Multiresolution Modeling and Simulation of an Air-Ground Combat Application

Martin Adelantado
ONERA-CT/DPRS/SAE
French Aeronautics and Space Research Center
Long-term Design and Systems Integration Department
2, Avenue Edouard Belin
31055 Toulouse Cedex, France
Martin.Adelantado@cert.fr

Pierre Siron
ONERA-CT/DTIM/CVSI
Information Modeling and Processing Department
2, Avenue Edouard Belin
31055 Toulouse Cedex, France
Pierre.Siron@cert.fr

Keywords:
Distributed Simulation, High Level Architecture, Multiresolution,
Aggregation, Disaggregation, Air Defense Systems.

ABSTRACT: *The High Level Architecture (HLA) establishes a common modeling and simulation framework facilitating interoperability and reuse of simulation components. Since 1996, ONERA (French Aeronautics and Space Research Centre) carries out several studies on HLA in order to gain a better understanding of the underlying mechanisms of HLA implementations. The first critical step of this initiative was to develop our own RTI from the HLA specifications. In order to evaluate the cost of making a transition from legacy simulations to HLA, we first developed an HLA federation simulating an air-ground combat involving a set of aircraft's engaged against a surface to air defense system. Current studies on HLA distributed simulation include security, WAN simulations and multiresolution.*

Conventional simulations represent entities at just one single level of resolution. Multiresolution representation of entities consists in maintaining multiple and concurrent representations of entities. In this paper we address the problem of how HLA services may allow to achieve multiresolution modeling and simulation. Our goal is not to provide a general framework as a basis for designing simulations of entities at different levels of resolution concurrently. We focus on experience feedback we have obtained by migrating a single level resolution HLA federation to a multi-level resolution federation. The selected application is an Air-Ground Combat simulation involving aggregated patrols of aircraft's engaged against a surface to air defense system.

In this paper, we briefly describe the Air-Ground Combat simulation application. We then detail the multiresolution representation of entities (patrols and aircraft's), and discuss the chosen mechanisms allowing triggering aggregation from an entity-level representation, and conversely, triggering disaggregation from an aggregate representation. We focus on the HLA services we have selected to maintain several levels of representation concurrently and on methodological issues in designing multiresolution HLA simulations. We have tackled some difficulties and we propose a new HLA service that should make easier the user's task. This multiresolution management service can be added to our RTI or written by using existing HLA services. Finally, future trends are discussed.

1. Introduction

The DoD has proposed HLA for the next generation of modeling and distributed simulations [1]. HLA has to be considered as a collection of low level services supporting interoperability among a set of elementary simulators. However, the computing power needed to run large-scale simulations of high-fidelity entities, is not currently available. Therefore, simulations must be able to selectively vary the resolution of entities according to the situation. On the other hand, wargames and Command and Staff Trainers are typically provided by aggregate-level constructive simulations. Units are then represented at a low resolution level, without specifically modeling the individual components that make up these units. In order to avoid a re-engineering of legacy simulations, a linkage of existing heterogeneous simulations, has been suggested by several authors, for example see references

[2], [3] and [4]. Our purpose is to address a number of significant technical and methodological issues related to the multiresolution challenge, from an experimental perspective and using both the High Level Architecture and the ONERA Run-time Infrastructure. To achieve this goal, we have selected a simplified model of air-ground combat simulation. The engagement model between units has been suggested by a test application selected by the French CAD (Defense Analysis Centre) to illustrate the ESCADRE simulation support environment [5].

The remainder of the paper is organised as follows. In the next section, we describe the main features of HLA, and shortly outline the motivations of the ONERA RTI initiative. In section 3, we recall the main characteristics of the single level resolution application, focusing on the HLA services that have been used. In section 4, we detail the operational requirements of our multiresolution approach, and we discuss some methodological issues aiming at developing HLA federations in a cost-effective way. Section 5 discusses aggregation and disaggregation triggering mechanisms and presents how they can be implemented using HLA services. This work[1] has been also presented in [6]. In section 6, we propose an higher level service facilitating the design of multi-level resolution federations. In the last section, we discuss preliminary conclusions and current trends on multiresolution using HLA services, including improvements on fidelity (physical and tactical issues.

