

ARCHITECTURE DE COMMUNICATION POUR L'INTERCONNEXION TEMPS RÉEL DE SIMULATEURS DISTRIBUÉE

Akram HAKIRI^{*1}

Pascal BERTHOU* et Thierry GAYRAUD*

¹ahkiri@laas.fr

Résumé .

Sur la base d'un problème portant sur la réalisation d'une plateforme de communication pour une application de simulation distribué interactive, nous nous intéressant à la description de l'évolution des architectures de communication et des standards proposés pour répondre aux exigences de ces applications. Nous présentons des solutions permettant de répondre aux nouvelles exigences en terme de qualité de service et de communication temps réel dur tout en assurant l'interopérabilité et la réutilisation des simulations.

Mots clés

Simulation Distribuée Interactive, SIMNET, DIS, HLA, DDS.

1 INTRODUCTION

La simulation distribuée interactive (DIS) est issue du domaine militaire, l'objectif initialement visé étant la préparation, en simulation, des troupes militaires à toute forme d'engagement (combats, déplacement,...) dans des environnements divers (terrestres, maritime ou aérien). Les applications DIS visent à proposer une architecture de communication commune permettant l'intégration et l'interopérabilité d'un ensemble de simulateurs géographiquement distants. Ce étant, elles permettent de réduire le temps et le coût de la conception de prototypes, leurs développements, leurs tests et le raffinement de leur cycle de vie. En outre, Ils offrent aussi des moyens pratiques pour évaluer les performances des modèles conçus pour la distribution à large envergure. Ce travail présente un état de l'art sur la simulation distribuée.

Dans cet article, la section (2) décrit le protocole DIS. La section(3) décrit le standard de communication de haut niveau HLA. La section (4) dévoile DDS. La section (5) présente un exemple implémentation des protocoles HLA et DDS. Une conclusion est présentée à la section (6).

2 Le protocole DIS : Distributed Interactive Simulation

Le protocole DIS (figure 1) a été développé à travers une série de workshop au sein du symposium de la communication interactive à l'université de Florida (IST) pour améliorer les applications tournantes sur SIMNET [1]. Depuis 1993, DIS qui est devenu un standard IEEE 1278.x [2]. Un exercice DIS regroupe plusieurs sites reliés entre eux par des réseaux de communication. Sur chaque site s'exécute un ou plusieurs applications de simulation partageant un même espace virtuel. Chaque logiciel DIS simule localement le comportement d'un certain nombre d'entités en interaction entre eux et incorporant des hommes dans la boucle de simulation. Ces derniers pilotent la simulation en commandant par exemple le changement de direction ou l'exécution d'un tir. Chaque application simule également le comportement des entités distantes sur la base des informations communiquées sur le réseau de communication.

Les informations d'états sont encodées dans des messages dits Protocol Data Unit (PDU) et l'échange de données entre hôtes utilise le protocole de la couche de transport classique à travers UDP en Multicast. La version courante du protocole d'application définit 67 types de PDUs regroupés sous 12 familles. Les principales familles de PDUs les plus utilisées sont présentées dans le tableau 1.

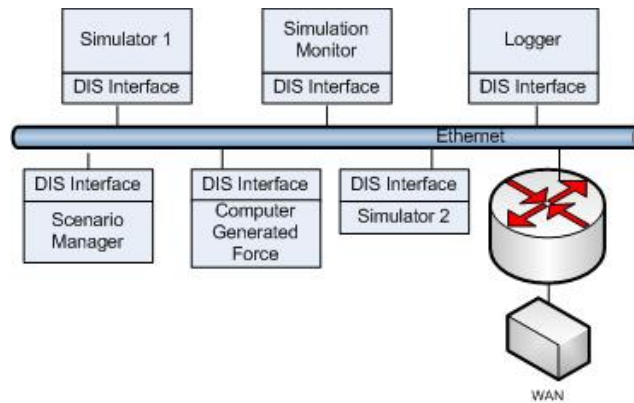


Figure 1 Architecture d'un simulateur DIS

Famille	Rôle
Entités Information/interaction	État des entités, collision, mise à jour des états des entités, ...
Entités Guerre	Feu, détonation,...
Entités Logistique	Service requête, réparation, réapprovisionnement,...
Management de la simulation	Start/Resume, Stop/Freeze, Ack, ...
Régénération de la distribution des émissions	Émission électromagnétique, acoustique sous marin, ...
Communication radio	Signal, récepteur, transmetteur, ...

