



Open Archive Toulouse Archive Ouverte (OATAO)

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible.

This is an author-deposited version published in: <http://oatao.univ-toulouse.fr/>
Eprints ID: 11468

To cite this version: Cardoso, Janette and Chaudemar, Jean-Charles and Hamez, Alexandre and Hugues, Jérôme and Siron, Pierre *PRISE : une plate-forme de simulation distribuée pour l'ingénierie des systèmes embarqués*. (2014) *Génie Logiciel* (n° 108). pp. 29-34. ISSN 1265-1397

Any correspondence concerning this service should be sent to the repository administrator: staff-oatao@inp-toulouse.fr

PRISE : une plate-forme de simulation distribuée pour l'ingénierie des systèmes embarqués

JANETTE CARDOSO, JEAN-CHARLES CHAUDEMAR, ALEXANDRE HAMEZ, JÉRÔME HUGUES ET PIERRE SIRON

Résumé

Dans cet article, nous présentons PRISE, une Plate-forme pour la Recherche en Ingénierie des Systèmes Embarqués, développée à l'ISAE, l'Institut Supérieur de l'Aéronautique et de l'Espace. PRISE est bâtie sur des technologies modernes axées sur l'ingénierie des systèmes aéronautiques et spatiaux. Ses objectifs sont, d'une part, de modéliser et de valider les aspects importants des systèmes critiques embarqués temps réel, d'autre part de valider de nouvelles approches scientifiques. Pour atteindre ces objectifs, PRISE combine un ensemble d'outils, de méthodes en ingénierie système (modélisation, validation, génération de code) et une plate-forme d'exécution basée sur la simulation distribuée et sur un ensemble de systèmes embarqués réels.

Mots clés

Plate-forme, systèmes embarqués, simulation distribuée, modélisation, validation, contraintes temps réel, systèmes cyber-physiques ■

1. INTRODUCTION

Le développement de systèmes embarqués implique la combinaison de plusieurs domaines d'expertise. En tant qu'établissement d'enseignement et de recherche, ISAE mène des activités afin de développer sa propre expertise dans les systèmes embarqués pour améliorer ses programmes d'enseignement et ses résultats de recherche. Le projet PRISE est un vecteur de recherche pour satisfaire deux objectifs complémentaires afin de fournir :

- une plate-forme de support à l'enseignement sur les systèmes embarqués à logiciels prépondérants,
- un banc de test pour des activités de recherche combinant modélisation, analyse, simulation de systèmes complexes.

Dans cet article, nous rapportons notre expérience dans la conception d'une plate-forme en présentant les besoins initiaux, l'implémentation qui en a découlé et les orientations futures.

2. CONCEPT DE PRISE

Le projet PRISE (Plate-forme pour la Recherche en Ingénierie des Systèmes Embarqués) est né en 2007 d'une initiative interne à l'ISAE (Institut Supérieur de l'Aéronautique et de l'Espace) de construction d'une plate-forme pour servir de support aux bonnes pratiques d'ingénierie dans le domaine spatial et le domaine aéronautique. En effet, en tant que grande école d'ingénieurs spécialisée dans ces domaines, l'ISAE s'est résolument orientée vers le transfert de technologies de haut niveau de la recherche à l'industrie à travers des projets de R&D. Ainsi, PRISE a été définie au sein du département de Mathématique, Informatique, Automatique (DMIA) comme un support expérimental pour des systèmes à logiciels prépondérants couvrant les activités d'ingénierie depuis la conceptualisation par des exigences de haut niveau à la simulation, l'implémentation et l'analyse (validation, performances, etc.).

De manière générale, le but de la plate-forme PRISE est d'être un support pédagogique et de recherche suffisamment représentatif des diverses activités d'ingénierie de systèmes complexes et critiques dans les domaines précités.

2.1 Besoins relatifs à PRISE

Dans cette section, nous analysons les besoins liés à la conception de PRISE en fournissant un ensemble, non exhaustif, d'exigences utilisateur.

