

Modélisation et évaluation de performance des systèmes basés composants

Nabila Salmi

LISTIC - Université de Savoie
BP 80449, 74944, Annecy le Vieux, France
nabila.salmi@univ-savoie.fr

1. Introduction

Depuis la dernière décennie, l'industrie du logiciel s'oriente de plus en plus vers la conception de systèmes sous la forme d'assemblage de *composants*. L'objectif de ce type de conception est de réduire le coût et le temps de développement par réutilisation des composants.

Un *composant* est une unité de composition, munie d'*interfaces* spécifiées contractuellement, et de *dépendances* de contexte explicites [4]. Une interface est un point d'accès au composant, définissant des services offerts ou requis. Dans certains modèles tel que *FRACTAL* [1], le composant expose en plus, à travers des interfaces spécifiques nommées *contrôleurs*, des propriétés non fonctionnelles liées à son cycle de vie ou à des fonctionnalités d'environnement. Les composants sont assemblés en connectant leurs interfaces, formant un *système basé composant (CBS)*. Un composant peut lui-même contenir un nombre fini de *sous-composants*, permettant un emboîtement hiérarchique (composant *composite*). Les composants du plus bas niveau de la hiérarchie sont dits *primitifs*.

Il existe de nombreux *modèles de composants* tant dans le domaine académique que dans le domaine industriel (EJB, CCM, .net, Fractal, PECOS, Koala, etc.). Pour la plupart de ces modèles, un *langage de description d'architecture (ADL)* permet de décrire un assemblage de composants définissant une application. À partir d'une telle description, un outil de compilation génère le code de l'application en réalisant les vérifications usuelles correspondantes. À notre connaissance, très peu d'outils assurent une vérification formelle des propriétés comportementales de l'application, et encore moins une évaluation de ses performances. Nous présentons ici une méthode générale visant à pallier ces insuffisances, en particulier dans le domaine d'analyse de performances, fondée sur le modèle *Stochastic Well-formed Net (SWN)* [2]. Les

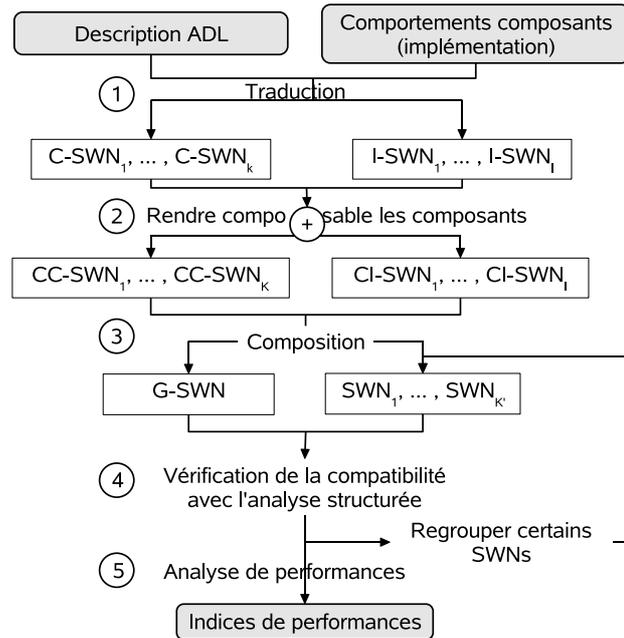


FIG. 1: Analysis method of a CBS

SWN sont une classe de réseaux de Petri stochastiques de haut niveau, largement utilisée pour l'évaluation des performances des systèmes à événements discrets. La méthode part de l'architecture à composants et modélise le CBS à étudier par conversion en SWNs à l'aide de règles systématiques. L'analyse de performances est ensuite conduite de manière structurée [3] sur le modèle obtenu. L'intérêt de cette méthode est de tirer parti de l'architecture compositionnelle de ces systèmes pour réduire la complexité d'analyse en termes de temps de calcul et de mémoire.

La section 2 présente les étapes de notre méthode. Nous décrivons en section 3 l'application de cette méthode aux systèmes basés *FRACTAL*.

2. Aperçu global de la méthode

Notre méthode (fig. 1) consiste en cinq phases :

1. À partir de la description du CBS, traduction du comportement (code source) de ses composants et de leurs interactions dans le contexte SWN. On obtient ainsi un ensemble de SWNs dits C-SWNs (pour les composants) et I-SWNs (pour les interactions complexes). Cette modélisation dépend fortement du modèle de composant utilisé. Les interactions peuvent être synchrones (invocation de méthode) ou asynchrones (envoi/réception d'événements).