2. HLA Background and the ONERA RTI Prototype

The High Level Architecture (HLA) provides a framework facilitating interoperability and reuse of simulations [1]. HLA defines an elementary simulation as a *federate*, and a set of federates as a *federation*. The Defense Modeling and Simulation Office (DMSO) has sponsored the design of several freely available RTI's, which constitute an implementation of the HLA interface specifications. The last available RTI is the RTI NG (Next Generation) version 3, implementing the version 1.3 of the HLA specifications. Those specifications are now an IEEE standard for distributed simulation.

ONERA is strongly linked to the French Ministry of Defense (DGA), and both institutions are sharing the need for a generic distributed simulation architecture, which emphasizes both interoperability and reusability of simulations. Therefore, ONERA carries out since 1996, several studies on HLA simulation, including the design of its own Run-time Infrastructure (RTI). HLA has also been selected by ONERA in other initiatives, as the design of a common modeling and simulation infrastructure dedicated to airport simulation [7].

The ONERA RTI offers a reduced set of HLA services excluding the DDM (Data Distribution Management) services and the Ownership Management services. It has been developed in C++ language under Solaris 2.6 operating system, and offers a C++ API. The implementation follows the version 1.0 of the HLA interface specifications. The RTI prototype, depicted in Figure 1, offers some original characteristics, in particular it is built around an architecture of communicating processes. The RTI is a distributed system involving two kinds of processes, a local one (RTIA) and a global one (RTIP). Finally a library (libRTI) is linked with the source code of each federate participating in the federation. Communications between different components use standard protocols, such as TCP/IP and UDP. A more detailed discussion of the ONERA RTI is given in [8] and [9]. Current studies on HLA distributed simulation include security, WAN simulations and multiresolution.

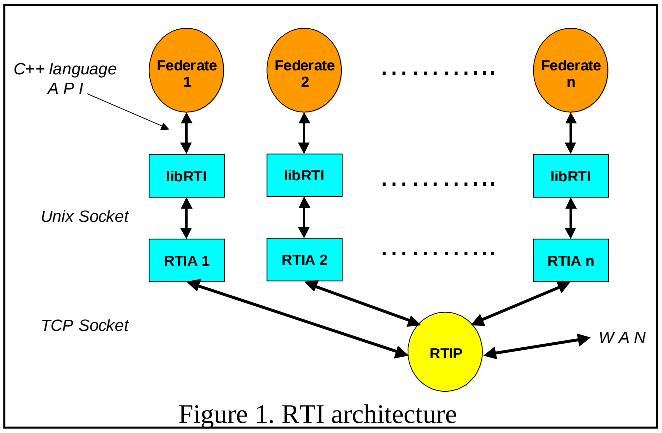

Figure 1. RTI architecture

3. Single Resolution Simulation: The Air-Ground Combat Application

In this section, we briefly describe the main features of the air-ground combat application we have extended to investigate HLA multiresolution simulations. The modeling and simulations details of both entities and engagement between them, have been described in depth in [10].

As mentioned above, the engagement model between units has been suggested by ESCADRE. The real world of the ESCADRE test application is depicted by Figure 2, taken from [5]. A patrol of aircraft's equipped with anti-radar missiles (ARM) is attacking a set of air defense units equipped with surface-to-air missiles (SAM) ramps. The ESCADRE application considers that aircraft's communicate through radio links, and that the air defense system is composed of several units (a command post, ground radars and surface to air missiles).

[1] This work has been partially funded by DGA (French Ministry of Defense).

In order to reduce development costs, the real world of the single level resolution application was simplified. First of all, each aircraft is equipped with its own radar devices and anti-radar weapons. Secondly, cooperation between the air defense system was not taken into account. Each ground unit is considered as being autonomous during the engagement. It detects aircraft's with its own radar devices and fires ground to air missiles independently from other ground entities.