Les besoins primaires concernent une plate-forme de support, à la fois logicielle et matérielle, à visée pédagogique et utile pour la recherche. À partir de ces besoins, ont été déduites les trois exigences fonctionnelles suivantes :

- La plate-forme doit servir de support à différentes phases du cycle classique en V de développement de systèmes embarqués, structurant les activités d'enseignement et de recherche du DMIA.

- La plate-forme doit intégrer des outils de support des activités d'ingénierie système dans divers domaines, tel que l'ingénierie des exigences (Reqtify, DOORS, KAOS, etc.), la modélisation (SysML, UML, AADL), les langages de programmation (C, ADA, SIMULINK), les systèmes d'exploitation temps réel (RTEMS, VxWorks, etc.) et les outils d'analyse (*model-checking*, analyse d'ordonnancement).
- La plate-forme doit couvrir le domaine spatial et le domaine aéronautique.

► 2.2 Implémentation de PRISE

Pour répondre à ces exigences, PRISE a été conçue en deux parties. La première répond aux besoins d'ingénierie tels que la modélisation, vérification, etc., tandis que la deuxième permet d'exécuter les modèles.

La première partie est purement logicielle puisqu'elle concentre tous les outils réunis au sein de PRISE. Ces outils ont pour objectif de permettre la modélisation de systèmes embarqués, leurs validations via la simulation ou des techniques de vérification formelles (*model-checking*). Une fois spécifiés (à l'aide des standards OMG SysML et SAE AS2-C AADL) et validés, les modèles peuvent être générés automatiquement (conformément aux standards industriels ECSS-E-40B, RT-POSIX, etc.) vers du code source dans le but d'être exécutés sur la seconde partie de la plate-forme.

En plus d'utiliser des outils commerciaux tels que SCADE et Matlab-Simulink, l'ISAE développe et maintient des outils libres avec des partenaires académiques : TASTE/Ocarina pour la génération des modèles AADL en partenariat avec l'ESA [6] ; TTool pour la vérification de modèles SysML avec TelecomParisTech [1] ; CERTI avec l'ONERA pour la simulation distribuée [8] ; Ptolemy pour la modélisation et la simulation de systèmes hétérogènes [11].

La seconde partie permet l'exécution des modèles en les interprétant directement (par exemple en utilisant Simulink ou Ptolemy) ou en exécutant le code source correspondant généré. Dans tous les cas, ces exécutions se déroulent soit sur des systèmes temps réel (RTEMS, Xenomai, FreeRTOS, VxWorks) déployés sur du matériel embarqué (PowerPC, LEON, CAN, SpaceWire) ; soit au sein d'un simulateur distribué présenté plus en détail dans les sections suivantes.

► 2.3 Usage de PRISE

PRISE est utilisée dans deux cadres au sein de l'ISAE :

- Enseignement : les étudiants des cursus ingénieur et master peuvent avoir une vision plus claire des fonctions particulières dans un système embarqué à l'aide de la simulation ou de la modélisation.
- Recherche : des chercheurs et de doctorants ISAE [5,9,4] testent de nouvelles fonctions pour intégrer ces

outils dans la plate-forme. Cela favorise la définition et l'amélioration des outils pour une plus grande facilité d'utilisation, pour la couverture des fonctions étudiées ainsi que pour l'amélioration de la maturité et du niveau TRL (*Technology Readiness Level*) de PRISE.

3. SIMULATION DISTRIBUÉE DE SYSTÈMES RÉPARTIS

Dans cette section, nous présentons une des facettes de la plate-forme PRISE : SDSE (Simulation Distribuée de Systèmes Embarqués), la simulation d'un système avionique complet avec un niveau de complexité réduit. Nous présentons ici le contexte de ce projet ainsi que les éléments matériels et logiciels utilisés au sein de PRISE pour réaliser ce projet.