Tableau. 1. Les principales Famille des PDUs DIS

Afin de réduire le nombre de mise à jour des états de ces entités, DIS employait plusieurs mécanismes permettant d'optimiser l'utilisation des ressources : L'agrégation de paquets se fait en fusionnant les messages en 1 seul paquet afin d'éviter l'envoi multiple des en-têtes ; l'algorithme PICA [3] permet la compression des données par envoi des changements des états entre deux messages consécutifs; Chaque site maintient un modèle de navigation à l'estime et de filtrage de relevance (Dead Reckoning) pour estimer les états des entités locales et distantes. Les états anticipés sont calculés à partir des informations d'états dans leur passé en utilisant des équations d'extrapolation. Ainsi, après chaque mise à jour de son entité, un site compare les valeurs réelles obtenues à partir d'un modèle de haute fidélité et son image extrapolée. Si la différence entre les deux modèles excède un certain seuil prédéfini, un message de mise à jour doit être envoyé aux sites distants. L'extrapolation de l'entité doit être corrigée par le modèle DR sur chaque site. Le filtrage de relevance concerne l'élimination de l'envoi des mises à jour inutiles.

Ce étant, avec DIS il était difficile de supporter les fortes interactions à cause du niveau de couplage qu'accorde le protocole aux entités simulées [4]. Les latences n'étaient pas maîtrisées et aucun service de gestion de temps n'était incorporé. En outre, le protocole contient des formats et des règles de message destinées aux applications militaires.

3 Architecture de Haut Niveau : HLA

Depuis des années, le standard du protocole DIS a fourni des bases solides pour la simulation distribuée temps réel. DIS a été largement accepté par les industriels et les gouvernements. Le protocole ALSP a été conçu pour supporter la simulation à événements discrets et a été implémenté avec succès dans les jeux de combats en ligne [5]. Pour unifier ces domaines et étendre leur succès

vers les applications existantes, le département de la défense Américaine (DMSO) a proposé en 1995 le développement d'un nouveau standard pour la modélisation et la simulation appelé HLA. HLA est une initiative pour saisir les meilleurs côtés de DIS et de ALSP et pour fournir en même temps une architecture standard pour la simulation logicielle.

HLA est un standard IEEE 1516 qui décrit une méthodologie qui focalise l'établissement de la simulation distribuée de haut niveau pour faciliter l'interopérabilité des modèles et des simulations et faciliter la réutilisation des composants existants de modélisation et de simulation (M&S). Elle permet d'exploiter de multiples programmes existants pour créer une simulation à grande envergure par la spécification de mécanismes permettant la distribution de la communication.

Le concept de simulation distribuée sous HLA est illustré dans la figure 2. Chaque simulation participante, dite fédéré, interagit avec d'autres fédérés au sein d'une collection de simulations dite fédération. Les fédérés communiquent entre eux au sein de la fédération à travers un middleware dit RTI (RunTime Infrastructure) [5]. Chaque fédéré peut publier des informations vers d'autres fédérés au sein d'une même fédération et s'abonner pour refléter certaines informations créées et mises à jour par d'autres fédérés.

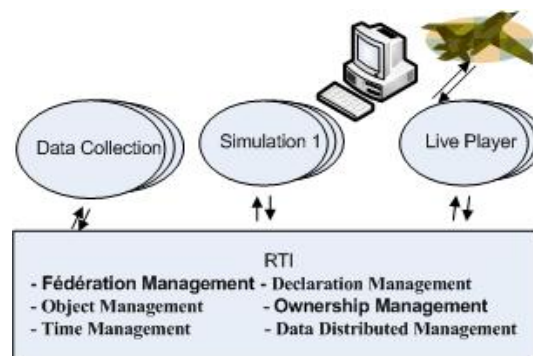


Figure 2 : Architecture d'un simulateur DIS

Les informations échangées en HLA sont représentées sous la forme d'objets. Les deux sortes d'objets échangés dans HLA sont les classes d'objets et les classes d'interactions. Les instances du premier type sont persistantes pendant la simulation, les autres sont juste transmises entre les fédérés.