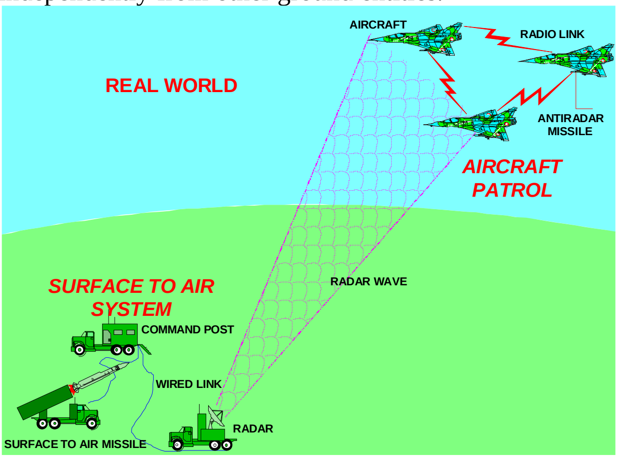

Figure 2. Air-Ground Combat Simulation

The aircraft mission planning model consists of a collection of aircraft’s and a set of Surface to Air Systems (SAM's), each of them including both a radar system and a missile system. In the following, each SAM will be referred as a Ground Radar. Aircraft’s are flying objects that fly through the air-defense system. Ground radar's are coupled objects of a radar system and a missile system. They are responsible for searching and tracking aircraft’s, and for launching a surface to air missile as soon as a target aircraft is closer than a lethal range. Each ground radar ensures target acquisition and target tracking. Missiles from ground radar's are flying objects that fly to the target aircraft and change their trajectory when they receive guidance commands from the radar. Conversely, missiles from aircraft’s fly to the target ground radar.

The aircraft’s mission scenario is randomly determined. In other words, the radar's position, the initial position of the aircraft’s and the corresponding trajectories are randomly set. As soon as an aircraft has been detected by a ground radar system, a missile is launched and guided to the target. The missile has given initial speed and operating range, that provide the aircraft with a level of survivability. Upon detection by a radar system, an aircraft changes its current trajectory, following a randomly generated fly path with a higher speed. A snapshot of an engagement is depicted in Figure 3.

Object-Oriented modeling of entities (aircraft's, ground radar's, missiles) has been performed according to the HLA Object Model Template (OMT). Missiles, Aircraft’s and Ground Radar's classes inherit from an super-class called Target, which has the following attributes:

State (destroyed, waiting, tracked)
X_Coordinate, *Y_Coordinate*
Color
ID (Object ID given by the RTI)
Radar_Range

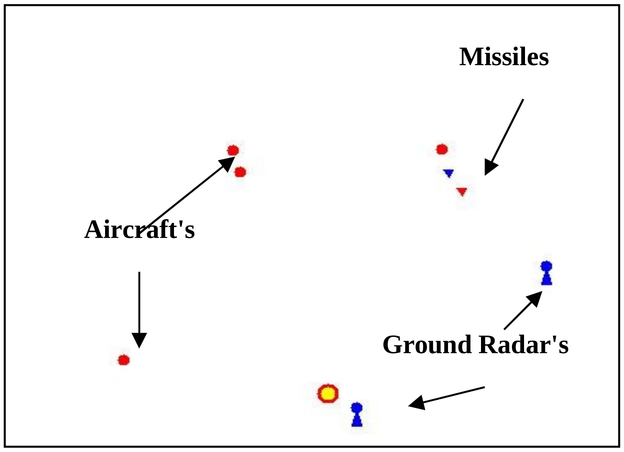

Figure 3. Snapshot of an engagement

The *State* attribute defines the current state of the target. X and Y Coordinates give the entity current position. The *Color* attribute is used to display the state of the corresponding entities on the simulated battlefield. *ID* attribute is an unique handle given by the RTI when a federate introduces a new object instance into the federation through the HLA registerObject() service call. Finally, the *Radar_Range* gives the range of the Radar or Aircraft’s radar system. The specific attributes of the Aircraft class are *X_Destination* and *Y_Destination*, the coordinates of the next point to reach in the fly path, *X_Direction* and *Y_Direction*, the direction coordinates, and *Speed* defining the current flying speed. Main attributes of the Ground Radar class, include *Survival*, providing a level of survivability to the corresponding ground system.

The HLA federation is organised around three main federates: *Aircraft* federates, *Ground Radar* federates and a *Display* federate providing a real time animation of the simulation. Notice that missiles (Air to Ground, Surface to Air) simulation is not provided by dedicated federates. The Air to Ground missile model has been included into the *Aircraft* federate, and each *Ground Radar* federate has to simulate the behaviour of the missiles it fires.