► 3.1 Simulation et application distribuées

La simulation est une technique mature utilisée pour l'entraînement, l'évaluation de performances et la recherche. Un simulateur de vol recrée le vol d'un avion dans son environnement, modélise son comportement dynamique selon les lois de l'aérodynamique, les lois de commande régissant son vol, la poussée et la force de gravité, tout ceci selon des caractéristiques environnementales extérieures (densité de l'air, vent, turbulences, etc.).

Dans cette optique, notre simulateur de vol comprend plusieurs composants. L'essentiel étant la description mathématique de l'avion et de son environnement. Plus le modèle est précis, plus la simulation est réaliste. Un ordinateur avec un OS temps réel calcule ce modèle. La simulation peut-être complétée d'entrées (joystick, etc.), d'écrans d'affichage, d'environnements reproduisant le cockpit, etc.

Nous estimons que les simulations distribuées se comportent comme les applications réelles, car elles ont en commun :

- les mêmes exigences de modularité ;
- des codes similaires ;
- l'usage d'un intergiciel et de plates-formes spécifiques d'exécution ;
- des contraintes temps réel.

► 3.2 Standard d'interopérabilité

Le calcul distribué offre une solution de haute performance grâce aux avancées dans les technologies réseau. Différents programmes situés sur plusieurs ordinateurs interagissent dans un but commun, ici la simulation. De plus, les concepteurs et développeurs d'applications réparties doivent faire face à l'hétérogénéité matérielle et logicielle (système d'exploitation, protocoles de communication, etc.).

Le développement de standards d'intergiciels permet

de répondre à ces problèmes. Un intergiciel est une couche logicielle faisant office d'intermédiaire entre des processus répartis. Ce logiciel fait donc partie du domaine de l'interopérabilité : c'est un logiciel de connectivité qui permet l'exécution de plusieurs applications interagissant sur un ou plusieurs ordinateurs liés. Cette interopérabilité permet la construction de systèmes plus complexes, la réutilisation de composants et accroît la flexibilité en facilitant le remplacement d'un composant par un nouveau.

Nous estimons que le choix d'un standard de simulation distribuée et de son intergiciel associé est une exigence importante afin d'obtenir une simulation temps réel fiable, valide et pouvant passer à l'échelle. Ce choix impose les systèmes d'exploitation, les langages et le matériel pouvant être utilisés avec l'intergiciel choisi. Notre choix s'est porté sur le standard HLA (*High Level Architecture*) [13].

HLA est un standard de simulation distribuée à événements discrets, initialement proposé par le département de la défense des États-Unis, puis accepté en tant que standard IEEE (IEEE 1516). La réutilisabilité et l'interopérabilité sont des objectifs fondamentaux d'HLA. Des simulations complexes (« fédérations ») sont réalisées en interconnectant des simulateurs indépendants (« fédérés »). Ces fédérés se connectent à une infrastructure d'exécution (*Runtime Infrastructure*, RTI). Les fédérés gèrent et partagent des objets et échangent des informations, se coordonnent temporellement en suivant les spécifications HLA.

Les spécifications HLA consistent en un ensemble de règles concernant les fédérés et fédérations, un modèle objet pour les fédérés (particulièrement pour les objets publics et les interactions), ainsi qu'un ensemble de services requis pour gérer les objets et interactions. Certains de ces services doivent être fournis par les fédérés au RTI, les autres étant fournis par le RTI aux fédérés. Les fédérés publient et souscrivent à des classes d'objets et d'interactions afin de créer et échanger des informations. Une autre caractéristique est le service de gestion de temps permettant des simulations déterministes et reproductibles.

► 3.3 Plate-forme logicielle et matérielle

Un choix important a été fait pour utiliser des logiciels libres et du matériel générique dans PRISE.

La plate-forme avionique PRISE utilise 4 ordinateurs temps réel Opteron avec 6 cœurs, 2 stations graphiques avec processeurs Intel Xeon et GPGPU haute performance, utilisant le système d'exploitation Linux Red Hawk et un noyau conforme au standard POSIX. Une technologie matérielle d'horloge distribuée permet la distribution d'une même référence d'horloge à chaque nœud. Ces ordinateurs sont reliés par un réseau spécialisé avec un commutateur Ethernet Gigabit. Deux systèmes physiques avec manette de pilotage palonnier-pédale font aussi partie de cette plate-forme.