3.1. Les Règles HLA:

Les règles sont des conventions qui doivent être respectées pour permettre l'interaction entre les fédérés au cours de l'exécution de la fédération. Ils expriment des concepts et des contraintes pour la conformité du standard HLA. Le standard spécifie 10 règles; 5 pour les fédérés et 5 pour la fédération. Plus de détails sont disponible dans le standard [6].

3.2. HLA Object Management Template (HLA-OMT)

HLA-OMT définit une structure formelle permettant de décrire le modèle objet guidé par SOM (Simulation Object Management) et FOM (Federation Object Management) [7]. C'est un méta modèle utilisé pour le développement de la simulation. Il fournit un mécanisme transparent pour l'échange de données et la coordination entre les fédérations. C'est un fichier texte (en format xml pour HLA 1516 ou Fed pour HLA 1.3) qui décrit des mécanismes communs et standards permettant de spécifier la capacité de chaque fédération. Le concept OMT est indispensable pour l'interopérabilité ; il focalise sur la description de l'aspect critique de l'échange et de partage entre fédérations ainsi qu'il permet de définir la qualité de service à accorder à chaque objet échangé.

3.3. Interface Spécification (IS)

HLA spécifie deux formes de modèles objets : FOM et SOM; HLA-SOM définit les informations que peut produire ou consommer une fédération. Les informations décrivent la simulation en terme de types d'objets, des attributs (dont les valeurs définissent l'état de chaque objet à n'importe quel point de la simulation) et les interactions qui décrivent les événements que peut générer une fédération ou peut y réagir.

HLA-FOM définit les informations échangées entre fédérés au cours de l'exercice. Toutes les fédérés doivent être d'accord sur les FOM afin d'assurer l'interopérabilité. L'échange des informations est géré par un processus producteur/consommateur. Chaque fédéré publie la mise à jour des états des entités qu'elle gère et s'abonne aux informations qui lui sont destinés par les autres fédérés. Ce mécanisme s'appelle l'exécution de fédération.

3.4. HLA-RTI

RTI est une collection de logiciels fournissant des services permettant la distribution des données au cours de l'exercice. Les fédérés envoient les données à travers cet intergiciel qui se charge de les acheminer aux endroits appropriés. RTI spécifie 6 groupes de services communs spécifiés dans l'interface de spécification HLA:

- **Management de la Fédération:** ces services fournissent un modèle pour l'administration de la simulation et des services de découvertes nécessaire à l'exécution de la fédération; il traite la création, l'initialisation, le contrôle dynamique, la modification, la suppression des fédérations en se basant sur les FOM, la façon avec laquelle les fédérés peuvent rejoindre ou quitter la fédération, la sauvegarde et la restauration des états au cours de l'exécution de l'exercice et la gestion des points de synchronisation et la destruction de la fédération.

- **Management des Déclarations:** ces services permettent aux fédérés de déclarer au RTI leurs intentions à publier ou à recevoir des interactions, des états et des attributs.

- **Management des Objets:** ces services fournissent un paradigme d'interactions des objets éloignés comme ceux de Microsoft D-COM et CORBA. Ils fournissent une couche d'abstraction entre l'invocation des méthodes des objets et les endroits où ces objets résident réellement et à quel moment de la simulation ils seront appelés. Ces services présentent la plupart du trafic réseau au cours de la simulation.

- **Management des propriétés:** ces services permettent aux fédérés de transférer la propriété des attributs d'objets vers les autres participants de la simulation. Ceci facilite la création d'un modèle coopératif d'une entité à travers de multiples hôtes et fournit les moyens de migration des objets d'un hôte vers un autre.

