

## 発表概要

# モデル検査を用いた分散システム開発用 プロトコルコンパイラの提案と実装

西山 達也<sup>1,a)</sup>

2015年1月13日発表

本発表では、分散システムにおける複雑なプロトコル実装を容易にし、かつモデル検査によるバグの検出が行える分散システム開発用プロトコルコンパイラを提案する。分散システムは複数のマシンが協調動作しながら処理を行うため、分散システムのプロトコルは通信過程における様々なマシンの状態を考慮する必要がある、実装が複雑化する。また、システムのバグの検出には、多くの試験を要するのに加え、試験自体も複雑なものになる。そこで、モデル化された分散システムにおいてとりうる状態を網羅的に探索し、システムが満たすべき性質を満たしているか検査するモデル検査の技術は上記課題において有力な手法である。しかし、システムをモデル化するには高度な専門性が要求されることに加え、実装から人手でモデルを抽出する、もしくはモデルを検証した後にモデルから人手で実装を行うことは、モデルと実装に乖離が発生する可能性がある。そこで、分散プロトコルの記述に特化した領域特化型言語（DSL）を定義し、DSL から SPIN によるモデル検査を行うための Promela のコードと実行するための Go 言語によるプロトコルの実装コードを生成するプロトコルコンパイラを提案する。DSL からモデルと実装のコードを生成することでモデル化に必要な専門性を持たない開発者であってもモデルと実装の乖離なく、分散プロトコルに対するモデル検査を実施することが可能となる。

## Proposal and Implementation of Protocol Compiler for Distributed System Development by Use of Model Checking

TATSUYA NISHIYAMA<sup>1,a)</sup>

Presented: January 13, 2015

This presentation proposes a protocol compiler for developments of distributed system by used of model checking. The protocol compiler can make complex implementations of protocols in distributed systems simple, and detects bugs using model checker. A implementation of protocols is complicated in a distributed system since components may run in parallel and network communication can be asynchronous and non-deterministic. Moreover, detecting bugs of distributed systems needs many difficult software tests, because the system's failures result from a combination of factors. Model checking is useful technique for systematically detecting these bugs, because it can check satisfying properties that the model must by exploring the reachable states. However, model checking is difficult for most system developers, because it requires technical knowledge to model target systems by formal modeling language. In addition, it is quite likely that modeling target system by developer causes a mismatch between a model and a implementation of target system. Therefore, the protocol compiler proposed by this presentation provides domain specific language (DSL) for describing distributed protocols, and generates executable codes written by go language and model codes written by promela from DSL. Then, the model wrtten by promela is chekked by SPIN model checker. In this apploach, developers can easily describe distributed protocol implementation and detect bugs using model checker without a mismatch between a implementation and a model.

---

<sup>1</sup> NTT ソフトウェアイノベーションセンタ  
NTT Software Innovation Center

<sup>a)</sup> nishiyama.tatsuya@lab.ntt.co.jp