Projet ANR-06-SETI-015
RIMEL (Raffinement Incrémental de Modèles événementiELs)
Rapport intermédiaire annuel

Coordonnateur Dominique Méry
LORIA,Université Henri Poincaré Nancy 1

Partenaire 2 Thierry Lecomte
ClearSy

Partenaire 3 Mohamed Mosbah
LABRI, Université Bordeaux I

January 31, 2008 Version at 8:46am

Abstract

Ce document présente une synthèse des résultats obtenus par les partenaires du projet RIMEL; il comprend un rappel des objectifs de ce projet et une analyse des tâches. En particulier, la tâche 1 est à son terme et une description plus détaillée est apportée; les résultats obtenus sont principalement attesté par des publications, des archives de modèles B et des spécifications de logiciels en cours de développement.

1 Introduction au projet

Le projet RIMEL (Raffinement Incrémental de Modèles événementiELs) concerne principalement le raffinement de modèles événementiels et la systématisation de cette technique dans le cadre d'applications ciblées notamment la conception d'algorithmes ou systèmes répartis. La systématisation de cette technique repose sur le développement de schémas conceptuels appelés *patrons de développement* s'appuyant sur une validation par la preuve (patrons de développement prouvés). Le développement du protocole IEEE 1394 a montré les limites du raffinement et conduit à rechercher à intégrer des extensions liées au traitement des aspects probabilistes et des contraintes éventuelles de temps. Le travail est organisé suivant les directions scientifiques suivantes:

1. Théorie du raffinement: raffinement probabiliste, raffinement avec intégration de temps, prise en compte de contrainte de fatalité.
2. Proof-based design patterns: méthodologie d'ingénierie système formelle
3. Self-Healing Systems et Algorithmes Répartis
4. Outils et diffusion

Les motivations de ce projet concernent principalement l'intégration de la preuve mathématique dans le processus de développement de systèmes informatiques pour justifier la confiance et deux domaines d'applications sont exploités pour inférer et valider les apports méthodologiques du raffinement.

Le projet comprend plusieurs tâches identifiées par des livrables et correspondant à des étapes du projet.

- **Deliverable 1 Development of distributed algorithms**: le projet s'appuie sur des études de cas constituées principalement de problèmes de l'algorithmique répartie et l'objectif est de les développer en utilisant la technique du raffinement.
- **Deliverable 2 Integration of timing constraints in an incremental proof-based development**: l'objectif est de mieux comprendre l'introduction d'éléments temporels dans le cadre du raffinement et un certain nombre d'algorithmes répârtis repoent sur des hypothèses temporelles.
- **Deliverable 3 Proof-based design patterns**: Le développement incrémental et prouvé de systèmes répartis posent des problèmes de preuves et certains développements peuvent être rejoués ou réutilisés; la notion de pattern ou patron reste à définir et à concrétiser dans le cadre de nos applications.

- **Deliverable 4 Formal system engineering**: Il s'agit de considérer non seulement des systèmes comme algorithmes mais des systèmes au sens général et de considérer le développement incrémental et prouvé dans ce cadre.
- **Deliverable 5 Probabilistic incremental proof-based developments**: Les aspects temporels sont étudiés dans une tâche précédente et il est aussi important de considérer les systèmes où une loi de probabilité intervient de manière explicite ou implicite.
- **Deliverable 6 Tools**: Cette tâche s'échelonne tout au long de ce projet et vise à fournir des outils pour la méthode de développement.

Chaque livrable correspond à des étapes accomplies du projet; la première année a permis des échanges sur les techniques de vérification (Coq et B) et sur la modélisation des systèmes répartis (Event B). Dans la proposition initiale, les systèmes *self healing* sont visés et les études de cas seront poursuivies dans le cadre de ce type de systèmes. Au-delà des systèmes répartis algorithmiques, nous sommes aussi intéressés par les systèmes répartis intégrant des aspects matériels; le challenge de ce projet est de combiner plusieurs approches complémentaires visant des systèmes de type algorithmique et de type industriel; ainsi, une méthodologie fondée sur le raffinement, la preuve et la représentation des systèmes répartis est au cœur de nos travaux et en constitue un des objectifs. Rappelons que l'intérêt des systèmes algorithmiques est d'être maîtrisés par les partenaires académiques et que ces systèmes constituent un vivier très important.

2 Actions du projet

Dans cette section, nous faisons un bilan intermédiaire de chaque action identifiée et en cours de réalisation. Les réunions d'avancement ont permis d'échanger sur les techniques, modèles et outils utilisés par les différents groupes. Nous donnons une description des actions menées ou qui sont apparues au cours du dernier semestre.