- **Management du temps :** fournit les services permettant la configuration, la synchronisation et la modification des horloges de simulation. Chaque fédéré maintient deux horloges locales : une horloge physique utilisé pour la synchronisation entre les individus et les entités réelles et une horloge logique (temps simulé) pour assure la délivrance des messages et des événements dans le bon ordre.

- **Management de la distribution des données:** Ce service facilite le transfert et la des mises à jour des objets entre les fédérés en utilisant le modèle producteur consommateur. Ce modèle est étend les mécanismes utilisé dans le prédécesseur de HLA (DIS-ALSP) pour les communications en multicast en fournissant un service à la demande.

4 Data Distribution Service : DDS

DDS est un standard spécifié par l'OMG [14], dont le rôle est de proposer une technologie évoluée d'échanges de données sur des réseaux allant des systèmes embarqués vers les réseaux à grande distance en se basant sur une architecture producteur/consommateur. Cette spécification

définit une API permettant à des applications hétérogènes de communiquer entre elles. Les API et la qualité du service (QoS) incluses dans DDS sont choisies pour équilibrer le comportement prédictible et l'implémentation efficace et performante.

DDS fournit de nouveaux aspects non inclus dans HLA [10]: Un ensemble riche de politiques de qualité de service, un modèle de données fortement typé, un soutien de la propagation d'état comprenant la distribution cohérente et ordonnée des données tout en omettant quelques autres aspects abordés par HLA comme le management du temps et le management de fédération.

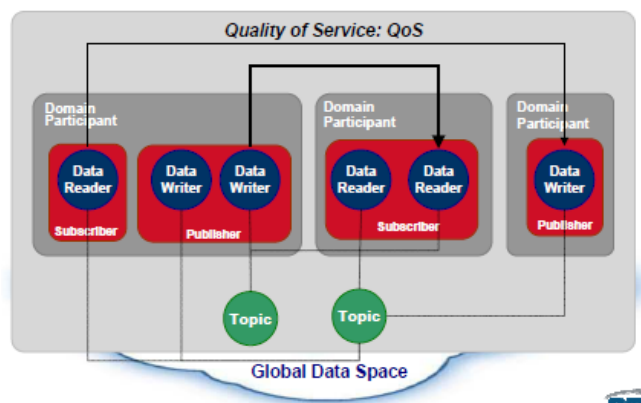


Figure 3 Architecture de Data distribution Service (DDS)

La spécification DDS offre deux niveaux d'interface (figure 4), l'un de bas niveau 'Data-Centric Publish-Subscribe' (DCPS) hautement configurable, étroitement lié aux données et riche de nombreux paramètres QoS pour déterminer le comportement requis; un niveau plus haut 'Data Local Reconstruction Layer' (DLRL) qui offre une approche amplement simplifiée à l'utilisateur et permet une meilleure intégration au niveau applicatif.

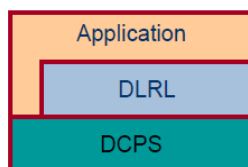


Figure 4 : modèle en couche de DDS

DDS présente un espace de données virtuel global (figure 3) où les applications peuvent partager les informations en consommant et/ou en produisant des objets de données adressés au moyen de l'identificateur du domaine (Domain ID), du nom de l'application (Topic) et d'une clef (Key).

DDS est décrit par un modèle centré sur les données qui permet le découplage complet entre les participants, favorisant ainsi l'auto configuration et l'auto découverte des nœuds. DDS fournit la possibilité de spécifier pour chaque type de données plusieurs paramètres de QoS (débit, latence, durée de validité,...) qui permettent au concepteur de l'application de réaliser une application distribuée basée sur le besoin et la disponibilité de chaque type de données.