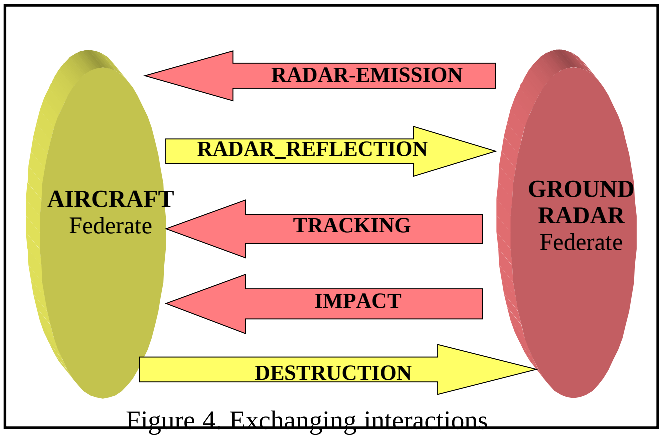

Figure 4. Exchanging interactions

Simulation of physical interactions (target acquisition, radar waves) between entities is based on the conventional interactions publication and subscription mechanisms provided by HLA. More precisely, simulation of target detection and localisation is performed by modeling emission and reflection of radar waves, using the concept of HLA *interaction*. As an illustration, the engagement between a ground radar and an aircraft is performed by exchanging interactions between the corresponding federates, according to the protocol given by Figure 4 and detailed in [10].

4. Multi-level Resolution: Basic Principles

Aggregate and disaggregate levels are respectively shown by Figures 5.1 and 5.2. At the aggregate level, a patrol of aircraft's has to attack a set of ground radar's. On entering the area of engagement, the patrol automatically disaggregates into its individual entities. The engagement is then managed according to the rules described for the single resolution application.

Classically several approaches may be used to manage the aggregation/disaggregation processes, according to the particular needs of the exercise [11]. These approaches include fixed geographical area, manual triggering, spheres of influence or event based. We have selected a sphere of influence based mechanism to automatically force a patrol to disaggregate into its components. Typically, a sphere of influence is placed around patrol entities, the radius of the sphere being chosen to be larger than the possible interaction range of the high resolution entities. In this way, patrols are disaggregated before interacting with ground radar's. Technical issues of the triggering conditions will be explained more deeply, in the next section.

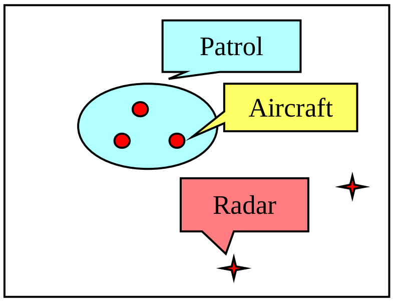

Figure 5.1 Aggregate level

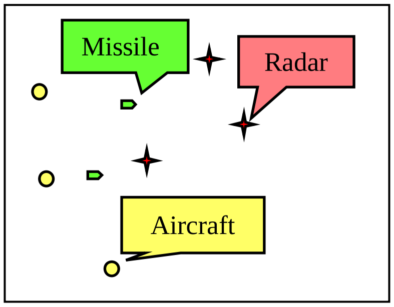

Figure 5.2 Disaggregate level

From a methodological perspective, a key issue is to determine the HLA federation architecture. Several architectures have been selected (Figure 6).

In the *centralized approach*, a single federate ensures the simulation of both low and high resolution entities. The advantage of this approach lies in the fact that interactions between entities do not have to use HLA services. The global state of the entities is then immediately known at anytime by everybody. The drawback of such a federation architecture is the low parallelism rate.

Conversely, in the fully *distributed approach*, each aggregate or disaggregate entity involves its own federate. With such an architecture, each federate may run on a different hosts of the network, thus improving the parallelism rate. The amount of exchanged messages between federates and the difficulty to extract a global state at a given time are the main drawbacks of this approach.

Finally, the 2 federates approach may offer a compromise between distribution and amount of exchanged messages.