2.1 Action Algorithmes répartis développés incrémentalement et prouvés

Cette action permet aux membres du projet d'échanger sur les algorithmes répartis et sur les techniques de vérification et de conception incrémentale prouvée pour les systèmes événementiels. Cette action aboutit à un livrable qui va reprendre les études de cas développées selon la méthode incrémentale prouvée et qui va aussi situer cette approche par rapport à l'approche suivie par le partenaire LABRI. Il est assez clair que le partenaire LABRI s'appuie sur une vérification *a posteriori* des algorithmes répartis et que cette vérification n'est pas encore effective sur le plan des outils de preuve comme Coq; l'assistant de preuve utilisé est Coq et une partie du travail de ce partenaire est de développer des bibliothèques de théories liées aux graphes et aux algorithmes répartis étudiés. Cependant, l'intérêt de la collaboration réside dans les échanges portant sur les algorithmes répartis et la description des principes de ces algorithmes. Ainsi, les échanges entre le partenaire LORIA et le partenaire LABRI ont permis de développer des algorithmes réputés non-triviaux dans un temps raisonnable.

- L'algorithme de Mazurkiewicz [6, 17] a été exposé par l'équipe du LABRI et redéveloppé dans le cadre de la méthode event B par Dominique Cansell; cet exemple a permis d'exposer la méthode event B à tous les membres du projet et de l'illustrer. Cet exemple était considéré comme très difficile à prouver avec un assistant de preuve et tout l'effort de modélisation incrémentale a permis le succès obtenu. Un rapport technique sur le développement a été rédigé et un article est en préparation pour une publication ultérieure. Le processus de modélisation s'est appuyé sur l'expertise d'Yves Métivier qui a su expliquer, au tableau, les éléments critiques de cet algorithme et sur la maîtrise de la méthode Event B de Dominique Cansell.
- Des algorithmes d'élection de leader [13] ont été développés à partir du développement du IEEE 1394; si l'algorithme IEEE 1394 a été simplement reconstruit et prouvé, les trois autres algorithmes ont été dérivés, imaginés et conçus à partir du développement prouvé du IEEE 1394 en modifiant certains choix. L'anecdote est assez intéressante car l'une des nouvelles solutions a été obtenue à la suite d'une erreur de modélisation mais à la suite de choix conduisant à une solution correcte. Le développement des modèles constitue donc une structure de raisonnement très importante et réutilisable. Une étude [8] récente de Dominique Cansell a répondu à un problème proposé par le partenaire LABRI et a conduit à un développement complet.
- Le problème de la détection de terminaison est modélisé dans le document écrit par Dominique Cansell [7]; cet algorithme est connu sous le nom de Szymanski, Shi et Prywes. Ce problème a été exposé par le partenaire LABRI et le document initial est en cours de complément.

- Une étude de cas a été menée par Nazim Benaissa [5] et il s'agissait de l'étude des systèmes communicants par transactions. Ainsi, le système MONDEX a été utilisé pour évaluer les éléments méthodologiques associés à ce type de système. Cette étude a permis de mettre en évidence des patrons de modélisation réutilisables pour des applications intégrant des transactions. Ainsi, une étude est en cours et porte sur les protocoles cryptographiques.

Nous avons limité ce livrable à ces quelques études de cas qui ont permis de bien comprendre le savoir-faire des uns et des autres au niveau des algorithmes répartis. Si le partenaire LORIA reste plus familier avec le développement incrémental, il est assez clair que la comparaison des approches notamment avec des outils comme VISIDIA [19] est très prometteuse. Les deux partenaires de cette tâche poursuivront le dévelopement d'études de cas et échangeront des expériences pédagogiques en lien avec les cours donnés en master à Nancy et à Bordeaux. Cet échange est important pour étudier le lien entre le plate-forme RODIN [23] et l'environnement VISIDIA [19] D'autres problèmes liés à des questions assez habituelles en informatique comme les agents mobiles seront probablement analysées par les deux groupes. Le concept d'agents mobiles[1, 16] a été développé, afin de pouvoir résoudre des problèmes dans des environnements hétérogènes et dynamiques. Dans de tels systèmes, les différents nœuds du réseau sont passifs et ce sont des agents mobiles se déplaçant de nœud en nœud qui sont en charge de l'exécution de l'algorithme. Nous devons considérer un modèle assez général de systèmes à agents mobiles et montrer qu'un tel système à agents mobiles à la même puissance de calcul qu'un système distribué où les processus communiquent par échange de messages si les graphes sous-jacents sont identiques. Un corollaire intéressant de ce résultat est une caractérisation des systèmes à agents mobiles où on peut résoudre le problème du rendez-vous. Nous avons également étudié le problème du rendez-vous d'agents mobiles dans des systèmes où les liens de communications et les places du réseau peuvent être défaillants, i.e., tout agent passant par un lien ou une place défaillant est immédiatement détruit. Nous avons caractérisé les systèmes à agents mobiles où on peut résoudre le problème du rendez-vous dans ce cadre et on a présenté des algorithmes optimaux permettant de résoudre ce problème lorsque c'était possible. Enfin, la classe des systèmes auto-stabilisants est intéressante comme sujet d'étude et cela constituera une étudee à mener pour une tâche nouvelle.