CERTI [8] est l'intergiciel multiplate-forme (Linux et Windows) choisi pour implémenter le standard HLA. Étant un logiciel libre, il permet de maîtriser l'implémentation du RTI et ainsi de faciliter l'intégration de modifications du code source. De plus, CERTI présente des bonnes réponses aux contraintes temps réel dans une étude ONERA/CNES de vol de satellites en formations [4,7].

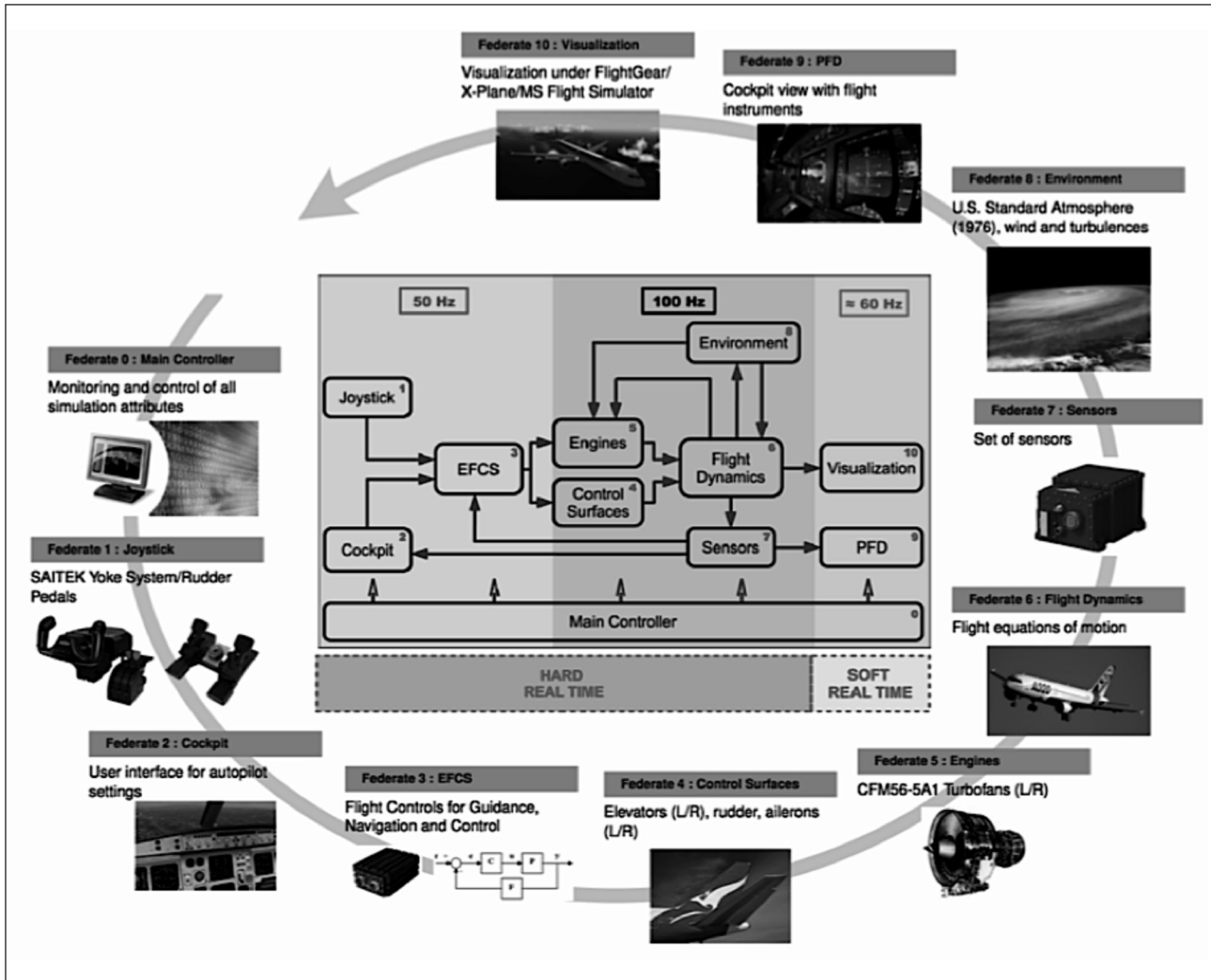
► 3.4 Fédération avionique SDSE

Onze fédérés représentent chacun une partie de l'avion (avion naturel, moteur, gouvernes) et communiquent en utilisant CERTI pour assurer l'interopérabilité (figure 1) :

- Fédéré 0 : Surveillance et contrôle tous les paramètres de la simulation
- Fédéré 1 *Joystick* : représente les différents organes de pilotage avec un ensemble volant/ manette de gaz/ palonnier
- Fédéré 2 *Cockpit* : interface utilisateur via un écran tactile
- Fédéré 3 *EFCS (Electronic Flight Control System)* : contrôle de l'aéronef et pilote automatique
- Fédéré 4 *Control Surfaces* : ailerons à gauche et à droite, des gouvernes de profondeur à gauche et à droite et une gouverne de direction.
- Fédéré 5 *Engines* : deux turboréacteurs (gauche et droite) CFM56-5A1
- Fédéré 6 *Flight Dynamics* : cœur du simulateur de vol (équations de mécanique de vol)
- Fédéré 7 *Sensors* : vingt capteurs tels que IRU, IMU, avec des phénomènes tels que retard pur, bruit, quantification
- Fédéré 8 *Environnement* : modèle de l'atmosphère US (1976) avec différentes turbulences (Dryden, Von Karman, windshear)
- Fédéré 9 (*Primary Flight Display*) : cockpit avec les instruments de vol
