氏　　名	张　　海　涛		
学　位　の　種　類	博士(情報科学)		
学　位　記　番　号	博情第 327 号		
学位授与年月日	平成 27 年 9 月 24 日		
論　文　題　目	Study on Model Checking Techniques for OSEK/VDX Applications （OSEK/VDX アプリケーションのためのモデル検査技術に関する研究）		
論文審査委員	主査　青木　利晃	北陸先端科学技術大学院大学	准教授
	平石 邦彦	同	教授
	緒方 和博	同	教授
	鈴木 正人	同	准教授
	岡野 浩三	信州大学	准教授

論文の内容の要旨

OSEK/VDX, a standard of automobile OS promulgated by the alliance of German and French automotive manufacturers in 1994, has been widely adopted by many automobile manufacturers to design and implement a vehicle-mounted OS, such as BMW, Opel and TOYOTA. Based on the OSEK/VDX OS, more and more applications are developed and deployed in vehicles to assist drivers to control vehicles, such as the cruise control system and temperature control system. However, with the growing functionalities in vehicles and increasing complexity in the development, how to straitly ensure the reliability of the developed OSEK/VDX applications is becoming a challenge for developers.

An application developed on OSEK/VDX OS consists of multiple tasks. To concurrently executing multiple tasks on one CPU, a scheduler within OSEK/VDX OS is used to dispatch tasks. Moreover, OSEK/VDX OS provides a standardized application interfaces (APIs) for its applications, and tasks within the applications can invoke the provided APIs to dynamically change the scheduling of tasks. To ensure the reliability of OSEK/VDX applications, model checking, which is an efficient and exhaustive verification technique for concurrent software, can be applied to verify OSEK/VDX applications for detecting subtle and logic errors.

There exist many model checking methods that have been applied to verify concurrent software such as the ANSI-C multi-threaded software and SystemC multi-threaded software. In the existing methods,

due to the concurrency of target software, all of the interleavings of threads are checked in the verification process in order to exhaustively verify the given software. Although the existing methods can exhaustively check concurrent software, these methods usually perform an approximate verification since the behaviours of scheduler are not taken into account in verification process. If we apply these existing model checking methods to verify OSEK/VDX applications, it is too imprecise since a lot of unnecessary interleavings of tasks will be superfluously checked in the verification stage, especially these unnecessary interleavings will usually result in a spurious bug. Furthermore, as a result of the spurious bug, developers have to spend extra costs judging whether the detected bug is real one or not after completing verification. As to reduce the checking costs, a more accurate model checking approach should be used in the verification of OSEK/VDX applications.

Recently, to accurately check concurrent software such as ANSI-C multi-threaded software and SystemC multi-threaded software, several prominent model checking methods have been proposed by some senior researchers. In these methods, as to seek a more accurate verification result, the behaviours of scheduler are taken into account in the verification process. Even so, these prominent model checking methods are still unsuitable to accurately verify OSEK/VDX applications because of different scheduling policy. In the existing methods, the running thread within the concurrent software is arbitrarily determined by scheduler (non-deterministic scheduler is used to dispatch threads). However, compared with the non-deterministic scheduler, in OSEK/VDX applications which task is to be run is explicitly determined by OSEK/VDX scheduler — a deterministic scheduler is used to dispatch tasks.

The purpose of this thesis is to make model checking technique more accurate in the verification of OSEK/VDX applications. In order to achieve this purpose, in this thesis we describe and develop three approaches that can accurately and automatically verify the safety property of OSEK/VDX applications using model checking technique. To the best of our knowledge, our work is first to apply model checking technique to accurately verify the multi-tasks/threads software which is dispatched by a deterministic scheduler.

To accurately verify OSEK/VDX applications using model checking technique, the behaviours of OSEK/VDX OS such as scheduler behaviours should be taken into account in the checking process since

the running task is explicitly determined by OSEK/VDX scheduler and the APIs invoked from tasks will haphazardly change the scheduling of tasks. In our first approach, we investigate a checking method based on the existing model checker Spin. In the approach, as to employ Spin model checker to accurately verify OSEK/VDX applications, we propose a checking model which is a combination model of application model and OSEK/VDX OS model to precisely simulate the executions of the OSEK/VDX applications in real OSEK/VDX OS. Compared with our first approach (Spin-based checking approach), in our second approach we develop a new technique named execution path generator (EPG) to verify the OSEK/VDX applications based on the SMT-based bounded model checking. In the approach, as to avoid the behaviours of OSEK/VDX OS to be explored in the verification stage, the OSEK/VDX OS model is embedded in EPG (constructing model algorithm flat) to dispatch tasks and respond to the APIs invoked from tasks in the process of constructing checking model of application. Furthermore, based on the sequentialization idea, in our last approach we present a novel method to translate OSEK/VDX applications into sequential models in order to efficiently apply existing model checkers such as Spin and cbmc to directly check OSEK/VDX applications. In particular, as to avoid the behaviours of OSEK/VDX OS to be poured into the sequential models, like our second approach (EPG technique), in the approach the OSEK/VDX OS model is embedded in translation algorithm flat to dispatch tasks and respond to the APIs invoked from tasks in the sequentialization process.