Perspectives: Cette action mène au livrable [3] **Development of distributed algorithms**; il présente une collection d'études de cas, afin de mieux comprendre les algorithmes répartis et leur conception mais aussi de dégager des patrons liés à ces algorithmes.

2.2 Action Intégration du temps dans le raffinement

Le temps joue un rôle fondamental dans les algorithmes répartis et nous étudions dans cette action l'introduction du temps au cours du développement incrémental. Des résultats préliminaires ont été donnés et publiés [15] et plus récemment nous [21] avons publié des éléments étendant le modèle event B avec du temps.

Perspectives: Cette action se focalise sur le temps dans les systèmes répartis mais au travers du raffinement et il convient de développer des solutions réparties en prenant soin de comprendre comment des patrons de développement peuvent aider dans cette démarche. Un livrable fera le point sur les éléments apportés et un article de revue est en cours de préparation.

2.3 Action Ingénierie système formelle

Cette action est aussi fondée sur le développement d'études de cas mais ces études de cas sont liées à des systèmes utilisant des algorithmes ou du logiciel. Nous avons plusieurs résultats intermédiaires:

- le développement de systèmes de vote électronique et leur analyse [10–12].
- l'étude du contrôle d'accès dans le cadre du raffinement [4]
- l'étude de patrons de développement [20].
- le développement de programmes séquentiels structurés [14] a mis en évidence un patron de développement que nous somems en train de mettre en œuvre dans le cadre de la plate-forme RODIN.

Perspectives Il s'agit ici aussi de développer des patrons spécifiques mais cela repose sur des études de cas. Nous allons essayer d'identifier quelques patrons à partir des étdues de cas sur les algorithmes répartis. L'expérimentation du patron de conception de programems séquentiels structurés sera mise à profit pour trouver une expression de ces patrons.

2.4 Action Outils

Cette action vise à réaliser des outils dans le cadre du développement prouvé; elle aura sans doute comme support la plate-forme RODIN qui supporte la notation event B et qui est ouverte au développement d'applications dans le cadre de l'atelier ECLIPSE. Deux outils sont donc en cours de réalisation:

- le développement d'un outil de raffinement automatique par la société ClearSy et un document présente les spécifications acceptées par des partenaires industriels du partenaire ClearSy.
- l'intégration de VISIDIA dans le cadre de la notation event B et de RODIN et un ingénieur a visité le partenaire LORIA pour acquérir les compétences nécessaires en B.
- un stagiaire tunisien Internship INRIA va mettre en œuvre le patron des programmes séquentiels structurés.

Perspectives Cette action enrichira les applications disponibles sur le plate-forme RODIN notamment et VISIDIA. Il en va de même pour l'outil de raffinement automatique BART.

References

[1] Shehla Abbas, Mohamed Mosbah, and Akka Zemmari. Collecte d'informations par des agents mobiles. In *NOuvelles TEchnologies de la REpartition, NOTERE 2007*. Hermès - Lavoisier, 2007.

[2] Shehla Abbas, Mohamed Mosbah, and Akka Zemmari. Merging time of random mobile agents. In *International Conference on Dynamics in Logistics 2007 (LDIC 2007)*, Lect. Notes in Comp. Sciences. Springer-Verlag, 2007. To appear.

[3] Projet ANR-RIMEL. Développement d'algorithmes répartis. Rimel, LORIA, Janvier/Février 2008.

[4] Nazim Benaïssa, Dominique Cansell, and Dominique Méry. Integration of security policy into system modeling. In *B*, pages 232–247, 2007.

[5] Nazim Benaïssa and Dominique Méry. Développement incrémental prouvé de systèmes répartis : le cas mondex. Rimel, LORIA, January 2008.

[6] Dominique Cansell. Modélisation incrémentale de l'algorithme de mazurkiewicz. Présentation orale des modèles event B, 2007.

[7] Dominique Cansell. Développement incrémental et preuve de l'algorithme de Szymanski, Shi et Prywes. Rimel, LORIA, January 2008.

