Modélisation et vérification d'un réseau de communication embarqué avec FIACRE/TINA

Pierre-Alain Bourdil, Bernard Berthomieu, Eric Jenn, François Vernadat

Modélisation et vérification d'un réseau de communication embarqué avec FIACRE/TINA

Pierre-Alain Bourdil [1,2,4], **Bernard Berthomieu** [1,3], **Eric Jenn** [4], **François Vernadat** [1,2]

1. *CNRS, LAAS,7 av. colonel Roche,F-31400 Toulouse,France*
   {pierre-alain.bourdil,bernard.berthomieu,francois.vernadat}@laas.fr
2. *Univ de Toulouse, INSA,135 av. de Rangueil, F-31400 Toulouse, France*
3. *Univ de Toulouse, LAAS, 135 av. de Rangueil, F-31400 Toulouse, France*
4. *Thales Avionics, 105 av. du Général Eisenhower, F-31100 Toulouse, France*
   eric.jenn@fr.thalesgroup.com

RÉSUMÉ. *Ce poster présente nos travaux de thèse sur l'utilisation des techniques de modélisation et de vérification avec FIACRE/TINA. Après avoir modélisé l'objet notre cas d'étude avec FIACRE, un langage formel pour la description des systèmes temps réel, nous en exploitons les symétries structurelles pour améliorer le passage à l'échelle des techniques de model-checking. TINA est un outil d'analyse des réseaux de Petri temporels. Il permet la vérification de propriétés LTL par model-checking sur une abstraction de l'espace d'états d'un TPN. Nous étendons FIACRE et TINA pour la définition et l'exploitation des symétries. Les premiers résultats expérimentaux sont présentés.*

ABSTRACT. *In this poster, we present our thesis work on modelisation and verification with FIACRE/TINA. First, we build a formal model of our case study with FIACRE, a formal language to represent both the behavioural and timing aspects of real time systems. Then we exploit its structural symmetries to minimize the combinatorial explosion. TINA is a tool for the analysis of Time Petri Net. It allows for the model-checking of LTL formula on a state space abstraction of a TPN. We extend FIACRE and TINA for the specification and reduction of symmetry. First experimental results are shown.*

MOTS-CLÉS : *model-checking, TPN, TINA, FIACRE, réseaux, symétries*

KEYWORDS: *model-checking, TPN, TINA, FIACRE, network, symmetry*

1. Introduction

Dans le cadre de notre travail de thèse, nous modélisons formellement un réseau de communication afin d'en vérifier son comportement. Ce réseau est fortement contraint par son domaine d'application dont, notamment, des exigences de certification.

Nous modélisons le réseau avec le langage FIACRE (Berthomieu *et al.*, 2007). Nous utilisons la boîte à outils TINA (Berthomieu, Vernadat, 2006) pour vérifier son comportement avec les techniques de model-checking.

Pour permettre le passage à l'échelle, nous exploitons les symétries et les intégrons dans TINA et le langage FIACRE.

2. Modélisation avec FIACRE/TINA

FIACRE – Format Intermédiaire pour les Architectures de Composants Répartis Embarqués – est un langage formel pour la description des systèmes temps réel. Il a été initialement développé dans le cadre du projet AESE/Topcased. Le langage FIACRE décrit des composants. Les composants élémentaires sont des machines à états étendues ; des composants plus complexes sont décrits hiérarchiquement et de façon compositionnelle par assemblage de composants plus simples dont les interactions peuvent être contraintes par des exigences temporelles et des priorités.

TINA – Time Petri Net Analyzer – est un environnement logiciel permettant l'édition et l'analyse de réseaux de Petri temporels étendus avec des priorités et un traitement de données. Sur ces modèles, TINA permet la vérification de propriétés par model-checking pour, notamment, la logique temporelle à temps linéaire State/Event LTL.

Nos activités de modélisation et de vérification s'inscrivent dans un processus industriel contraint. Le réseau de communication objet de notre expérimentation répond à des exigences de performance (latence, gigue,...) et de qualité de développement strictes. Ceci se traduit notamment par l'assurance de la traçabilité entre les exigences auxquelles doit répondre le système de communication et le modèle formel.

Dans un premier temps, nous avons sélectionné un sous-ensemble des spécifications industrielles relevant de notre objectif de vérification. L'ensemble retenu sert de base informelle pour la modélisation. Les abstractions et simplifications réalisées dans notre modélisation sont argumentées sur cette base.

