the dynamic multi-resolution conversion is executed.

Figure 10: The dynamic multi-resolution conversion between simulators.

ACKNOWLEDGEMENTS

We thank the Brazilian Army (Strategic Program ASTROS) for financial support through the SIS-ASTROS GMF project (TED 20-EME-003-00).

REFERENCES

Falcone, A., Garro, A., Anagnostou, A., and Taylor, S. J. E. (2017). An introduction to developing federations with the high level architecture (hla). In *2017 Winter Simulation Conference (WSC)*, pages 617–631.

IEEE (2010). Ieee standard for modeling and simulation (m&s) high level architecture (hla)– federate interface specification. *IEEE Std 1516.1-2010 (Revision of IEEE Std 1516.1-2000)*, pages 1–378.

Kong, Y. and Xing, L. (2013). Study of multi-resolution modelling framework in joint operations simulation. Proceedings of the 2013 International Conference on Information Tecnology and Computer Application Engineering (ITCAE 2013).

Li, M., Bi, C., Liu, X., and Niu, D. (2008). Research on rules of aggregation and disaggregation for aircraft entity models in air force combat simulation. In *2008 Asia Simulation Conference - 7th International Conference on System Simulation and Scientific Computing*, pages 1286–1290. IEEE.

Macal, C. (2016). Everything you need to know about agent-based modelling and simulation. *Journal of Simulation*, 10:144–156.

Paul, R. L., Nunes, R. C., Oliveira, V. D., and Kunde, D. (2017). Doctrine based multi-resolution hla distributed simulation. *Proceedings of the Symposium on Applied Computing*, pages 59–64.

Pozzer, C., Martins, J., Fontoura, L., Silva, L., Rutzig, M., Nunes, R., and Pignaton de Freitas, E. (2022). Sisastros: An integrated simulation system for the artillery saturation rocket system (astros). In *12th International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH 2022)*, pages 194–201. SCITEPRESS – Science and Technology Publications, Lda.

Reynolds, C. W. et al. (1999). Steering behaviors for autonomous characters. In *Game developers conference*, volume 1999, pages 763–782. Citeseer.

Tolk, A. (2012). Engineering principles of combat modeling and distributed simulation. Wiley.

Zhou, L., Mao, Z., Jing, F., Chen, Y., Gao, M., Wang, J., and Liu, Z. (2019). Design of satellite communication simulation training system based on hla, vr and opnet. In *2019 International Conference on Robots & Intelligent System (ICRIS)*, pages 125–128.