4.1.Data-Centric Publish-Subscribe (DCPS)

La couche DCPS est composée de 5 modules (figure 5): infrastructure, Topic-Definition, Publication, Suscription et Domain : Le module infrastructure contient les classes DCPSntity, QoSPolicy, Listener, Condition and WaitSet ; Le module Topic-Definition comporte les classes Topic et TopicListener et ce que l'application requiert pour définir les types de données, créer les Topic et attacher les paramètres de QoS à ces données; Le module Publication contient les classes Publisher, DataWriter et PublisherListener. Le module Suscription comprend les classes

Subscriber, DataReader et SubscriberListener. Le module Domain (n'est représenté sur la figure 5) renferme les classes DomainParticipantFactory et DomainParticipant.

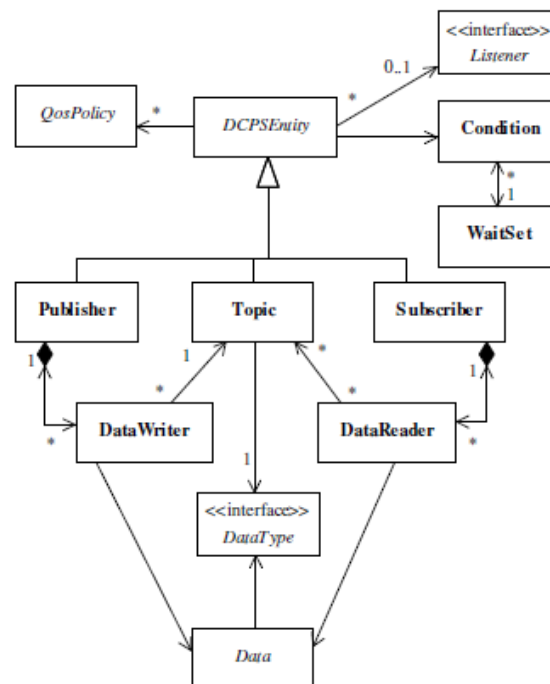


Figure 5 Description UML de DCPS

4.2.Data Local Reconstruction Layer (DLRL)

La couche DLRL est la couche optionnelle et non encore complètement spécifiée dans le standard, permet aux développeurs d'employer les dispositifs fondamentaux de DCPS. DLRL est la partie supérieure de DDS et qui résume la façon à laquelle une application peut se connecter à DCPS. DLRL est une couche facultative selon les spécifications de DDS.

5 Implémentation de HLA et DDS

Dans cette partie, nous illustrons une implémentation de HLA et DDS sur des réseaux de communication hétérogènes. Nous utilisons pour HLA une version Lite de pRTI de la société pitch [12]. Il s'agit d'une application de communication par envoie de message hello.

L'exemple présenté dans les figures 6, 7 et 8 décrit la manière à laquelle deux applications peuvent communiquer selon le modèle producteur/consommateur, formant ainsi une Fédération en cours d'exécution.

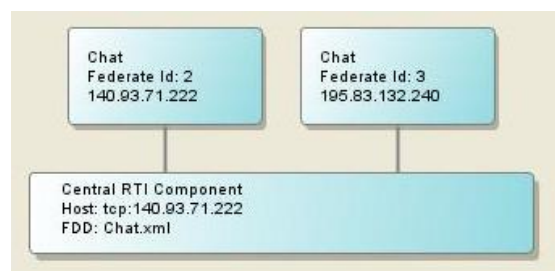


Figure 6 Exemple d'exécution de fédération sous HLA

```

Enter the IP address of the CRC host [localhost]: 140.93.71.222
Enter your name: lakram has joined]
> bonjour akram
Type messages you want to send. To exit, type . <ENTER>
> akram: bonjour ihssen
> ceci est test pour voir comment rejoindre la federation
> moi je suis federe
> Je suis producteur de message
> akram: moi je suis le composant centrale
> akram: vous avez reussi à rejoindre la federation
> akram: moi aussi je suis producteur de ce message
> akram: et je suis consommateur de votre message
> akram: merci
> Message reçu: je suis aussi consommateur de votre message
> tout va bien
>

```

Figure 7 Fédéré géré au niveau d'une machine productrice

```

Enter the IP address of the CRC host [localhost]: 140.93.71.222
Enter your name: AKRAM
Type messages you want to send. To exit, type . <ENTER>
> IHSEM has joined]
> IHSEM: BONJOUR AKRAM
> IHSEM: C IHSEM
> bonjour tout le monde
> ici c le crc
> IHSEM: ICI C LRC

```

Figure 8 Fédéré géré au niveau d'une machine productrice

Le même exemple est développé sous le standard DDS. Une machine productrice envoie un message en Broadcast sur le réseau pour rejoindre le consommateur situé sur une autre machine distante. Les résultats sont présentés sur la figure 9.