- Fédéré 10 *Visualization* : environnement virtuel (FlightGear, Microsoft Flight Simulator et X-Plane)

► 3.5 Contraintes temps réel

Les techniques modernes de simulation de vol nécessitent très souvent des calculs complexes et coûteux demandant une grande puissance de calcul. Or les simulateurs de vol requièrent souvent de respecter des échéances temporelles. C'est la problématique principale des systèmes temps réel dont la validité ne dépend pas seulement du résultat des calculs, mais aussi du temps auquel ces résultats sont construits [12].



▲ Figure 1 : SDSE (Simulation Distribuée de Systèmes Embarqués)

Les systèmes temps réel sont classés en deux catégories, en se basant sur la nature des échéances temporelles. On parle ainsi de systèmes temps réel dur (par exemple, contrôles de vol), pour lesquels le non-respect d'une échéance peut entraîner de graves conséquences, et de systèmes temps réel mou (par exemple un affichage), pour lesquels le non-respect d'une échéance entraîne l'invalidité des résultats produits en retard.

Le calcul du temps d'exécution le plus défavorable (*Worst Case Execution Time*, WCET) est la clef pour un ordonnancement réussi. Dans notre cas, ce calcul doit prendre en compte les calculs effectués par chaque fédéré. Les WCET ont été estimés pour chaque fédéré, ainsi que les temps de transmission les plus défavorables (*Worst Case Transit Time*, WCTT) pour chaque message de CERTI.

La version actuelle de CERTI met en œuvre les services d'HLA en offrant des garanties temps réel [3]. Globalement, l'architecture de simulation distribuée et l'application de simulation sont analysées afin de trouver et valider *a priori* leur ordonnancement [5].