2. Modification des C-SWNs et des I-SWNs de manière à les rendre composables au sens de la composition des réseaux de Petri (fusion de places ou transitions), menant à des SWNs de Composants Composables (CC-SWNs) et de Composants d'Interactions (CI-SWNs).
3. Composition des CC-SWNs et CI-SWNs (vus comme un ensemble unique $SWN_1, \dots, SWN_{K'}$) en fusionnant les éléments modélisant les interfaces. On obtient un G-SWN modélisant le CBS.
4. Vérification des conditions de l'analyse structurée (voir [3]) sur le G-SWN et l'ensemble $(SWN_{k'})_{k' \in \{1, \dots, K'\}}$. Si elles ne sont pas satisfaites, les SWN_k (cause de non satisfaction des conditions) sont regroupés dans un nouveau $SWN_{k'}$ qui les remplacera. Cette étape est répétée jusqu'à satisfaction des conditions.
5. Calcul des indices de performance par la méthode d'analyse structurée[3].

Le principe de l'analyse structurée est d'étudier d'une manière séparée les sous-réseaux associés aux composants, augmentés de "parties" agrégeant les interactions avec les autres sous-réseaux, et ce en calculant leurs graphes d'états symboliques (SRGs) respectifs. Ces SRGs sont utilisés pour dériver une représentation tensorielle du générateur d'une chaîne de Markov agrégée associée au réseau global. Les indices de performances sont calculés sur cette chaîne, ce qui engendre des gains importants en temps de calcul et en espace mémoire. Un outil *compSWN* a été développé à cet effet.

Nous détaillons ci-dessous la déclinaison de cette méthode d'analyse pour les CBS basés *FRACTAL*.

3. Modélisation et évaluation des CBS *FRACTAL*

3.1. Génération du G-SWN

La modélisation d'un CBS ne peut se faire sans connaître les détails d'implémentation. Pour cela, nous présentons le cas de CBS *FRACTAL* conçus sous l'implémentation de référence *Julia*. La construction du G-SWN se fait comme suit :

1. On part du plus bas niveau de l'architecture et on modélise les composants primitifs, obtenant un ensemble de C-SWNs. Les classes de couleur utilisées modélisent des entités de données (comme les requêtes), ou des entités actives (comme les threads). Les interfaces fonctionnelles sont modélisées par des transitions suivant des règles systématiques. Les contrôleurs ne sont pas modélisés ici car

on s'intéresse, pour calculer des indices de performances, à des configurations stables (à l'équilibre), où il n'y a pas de reconfiguration dynamique réalisée à l'aide des contrôleurs.

2. Le C-SWN obtenu est transformé en CC-SWN (via des règles), en traduisant les interfaces dans la forme adéquate à la composition. Dans le cadre de *Julia*, l'unique moyen d'interaction entre composants est l'appel de méthodes. Les interfaces sont donc modélisées conformément à ce type d'interaction synchrone.
3. Une fois les composants primitifs modélisés, on remonte dans les niveaux : pour chaque composite, on construit son CC-SWN en fusionnant les éléments d'interfaces des CC-SWNs correspondant aux sous-composants interconnectés, ainsi de suite jusqu'à atteindre le plus niveau, qui est celui de l'application.

3.2. Analyse du CBS

L'approche d'analyse structurée est appliquée aux CBS *FRACTAL*. Des conditions sur les entités colorées au sein d'un composant ont été spécifiées afin de garantir la satisfaction des conditions de notre méthode d'analyse.

4. Conclusion

Nous avons proposé dans ce papier une méthode qui permet d'étudier d'une manière efficace les performances des CBS basés modèle *FRACTAL*. Notre méthode fournit un SWN pour le système global et une collection de SWNs de composants qui seront analysés par notre méthode structurée pour le calcul d'indices de performance. L'automatisation de cette méthode d'analyse est en cours.

Bibliographie

1. Bruneton (E.), Coupaye (T.) et Stefani (J.B.). – The Fractal component model, version 2.0-3. – <http://fractal.objectweb.org/specification>, 2004.
2. Chiola (G.), Dutheillet (C.), Franceschinis (G.) et Haddad (S.). – Stochastic Well-Formed Colored Nets and Symmetric Modeling Applications. – IEEE Trans. Comp., 42(11), 1993, p.1343-1360.
3. Delamare (C.) et Gardan (Y.) et Moreaux (P.). – Efficient implementation for performance evaluation of synchronous decomposition of High Level Stochastic Petri Nets. – Proc. ICALP2003, Eindhoven, Holland, Jun.21-22, p.164-183.
4. Szyperski (C.). – Component software – Addison-Wesley, 2002, 2nd Edition.