<pre> ca. Invite de commandes - hello_pub.exe wrote a sample wrote a sample wrote a sample wrote a sample </pre> <p>a)</p>	<pre> connect() error: No error select error: No error select error: No error New neighbor: hcast: 172.18.106.215:3113 server tcp: 172.18.106.215:59676 </pre> <p>b)</p>	<pre> ca. Invite de commandes - hello_sub.exe msg 0 = Hello WORLD! msg 0 = Hello WORLD! msg 0 = Hello WORLD! </pre> <p>c)</p>
--	--	---

Figure 9 modèle producteur consommateur selon DDS : a) Producteur de la variable (msg) ; b) Tunnel de la communication entre les entités) le consommateur reçoit la variable

6 Conclusion:

Cet article a visé la description du besoin d'une architecture de communication pour la simulation distribuée interactive. Il décrit une vue d'ensemble de l'évolution des technologies qui ont été proposées. Il s'avère que les solutions classiques telles que l'égal à égal (SIMNET, DIS), le client/serveurs (ALSP, CORBA) et les modèles producteur/consommateur classiques (HLA) sont très limités en termes de scalabilité, de fiabilité de transmission et de communication temps-réel. Bien qu'ils ne soient pas encore matures, DDS est une architecture ouverte centrée sur les données. Nos travaux futurs focalisent sur la coopération entre HLA et DDS pour exploiter les caractéristiques de ces deux standards afin d'avoir des simulations interopérables et réutilisables et d'implémenter une communication temps réel basé sur le besoin des participants en QoS. L'évolution de ces deux standards pourra fournir des bases solides pour les simulations temps réel critique sur des réseaux WAN à qualité de service .

Références

- [1] D.C. Miller, J.A. Thorpe “SIMNET: the advent of simulator networking”. Proceedings of the IEEE. Aug 1995. pp: 1114-1123. Lincoln Lab., MIT, Lexington, MA
- [2] IEEE 1278.1A-1998 - Standard for Distributed Interactive Simulation - Application protocols
- [3] H. Du, H Qi, X Wang. A Parallel Independent Component Analysis Algorithm. Proceedings of the 12th International Conference on Parallel and Distributed Systems - Volume 1-pp 151-160
- [4] F. Garcia. Conception, Implémentation et Mesures des Performances d’une Architecture de Communication de bout en bout à QoS garantie en environnement internet nouvelle génération. Laas-CNRS 2002
- [5] J.S. Dahmann, R.M. Fujimoto , R.M. Weatherly. The DoD High Level Architecture: an update. appears in: Simulation Conference Proceedings, 1998. Winter. Defense Modeling & Simulation Office, Alexandria, VA; 13-16 Dec 1998. pp 797-804 vol.1.
- [6] IEEE, P 1516. Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Framework and Rules, February 2000.
- [7] IEEE. P 1516.2, Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - HLA Object Model Template (OMT), February 2000.
- [8] IEEE, P 1516. Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Interface specification, February 2000.
- [9] A L. Wilson, M.W. Richard.” The Aggregate Level Simulation Protocol: An Evolving System”. Proceedings of the 1994 Winter Simulation Conference.
- [10] Gerardo Pardo-Castellote. “OMG Data-Distribution Service: Architectural Overview”. Real-Time Innovations, Inc.
- [11] Rajive Joshi, Gerardo-Pardo Castellote.” A Comparison and Mapping of Data Distribution Service and High-Level Architecture Real-Time Innovations, Inc. 3975 Freedom Circle, Santa Clara, CA 95054
- [12] www.pitch.com
- [13] www.rti.com
- [14] www.omg.org/technology/documents/dds_spec_catalog.htm