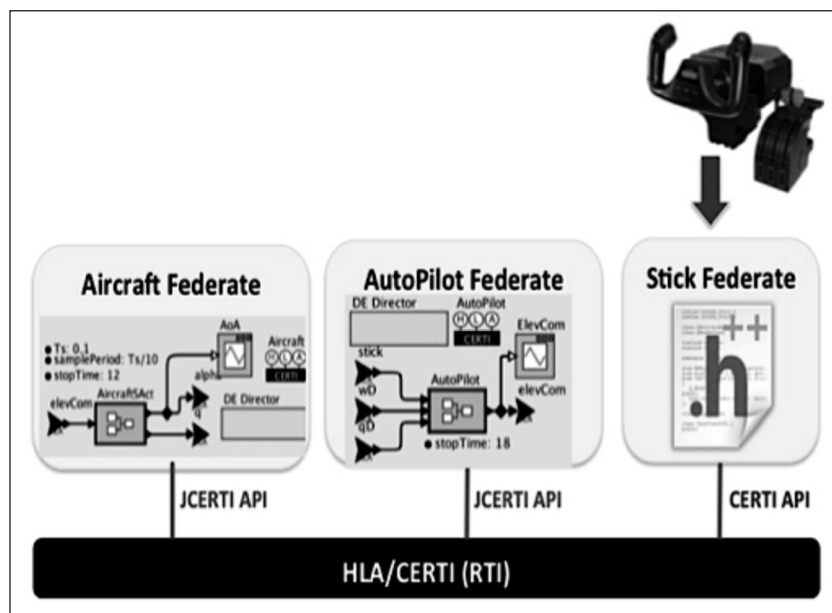
► 3.6 Environnement de co-simulation Ptolemy et HLA/CERTI

Les fédérés 4 à 8 (figure 1) sont des simulateurs de modèles mathématiques continus représentant les systèmes physiques ; le fédéré 3, EFCS, met en œuvre le code du contrôleur. Ces 6 fédérés sont codés en C++. Le fédéré Joystick implémente, en C++, l'interface avec les organes de pilotage. HLA/CERTI permet des simulations distribuées interopérables et l'intégration des éléments matériels comme les organes de pilotage. Grâce à l'interopérabilité garantie par le standard HLA, ces fédérés peuvent être utilisés dans d'autres fédérations et interagir avec d'autres fédérés tant que les variables d'entrée et de sortie, et tant que les avancées temporelles sont compatibles.

Pour permettre la modélisation hiérarchique de modèles hétérogènes, nous avons développé un environnement de co-simulation [9] permettant à l'environnement logiciel libre Ptolemy [11] de réaliser des simulations distribuées en utilisant CERTI. Ptolemy est particulièrement bien adapté à la modélisation des systèmes cyber-physiques en utilisant plusieurs modèles de calcul : continu, à événements discrets, data-flow synchrone, etc. Ces modèles de calcul

différents peuvent être imbriqués dans des modèles hiérarchiques. L'environnement développé permet de connecter directement un modèle Ptolemy à une fédération HLA, sans avoir à récrire le code, simplement en déployant des blocs prédéfinis pour les communications de données.

Une étude complète d'un système de commande de vol longitudinal, plus simple que celle modélisée dans SDSE, a été réalisée dans [10]. Cette fédération F14 (figure 2) est constituée de deux fédérés Ptolemy -- Aircraft (pour le modèle physique du F14), AutoPilot (pour le contrôleur) -- et le fédéré Joystick (utilisé dans la Fédération SDSE) de façon à utiliser un vrai manche. Ainsi, selon l'action du pilote, le fédéré AutoPilot calcule la commande de la gouverne et le fédéré Aircraft réagit en affichant le bon angle d'attaque. Cette étude a permis de montrer l'interopérabilité entre des fédérés Ptolemy et des fédérés écrits directement en C++.



▲ Figure 2 : Fédération F14 avec un fédéré Joystick

4. CONCLUSION ET PERSPECTIVES

Nous avons présenté PRISE, la plate-forme pour la Recherche en Ingénierie des Systèmes Embarqués. Elle fournit un support pédagogique et de recherche suffisamment représentatif des diverses activités d'ingénierie des systèmes complexes et critiques dans le domaine aérospatial, couvrant les différentes étapes du cycle en V. Les activités au sein de cette plate-forme s'articulent sur deux axes : la plate-forme d'atelier logiciel et la plate-forme d'exécution, les résultats du premier pouvant être utilisés dans le second. Par exemple, les travaux réalisés permettant d'intégrer des modèles Ptolemy dans une fédération HLA ont pu être réutilisés directement dans SDSE.