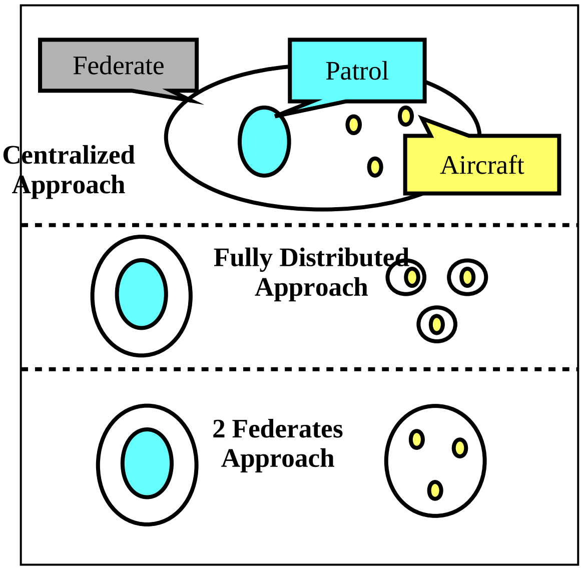

Figure 6. Federation architectures

Some other criteria may influence the architecture design: for example, in the context of updating an existing federation where all entities were simulated at a high resolution level, the centralized approach implies a complete re-engineering of the existing federates, while the fully distributed approach requires less massive changes.

We have implemented both the centralized and the fully distributed architectures, requiring in both cases the introduction of a Patrol entity, modeled by a Patrol class. The attributes of this class are rather similar to the aircraft's ones in the single resolution simulation, but they do not inherit from the Target class. In the distributed design, adding a "Name" attribute is necessary for the aircraft's federates to identify the patrol they belong to.

5. Aggregation-Disaggregation Mechanisms

5.1 Basic principles

With a "sphere of influence" approach, one of the main concerns lies in the problem of identifying the event that triggers the resolution level change. Figure 7 depicts the notion of sphere of influence – in the case of a radar, notice the difference between its sphere of influence and its action field. The patrol entity must be disaggregated as soon as its location makes the risk of combat initiation higher. In the same time, we have to keep in mind that high frequency resolution changes have to be strictly avoided regarding to simulation performance and consistency maintenance.

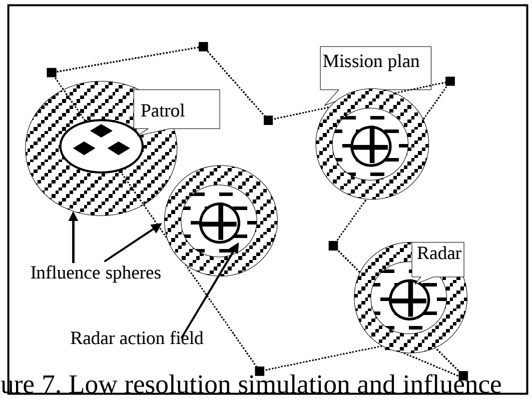

Figure 7. Low resolution simulation and influence spheres

Maintaining consistency within the federation precisely leads to some questions:

- How to deduce the positions of the aircrafts from the patrol's one after a desaggregation ?
- After an aggregation, does it make sense to still consider the patrol notion if several of its aircraft's have been destroyed ?

Even though the implementation of the aggregation-disaggregation mechanisms involved is quite different in the two chosen approaches, the main concepts remain the same in both alternatives. In this section, we first explicit them, before analysing separately the particularities of the two architectures under investigation.

When a patrol federate enters a radar sphere of influence, it must disaggregate. All the aircraft's of the patrol need to be initialized, and registered within the HLA federation. Their attributes are deduced from the patrol's ones together with the configuration of the patrol – for example, the position of the patrol could be the imaginary point matching the center of gravity of the set of aircraft's: in that case, the location of each aircraft would be easily computable. More generally, several disaggregation templates have been introduced. Each template has to be considered as a disaggregation function providing both location and state of the aircraft's from the patrol one's. Upon disaggregation, each aircraft must immediately get its planning mission: considering that solving this problem in a realistic manner would require an actual knowledge of the real-world doctrines, we chose to assign all the aircraft's the same mission plan than the patrol they belong to.

Identifying the conditions that may trigger the aggregation process is rather unclear if we consider the behaviour of the aircraft's in the initial single resolution simulation. As they take a random escape when they are tracked, it is rather difficult to imagine any realistic aggregation event. So we have established a *rendez-vous* approach, to which we have associated a set of requirements:

- None non-destroyed aircraft is located within the influence sphere of any radar,
- None aircraft is being tracked by a surface-to-air missile from a ground radar,
- None aircraft is currently guiding any missile.

When these requirements are met, and when all the aircraft's have reached the assigned *rendez-vous* point, then an aggregation can be triggered.

5.2 Centralized approach

One unique federate is handling the patrol and its aircraft's. Thus if we do not consider that the display federate has to know about them, the resolution level transition processes do not require communication between federates through the HLA services. 2 interactions are then only needed, called respectively AGGREGATION and DISAGGREGATION. These interactions are published by the patrol-aircraft's federate while the display federate subscribes to them.