We have implemented our approaches and conducted two types of experiments to evaluate the proposed approaches. In the first experiments, as to show the accuracy of our approaches, the sequentialization-based checking approach is selected as candidate from our checking approaches, and we compared the checking approach with the existing model checking methods for concurrent software. The experiment results denote that our approach is an accurate checking method for OSEK/VDX applications in contrast with the existing model checking methods for concurrent software. In the second experiments, we evaluated the efficiency and scalability of our approaches based on a series of experiments. According to the conducted experiment results, we find the following results,

- The Spin-based checking approach can accurately verify the OSEK/VDX applications, but the scalability of this approach is limited because too many details about OSEK/VDX OS model are

explored in the verification stage, especially the state space explosion will happen if the checked applications hold a lot of tasks and APIs.

- EPG technique is more scalable than Spin-based checking approach in checking the applications which hold a lot of tasks and APIs. However, it is not efficient to check the applications which hold a large number of branches, since the checking model of application is constructed based on the execution paths in EPG technique. If an application hold a lot of branches, it will spend a lot of time exploring execution paths in the verification process, which will slow down the performance of EPG technique.
- Based on the sequentialization process of our approach, the selected model checker Spin can efficiently verify the applications which hold a lot of tasks, branches and APIs with the less cost in terms of states, time and memory compared with the Spin-based checking approach and EPG technique.

Furthermore, we have used our sequentialization-based approach to sequentialize many experimental OSEK/VDX applications which hold about 1000 lines of C code, and verified the sequentialized applications with the well-known bounded model checker cbmc. The performances indicate that the sequentialization-based approach and cbmc can be considered as a practical method to verify the OSEK/VDX applications with industrial complexity.

keywords:

OSEK/VDX applications, deterministic scheduler, model checking, Spin, bounded model checking, SMT, sequentialization

論文審査の結果の要旨

この博士論文では，車載 OS の世界標準である OSEK/VDX に基づいた OS(以下，OSEK OS と略す)上で動作するアプリケーションを検証するためのモデル検査手法およびツールを提案している．

OSEK OS 上で動作するアプリケーションは複数の並行タスクで構成され，それらは優先度ベ

ースのスケジューラで実行制御されている．ここで，OSEK OS のスケジューラは決定的スケジューラと呼ばれるものであり，一般的な並行性とは異なり，ディスパッチするタスクが決定的に決まる．このようなアプリケーションをモデル検査により検証するためには，スケジューラの考慮が必要であり，これが本論文の動機となっている．本論文では，モデル検査において，スケジューラを考慮する手法を 3 種類提案している．1 つ目は，既存のモデル検査ツール Spin を用いる手法である．この手法では，Spin の入力言語である Promela で OSEK OS のスケジューラを実現し，アプリケーションの振る舞いと組み合わせてモデル検査を実施する．しかしながら，この方法では，状態探索の際，OSEK OS のスケジューラ記述の部分まで探索されるため，効率が悪い．そこで，2 つ目の方法として，スケジューラの動作を検査対象の記述ではなく，有界モデル検査における状態探索のアルゴリズム内で実現する方法を提案した．この方法では，1 つ目の方法よりは効率が良いが，アプリケーションの動作を実行パスとして展開する際に効率が悪くなる場合があることが判明した．そこで，3 つ目の方法として，sequentialization に基づいた方法を提案した．この方法では，アプリケーションの並行動作を，OSEK OS のスケジューラを考慮して，並行動作を含まない動作に変換する(sequentialization)．そして，変換された動作を既存のモデル検査ツールで検証を行う．ここでは，スケジューラは，sequentialization 処理する際に考慮され，モデル検査における状態探索の対象にはならない．この方法は，上記 2 つの手法と比較して，大幅に効率的であることがわかった．

本論文の貢献は，モデル検査で決定的スケジューラを考慮するに適した方法を発見したことである．sequentilization は，並行動作を逐次処理に変換するというアイデアであり，決定的スケジューラにより制御されるアプリケーションのモデル検査では，このアイデアが有効に働くことがわかった．また，決定的スケジューラにより制御されるアプリケーションを対象としたモデル検査手法(有界モデル検査と sequentilization 手法)の提案は，本研究が最初であり，十分な新規性がある．

以上，本論文は，車載 OS の世界標準である OSEK/VDX に基づいた OS 上で動作するアプリケーションの信頼性を向上させる手法の提案を行っており，学術的に貢献するところが大きい．よって，博士（情報科学）の学位論文として十分に価値があるものと認めた．