La fédération avionique SDSE illustre bien les activités d'enseignement et de recherche au sein de PRISE. Elle

permet aux étudiants une meilleure compréhension des défis techniques et technologiques des systèmes avioniques complexes, en allant du génie logiciel à la validation et vérification en passant par l'évaluation de lois de commande de bout en bout, l'évaluation de nouveaux capteurs, interfaces, moteurs, etc. Soulignons les aspects multidisciplinaire et intégrateur de cette plate-forme. Du point de vue de la recherche, PRISE permet de valider les contributions des chercheurs du DMIA/ISAE dans le domaine de la simulation (projet CERTI en collaboration avec l'ONERA, projet Ptolemy avec l'Université de Berkeley), la modélisation et analyse de systèmes, l'extension à d'autres outils à l'aide du standard FMI (*Functional Mockup Interface*) [2], ainsi que la génération de code (projets TTool et Ocarina).

Les perspectives pour PRISE sont multiples. Au niveau de la plate-forme logicielle, il faut continuer et améliorer la production sûre de composants, en particulier pour

des automaticiens depuis Ptolemy ou Simulink, automatiser la production de la fédération depuis un langage de description d'architecture, poursuivre l'étude des aspects temps réel, sûreté de fonctionnement, etc. Au niveau de la plate-forme d'exécution, nous pouvons intégrer et tester de nouveaux composants : différents modèles de turbulence, modèles de drones, instruments, écrans tactiles et tablettes, organes de commande à retour d'effort, processeurs embarqués voire multicœurs, etc.

Nous avons focalisé cette présentation sur la plate-forme d'exécution avionique, une plate-forme d'exécution spatiale est développée en parallèle. Elle illustre les similitudes et les différences entre ces deux domaines au niveau du génie logiciel (normes par exemple) et au niveau de l'architecture d'exécution (processeurs et réseau).

5. BIBLIOGRAPHIE

- [1] L. Apvrille et P. de Saqui-Sannes : *AVATAR/TTool : un environnement en mode libre pour SysML temps réel* ; Génie Logiciel, n° 98, pp. 22-26, septembre 2011, ISSN 1265-1397.
- [2] J.-C. Chaudemar, V. Savicks, M. Butler et J. Colley : *Co-simulation of Event-B and Ptolemy II models via FMI* ; ERTS2 2014 (Embedded Real-Time Software and Systems), Toulouse, 5-7 février 2014.
- [3] J.-B. Chaudron, E. Noulard et P. Siron : *Design and model-checking techniques applied to real-time RTI time management* ; 2011 Spring Simulation Multiconference - SpringSim'11, Boston, États-Unis, 4-8 avril 2011.

- [4] J.-B. Chaudron, D. Saussié, P. Siron et M. Adelantado : *Real-time aircraft simulation using HLA standard* ; IEEE AESS Simulation in Aerospace 2011, Toulouse, 8 juin 2011.
- [5] J.-B. Chaudron : *Architecture de simulation distribuée temps réel* ; Thèse Doctorat, École Doctorale MITT, 25 janvier 2012.
- [6] J. Delange, J. Hugues et P. Dissaux : *Validate implementation correctness using simulation: the TASTE approach* ; ERTS2 2012 (Embedded Real-Time Software and Systems), Toulouse, 1-3 février 2012.
- [7] E. Noulard, B. d'Ausbourg et P. Siron : *Running real-time distributed simulations under Linux and CERTI* ; European Simulation Interoperability Workshop, Edimbourg, 16-19 juin 2008.
- [8] E. Noulard J.-Y. Rousselot et P. Siron : *CERTI: an open source RTI, why and how ?* ; Fall Simulation Interoperability Workshop, 2009.
- [9] G. Lasnier, J. Cardoso, C. Pagetti et P. Siron : *Environnement de coopération de simulation pour la conception de systèmes cyber-physiques* ; MSR13, Brest, 2013.
- [10] G. Lasnier, J. Cardoso, P. Siron, C. Pagetti et P. Derler : *Distributed simulation of heterogeneous and real-time systems* ; 17th IEEE/ACM International Symposium on Distributed Simulation and Real-time Applications, novembre 2013.
- [11] C. Ptolemaeus (Ed.) : *System design, modeling, and simulation using Ptolemy II* ; Ptolemy.org, 2014.
- [12] J. A. Stankovic : *Misconceptions about real-time computing* ; IEEE Computer, vol. 21, n° 10, 1988.
- [13] The Institute of Electrical and Electronics Engineers (IEEE) Computer Society : *IEEE Standard for Modeling and Simulation High Level Architecture (HLA) - Federate Interface Specification* ; septembre 2000.