5.3 Fully distributed approach

Among some other possibilities, we chose to centralize the knowledge of the global state of the simulation in the patrol federate: this leads to the definition of a minimal activity for the patrol federate even during high resolution phases – the patrol federate has to know permanently the position of its components, as well as their state (tracked, normal, destroyed, waiting). This approach is typically referred as *pseudo-disaggregation*.

Obviously, more communication between federates than in the centralized approach is needed. We define three main interactions allowing the handling of the aggregation and disaggregation processes:

- AGGREGATION and DISAGGREGATION. These two interactions play an extended role, as not only the display federate but also the aircrafts' ones have to subscribe to them. By sending a DISAGGREGATION interaction with the appropriate parameters to all its patrol members, the federate initializes the control transfer from the low level resolution to the higher one.
- WAITING. When an aircraft reachs *a rendez-vous* and satisfies the 3 conditions detailed previously, it sends a WAITING interaction to its patrol federate. When all the federates have sent their WAITING interactions, the patrol federate can answer by emitting its AGGREGATION interaction.

To minimize the amount of shared messages, an optimization of the publish and subscribe HLA rules is needed: the federates update dynamically their publication and interests, according to their current level of resolution.

5.4 Executing the federation

The federation, developed in C++ language, is automatically started by a script. Running the federation consists in executing the participating federates including the RTI, on different workstations of a local network. The number of engaged entities (patrols, aircraft's per patrol and ground radar's) is selected by the user before simulation. A screenshot of an exercise provided by the *Display* federate is depicted by Figure 8.

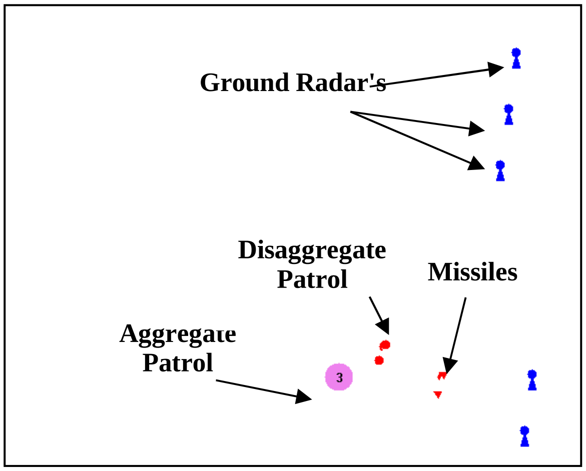

Figure 8. Screenshot of an engagement

This figure shows both an aggregate patrol, and a disaggregate patrol consisting of 3 aircraft's engaged against 2 ground radar's.

6. Multiresolution Management Service

6.1 Statement

Before to propose the specifications of a multi-resolution management service, it is necessary to well understand what are the technical problems stated by the management of multiple resolution levels in our application, and then to analyse what are the points that could be generalised.

In the case of many patrols, how a given aircraft can clearly identify to which patrol it belongs? The code of the aircraft federate is generic; it is the same for each aircraft. When an aircraft federate starts, it receives the name of the patrol as a parameter. It saves this name in an attribute of a federate class, attribute that will be used during the whole federation life. These identifiers must be global and unique. The multi-resolution management service must propose a mechanism to make real and easy the membership relation, that we will call delegation relation between HLE and LRE (High and Low resolution Entities).

At disaggregation time, the patrol sends a DISAGGREGATION interaction that its members have to receive. This mechanism should be masked in the proposed service. Parameters of the interaction include the patrol position, the theoretical aggregation point, etc. Each aircraft analyses these parameters and determines its position with a standard pattern. Therefore the attributes of the LREs are computed from the attributes of the HRE, and it is necessary to formalise these mapping functions between the attributes.

The aggregation problem is in one hand similar, we are using the delegation relation and reverse mapping functions, and it is on the other hand more complex. We can use an AGGREGATION interaction but we have especially to implement a solution to the distributed consensus problem. When do all the aircraft agree to make again a single entity? This is a well-known problem in distributed computing, and this problem can even be impossible to solve in some cases (faulty federate). We will propose only basic functions.

Therefore we think that the multi-resolution management service has to address four points: delegation relation, aggregation, disaggregation and attribute mapping.