[8] Dominique Cansell. Développement incrémental et preuve de l'algorithme d'élection du leader dans un 2-arbre. Rimel, LORIA, January 2008.

[9] Dominique Cansell. Développement incrémental et preuve de l'algorithme d'énumération de mazurkiewicz. Rimel, LORIA, January 2008.

[10] Dominique Cansell, J. Paul Gibson, and Dominique Méry. Refinement: A constructive approach to formal software design for a secure e-voting interface. *Electronic Notes in Theoretical Computer Science*, Volume 183:39–55, July 2007. Proceedings of the First International Workshop on Formal Methods for Interactive Systems (FMIS 2006).

[11] Dominique Cansell, Paul Gibson, and Dominique Méry. Formal verification of *tamper-evident* storage for e-voting. In *(SEFM 2007 5th IEEE International Conference on Software Engineering and Formal Methods*.

[12] Dominique Cansell, Paul Gibson, and Dominique Méry. Refinement: A constructive approach to formal software design for a secure e-voting interface. *Electr. Notes Theor. Comput. Sci.*, 183, 2007.

[13] Dominique Cansell and Dominique Méry. *Festschrift Egon Börger*, chapter Designing old and new distributed algorithms by replaying an incremental proof-based development. Springer, 2007.

[14] Dominique Cansell and Dominique Méry. Proved-patterns-based development for structured programs. In *CSR*, pages 104–114, 2007.

[15] Dominique Cansell, Dominique Méry, and Joris Rehm. Time constraint patterns for event b development. In *B*, pages 140–154, 2007.

[16] Jérémie Chalopin, Emmanuel Godard, Yves Métivier, and Rodrigue Ossamy. Mobile agent algorithms versus message passing algorithms. In *OPODIS*, pages 187–201, 2006.

[17] Jérémie Chalopin and Yves Métivier. An efficient message passing election algorithm based on mazurkiewicz's algorithm. *Fundamenta Informaticae*, 80:221–246, 2007.

[18] Jérémie Chalopin and Daniel Paulsuma. Graph labelling derived from models in distributed computing: a complete complexity classification. Technical report, LABRI, 2008.

[19] Laboratoire LABRI. Visidia. http://www.labri.fr/projet/visidia/, 2008.

[20] Thierry Lecomte, Dominique Cansell, and Dominique Méry. Patrons de conception prouvés. In *Journées Neptune 22 et 23 mai*, 2007.

[21] Joris Rehm and Dominique Cansell. Proved Development of the Real-Time Properties of the IEEE 1394 Root Contention Protocol with the Event B Method. In *En cours de publication ISoLA Workshop On Leveraging Applications of Formal Methods, Verification and Validation*, page ?, Poitiers-Futuroscope France, 12 2007.

[22] Antoine Requet and Lilain Burdy. Bart: B automatic refinement tool. Technical report, ClarSy, 2007. Diffusion Restreinte.

[23] Projet RODIN. Rodin platform. http://rodin-b-sharp.sourceforge.net/, 2007.

A Liste des documents fournis

1. Spécification de l'outil BART en cours de développement par ClearSy [22]: **Document à diffusion restreinte**
2. Développement de l'algorithme de Mazurkiewicz par Dominique Cansell du LORIA [9]
3. Développement incrémental et preuve de l'algorithme d'élection du leader dans un 2-arbre par Dominique Cansell du LORIA [8]
4. Modélisation du porte-monnaie électronique Mondex par Nazim Benaissa et Dominique Méry [5]
5. Modélisation et développement de trois nouveaux algorithmes d'élection de leader en utilisant le développemnt de l'algorithme d'élection du leader IEEE 1394 par Dominique Cansell et Dominique Méry [13]
6. Proposition d'un pattern de développement pour les programmes séquentiels structurés par Dominique Cansell et Dominique Méry [14]
7. Modélisation de la technique Manchester pour contrôler le stockage des votes électroniques par Dominique Cansell, Paul Gibson et Dominique Méry [11]
8. Mobile agent algorithms versus message passing algorithms par Jérémie Chalopin, Emmanuel Godard, Yves Métivier, and Rodrigue Ossamy [16]
9. An efficient message passing election algorithm based on mazurkiewicz's algorithm par Jérémie Chalopin and Yves Métivier [17]
10. Collecte d'informations par des agents mobiles par Shehla Abbas, Mohamed Mosbah, and Akka Zemmari [1]
11. Graph labelling derived from models in distributed computing: a complete complexity classification par Jérémie Chalopin and Daniel Paulsuma [18]
12. Merging time of random mobile agents par Shehla Abbas, Mohamed Mosbah, and Akka Zemmari [2]