BIOGRAPHIES

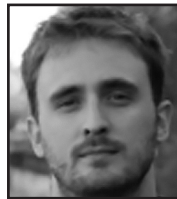


Janette Cardoso est professeure à l'Institut Supérieur de l'Aéronautique et de l'Espace, Toulouse. Elle est ingénieure de l'Université Fédérale de Santa Catarina, Brésil, obtient son doctorat à l'Université de Toulouse en 1990 et son Habilitation à diriger des

recherches en 2007. Ses intérêts de recherche portent sur la modélisation, simulation et implémentation de systèmes distribués, les méthodes formelles pour la conception de logiciels pour les systèmes embarqués utilisant les réseaux de Petri et le traitement formel des informations incomplètes utilisant la théorie de possibilités.



Jean-Charles Chaudemar est enseignant-chercheur au Département de Mathématiques Informatique et Automatique de l'ISAE. Diplômé ISAE (docteur de l'Université de Toulouse/ISAE en 2012 et ingénieur SUPAERO en 1998), il a d'abord travaillé dans les industries aéronautique et automobile. Au sein de l'ISAE, ses activités de recherche s'orientent autour des thèmes de la conception d'architectures de sécurité, la modélisation et la simulation de systèmes critiques complexes. Pour ce qui est de l'enseignement, il est responsable du Mastère Spécialisé en Ingénierie Système.



Alexandre Hamez est ingénieur chercheur au Département Mathématiques Informatiques et Automatique de l'ISAE. Il obtient son doctorat à l'université Paris 6 en 2009. Ses recherches se focalisent sur une technique de vérification formelle, le *model-checking*, basée sur les diagrammes de décision et sur le développement d'outils efficaces dédiés à cette tâche.



Jérôme Hugues est maître de conférences au Département de Mathématiques, Informatique et Automatique de l'ISAE. Diplômé de Télécom ParisTech (thèse, 2005 et ingénieur 2002), ses activités de recherche portent sur la conception de logiciels embarqués. Il est co-responsable de l'approfondissement d'Informatique de la formation ISAE, et du Mastère Spécialisé en Systèmes Embarqués de l'ISAE/INP-ENSEEIH. Il est membre du comité de standardisation SAE AS2-C et contributeur des projets Ocarina et TASTE.



Pierre Siron, ingénieur ENSEEIHT 1980, DEA informatique de l'Université Paul Sabatier 1980, docteur de l'ENSAE (SUPAERO) 1984, HDR 2003, maître de recherche ONERA, Département Traitement de l'Information et Modélisation, puis professeur ERE de

l'ISAE, responsable de l'unité de formation informatique de SUPAERO, Département Mathématiques Informatique Automatique et animateur de l'axe IpSC, Ingénierie pour les Systèmes Critiques. Ses activités de recherche ont concerné dans l'ordre les architectures parallèles et distribuées, la sécurité informatique, la simulation distribuée. Parmi les projets importants en cours, figurent CERTI, la mise en œuvre *open source* d'une infrastructure de simulation distribuée à événements discrets au standard HLA et PRISE, une plate-forme pour la recherche en ingénierie des systèmes embarqués.