6.2 Implementation alternatives

This new service could later be implemented in our CERTI, more precisely we could limit the modifications to the libRTI. That is to say that an alternative is to build this service on the existing and standard HLA services and probably a new federate. This solution should be applicable to other RTI. Another alternative is to update the CERTI kernel to improve performance.

6.3 Delegation relation management service

All the entities can be organised in a hierarchical structure. We need functions allowing each federate to declare itself if it is a LRE or an HRE with a delegation relation.

We cannot avoid the use of static alphanumerical identifiers for each federate. But the RTI can manage a correspondence table between these names and the corresponding dynamic handles that will be used in the following.

Four new functions are defined that we describe briefly in this paper. The specifications are explained in depth in [12] as well as UML scenarios.

> **void registerAsLRE (String oneAlphanumericId)**
>
> The federate is a LRE.
>
> **void registerDelegationRelation (String oneLREAlphanumericId)**
>
> The federate is an HRE linked to the federate which the static identifier is the input parameter.
>
> **void registeredAsLRE (ObjectID lreID, String lreSID)**
>
> This is a RTI initiated method for all federates. Parameters are the dynamic and static identifiers.
>
> **void delegationRelationRegistered (ObjectID lreID, ObjectID hreID)**
>
> This is the callback associated to the *registerDelegationRelation* method, parameters are the identifiers of the LRE and one HRE federate.

6.4 Disaggregation management service

The LRE federates make use of a disaggregation function and the HRE federates will receive a callback. The delegation relation information serves so that only the affected federates are informed.

This mechanism is a little more complex if another federate, for example a Display federate, wants to receive the transition management information. To cope with this problem, we propose a new method that an HRE or LRE federate does not need to call.

We add three functions to the multi-resolution management service.

> **void disaggregation (ParameterList listofParameters)**
>
> The list of parameters is useful to send some initialisation data to the HRE federates.
>
> **void reflectDisaggregation (ObjectID lreID, ParameterList listOfParameters)**
>
> This is a RTI initiated service, *lreID* is the identifier of the LRE federate which is disaggregated.
>
> **void subscribeToResolutionTransitions**

(FederateType lreType)

6.5 Aggregation management service

This service must be enough flexible to allow the user to choose between various strategies:

- The HRE federates can call the *readyForAggregation* method if they are the initiators of the aggregation process. In this case, the RTI sends a *reflectAggregationRequest* to the implicated LRE federate.
- The LRE federate decides the aggregation at its own initiative or after it has received the requests of its HRE federates. It then uses the *aggregation* method, which involves *reflectAggregation* callbacks.

So we have four new services.

void readyForAggregation ( )

void reflectAggregationRequest ( )

void aggregation ( )

void reflectAggregation (ObjectID lreID)

6.6 Attribute mapping management service

The design of the mapping functions remains obviously the programmer task. The multi-resolution management service can help to the attribute transport from one resolution level to the other, that is to say from one federate to the other in a distributed implementation. Each federate designer defines the list of attributes to transmit and writes the methods to analyse the list of received attributes. This mechanism is mainly used at aggregation time. We can propose four new services.

void requestAttributesForAggregation ( )

The LRE federate asks the HRE federates for their attributes.

void sendAttributesForAggregation (ParameterList listofParameters)

This method is used by the HRE federate. The two following methods are RTI initiated.

void reflectAttributesForAgregationRequest ( )

void receiveAttributesForAggregation (ObjectID hreID, ParameterList listOfParameters)

7. Discussion and On-going work

ONERA has initiated a long term action on HLA simulation and RTI implementation. Multiresolution is one of the key technical issues we are interested in. The multiresolution simulation application we have described, has to be considered as an incremental and experimental test-bed aiming at investigating HLA mechanisms supporting multiresolution simulations.

First results show that a fully distributed approach facilitates the migration from a single-level resolution application to a multi-level resolution one, in that the underlying models of the components can be directly re-used. Conversely, maintaining a concurrent representation of both levels of resolution is more difficult, since the global state of the situation has to be known by all the entities. To face this requirement, a pseudo-disaggregation approach of low-level resolution entities is proposed. A critical issue of multiresolution simulation lies in the fact that triggering mechanisms are application dependent and then cannot be easily generalized. That is the reason why a specific application has been selected for investigation. The second preliminary conclusion is that HLA interactions provide a useful support to manage the aggregation and disaggregation processes.