Concrètement, un noeud exécute plusieurs processus séquentiels (lecture des trames, traitements, émissions des trames) qui partagent des variables. Chaque processus séquentiel est modélisé par un `process` FIACRE. Un composant FIACRE, qui modélise le noeud, encapsule processus et variables partagées. Le réseau est lui-même représenté par un composant qui compose les noeuds. L'opérateur de composition fait apparaitre explicitement la structure du réseau ; c'est un bon support pour la spécification des symétries.

Nous modélisons les communications entre deux noeuds n_i et n_j par des canaux temporisés f_j^i, du noeud n_i au noeud n_j. n_i écrit dans f_j^i et n_j y lit. L'initialisation des noeuds est asynchrone.

Le déterminisme temporel est une propriété essentielle du réseau. En particulier, nous devons garantir le temps maximal de transmission d'un message. Comme nous n'intégrons pas de modèle de fautes dans notre étude, une condition nécessaire et suffisante du déterminisme est la taille finie des files d'attentes qui garantit l'absence de congestion du réseau.

Les symétries décrivent les transformations d'un objet qui le laisse invariant, par exemple les réflexions sur les diagonales d'un carré. Cette idée peut s'appliquer à un ensemble de processus concurrents exécutant le même programme. En permutant les

processus on ne change pas le comportement du système. Si ces processus communiquent les permutations doivent être consistentes avec la topologie. L'application de ces permutations sur l'état global du système est la base du mécanisme d'abstraction que l'on appelle réduction par symétrie.

3. Symétries structurelles

La théorie des groupes donne un cadre formel pour l'étude des symétries (Emerson, Sistla, 1996). Dans un espace d'états $\mathcal{M}$, une formule CTL^* f est satisfaite si et seulement si elle est satisfaite dans le quotient $\mathcal{M}/G$. G est le groupe des symétries du système et de la formule. C'est un sous-groupe de $\mathcal{A}(\mathcal{M}) \cap \mathcal{A}(f)$. $\mathcal{A}(\mathcal{M})$ est le groupe des automorphismes de $\mathcal{M}$ et $\mathcal{A}(f)$ celui de f. Pour construire $\mathcal{M}/G$, il faut pouvoir déterminer si deux états appartiennent à la même orbite. Dans ce but (Emerson, Sistla, 1996) propose le calcul d'une forme canonique, le Constructive Orbit Problem (COP). Ce problème est démontré *NP*.

Pour exploiter les symétries, il faut construire G puis le quotient. Le calcul *automatique* de G passe difficilement à l'échelle. La description *manuelle* de G par l'utilisateur, notamment comme des permutations sur des places et transitions, est une tâche difficile.

Nous pensons que définir les symétries par composition est un bon compromis : à la fois simple pour l'utilisateur tout en garantissant par construction des structures de groupes pour lesquelles le COP est simplifié (Donaldson, Miller, 2006).

De nombreux travaux existent autour de la réduction par symétrie sur les réseaux de Petri bas niveau. (Starke, 1991) donne la définition d'une symétrie valide sur les réseaux de Petri et montre que l'accessibilité est préservée dans le quotient du graphe des marquages. Aucune solution n'est proposée pour le COP.

(Schmidt, 2000) propose trois solutions pour le calcul de l'équivalence de deux états : l'énumération des symétries, le calcul d'une permutation à la demande ou le calcul d'une forme canonique (COP). Le calcul de la forme canonique est le plus efficace, mais il n'est applicable qu'aux groupes S_n (par ex.: l'ensemble des permutations de n éléments) et C_n(par ex.: les rotations de n éléments). Dans (Junttila, 2003), l'auteur utilise une représentation arborescente des groupes proposée par (Sims, 1970), pour trouver une forme canonique quelle que soit la structure du groupe.

(Chiola *et al.*, 1997) construisent une représentation symbolique des marquages qui regroupent les marquages équivalents et une règle de tir symbolique. Ils évitent ainsi le calcul de la forme canonique. (Junttila, 2003) étend le traitement des symétries aux symétries de données.

Si les symétries de réseaux de Petri ont été largement étudiées, nous n'avons trouvé que peu de travaux sur les symétries pour les systèmes temporisés. (Thierry-Mieg *et al.*, s. d.) propose une démarche proche de la nôtre mais pour les systèmes temporisés à temps discret, tandis que TINA travaille en temps dense.