Nevertheless, much more work has to be done to achieve our expected objectives, and 3 main research directions are currently considered.

First of all, ONERA as an institution of the French Ministry of Defense, is interested in on-going work aiming at improving fidelity of the air-ground combat model. From this point of view, we are enhancing the mission planning model and designing a more accurate physical and tactical model of the engaged entities (patrols, aircraft’s, radar’s and missiles).

Secondly, we are extending the synthetic battlefield in order to face the critical problem of spatial consistency. This problem may occur when the disaggregation template controlling the location of the high-level resolution entities, is not valid with regard to the underlying terrain characteristics.

Finally, although HLA interactions are useful to design multiresolution simulations, they are too much low-level oriented. Therefore we are investigating HLA based higher level services, encapsulating both aggregation/disaggregation interactions and transfer of control from one level of resolution to another one.

8. References

[1] DMSO, "The DMSO High Level Architecture (HLA).", 1996.

[2] A. Cox, J.M. Maybury and N. Weeden, "Aggregation

Disaggregation research - A UK Approach." In Proceedings of the 13th DIS Workshop (Orlando, FL, September 18-22), pp. 449-464, 1995.

[3] N. Weeden and M. Smith, "Aggregation Disaggregation research - A UK Approach Update." In Proceedings of the 15th DIS Workshop (Orlando, FL, Sept.), pp. 305-314, 1996.

[4] J. E. Kester, "An approach to aggregation and disaggregation for J-MASS system models." In Proceedings of the 15th DIS Workshop (Orlando, FL, Sept.), pp. 71-78, 1996.

[5] J.-L. Igarza, and Ch. Sautereau, "Development of a HLA compliant version of the French ESCADRE simulation support environment (SSE): lessons learned and perspectives." In Proceedings of the 1998 Spring Simulation Interoperability Workshop (Orlando , FL, March 9-13). SISO, pp. 192-201, 1998.

[6| M. Adelantado, S. Bonnet and P. Siron, “Multiresolution Modeling and Simulation with the High Level Architecture.” In Proceedings ESS’2000, 12th European Simulation Symposium (Hamburg, September 28-30), 2000.

[7] M. Adelantado, "Experimenting the HLA framework for the ONERA Project Airport of the Future." In Proceedings of the 1999 Fall Simulation Interoperability Workshop (Orlando, FL, September 12-17). SISO, 1999.

[8] P. Siron, "Design and Implementation of a HLA RTI Prototype at ONERA." In Proceedings of the 1998 Fall Simulation Interoperability Workshop (Orlando, FL, September 14-18), SISO, pp. 185-188, 1998.

[9] P. Bieber, J. Cazin, P. Siron and G. Zanon, "Security Extensions to ONERA HLA RTI Prototype." In Proceedings of the 1998 Fall Simulation Interoperability Workshop (Orlando, FL, September 14-18), SISO, pp. 511-516, 1998.

[10]M. Adelantado and P. Siron, "Air-ground combat simulation through the ONERA HLA run-time infrastructure." In Proceedings of the Symposium on Military, Government and Aerospace Simulation, MGA2000 (Washington D.C. April 16-20). SCS, pp. 191-196, 2000.

[11]P.F. Reynolds, A. Natrajan and S. Srinivasan, "Consistency Maintenance in Multiresolution Simulations." ACM Transactions on Modeling and Computer Simulation, Vol. 7, N° 3 (July), pp. 368-392, 1997.

[12]S. Bonnet, “Simulations HLA à résolution variable.” DEA Student Report, ENSAE-ONERA, Toulouse, 2000.

Author Bibliographies

MARTIN ADELANTADO was graduate from a French engineer school of computer science (ENSEEIHT) in 1979, and received his doctorate in 1981. He is an ONERA (French Aeronautics and Space Research Center) Research Engineer and works at the Aeronautics Systems Research Unit of the Long Term Design and Systems Integration Department (DPRS/SAE). His fields of interest include simulation, real-time systems and distributed systems.

PIERRE SIRON was graduate from a French engineer school of computer science (ENSEEIHT) in 1980, and received his doctorate in 1984. Then he is a Research Engineer at ONERA and he works in parallel and distributed systems and computer security. He is a member of the Design and Validation of Computer Systems research unit.