Dans (Hendriks *et al.*, 2004), les auteurs traitent du cas des symétries sur les automates temporisés. Ils utilisent la définition manuelle des symétries via les *scalarsets* (Norris IP, Dill, 1996) pour spécifier des groupes $S_{|\mathcal{I}|}$ sur l'ensemble $\mathcal{I}$ des identifiants des processus. A condition que toutes les horloges soient réinitialisées à zéro et que la sur-approximation des enveloppes convexes ne soit pas utilisée, UPPAAL (Behrmann *et al.*, 2006) peut calculer la forme canonique d'un état.

4. Expérimentations

Nous avons développé un prototype de TINA qui permet la spécification et l'exploitation des groupes de symétries. Nous traitons les groupes C_n, S_n, leurs produits disjoints, et leurs produits couronnes (Donaldson, Miller, 2006).

Nous modélisons les trains de Berthomieu, le problème des philosophes, un système producteur/consommateur et des anneaux temporisés. Les premiers résultats sont très encourageants.

5. Conclusion

Les travaux en cours ont pour objectif d'étendre le langage FIACRE et l'outil TINA afin d'exploiter les propriétés de symétrie de certains systèmes.

Nous définissons explicitement des groupes de symétries par composition hiérarchique. Cette technique nous garantit par construction des structures de groupes pour lesquelles des solutions simples au COP sont possibles. Nous travaillons à des solutions efficaces au COP dans le contexte des systèmes temporisés. Nous envisageons d'étendre le langage FIACRE par l'ajout d'opérateurs de composition symétrique et l'exploitation de symétries sur les données.

Remerciements

Ce travail a été réalisé dans le cadre de la thèse CIFRE N°2011/0234, en collaboration avec Thales Avionics (Toulouse) et LASS-CNRS.

Bibliographie

Behrmann G., David A., Larsen K., Hakansson J., Petterson P., Yi W. (2006). UPPAAL 4.0. In *Quantitative evaluation of systems, 2006. qest 2006. third international conference on*, p. 125–126.

Berthomieu B., Bodeveix J.-P., Filali M., Lang F., Le Botlan D., Vernadat F. (2007). The syntax and semantic of FIACRE. *rapport LAAS*, vol. 7264.

Berthomieu B., Vernadat F. (2006). Réseaux de Petri temporels : méthodes d'analyse et de vérification avec TINA. In E. N. Navet (Ed.),, vols. Systèmes temps réel 1 – techniques de description et de vérification. Hermes.

Chiola G., Dutheillet C., Franceschinis G., Haddad S. (1997). A symbolic reachability graph for coloured Petri nets. *Theoretical Computer Science*, vol. 176, n° 1, p. 39–65.

Donaldson A., Miller A. (2006). Exact and approximate strategies for symmetry reduction in model checking. In J. Misra, T. Nipkow, E. Sekerinski (Eds.), *Fm 2006: Formal methods*, vol. 4085, p. 541-556. Springer Berlin / Heidelberg. (10.1007/11813040-36)

Emerson E. A., Sistla A. P. (1996). Symmetry and model checking. *Formal Methods in System Design*, vol. 9, p. 105-131. (10.1007/BF00625970)

Hendriks M., Behrmann G., Larsen K. G., Niebert P., Vaandrager F. W. (2004). Adding symmetry reduction to Uppaal. In K. G. Larsen, P. Niebert (Eds.), *Formal modeling and analysis of timed systems (FORMATS'03)*, p. 46–59. Springer–Verlag.

Junttila T. (2003). *On the symmetry reduction method for Petri nets and similar formalisms*. Research Report n° A80. Espoo, Finland, Helsinki University of Technology, Laboratory for Theoretical Computer Science.

Norris IP C., Dill D. L. (1996). Better verification through symmetry. *Formal Methods in System Design*, vol. 9, p. 41-75. (10.1007/BF00625968)

Schmidt K. (2000, janvier). How to calculate symmetries of Petri nets. *Acta Inf.*, vol. 36, n° 7, p. 545–590.

Sims C. C. (1970). Computational methods in the study of permutation groups. In *Computational problems in abstract algebra*, p. 169–183.

Starke P. H. (1991). Reachability analysis of Petri nets using symmetries. *Systems Analysis Modelling Simulation*, vol. 8, n° 4-5, p. 293–303.

Thierry-Mieg Y., Bérard B., Kordon F., Lime D., Roux O. H. (s. d.). Compositional analysis of discrete time Petri nets. In *1st workshop on Petri nets compositions (CompoNet 2011)*, vol. 726, p. 17–31.