Recursive Advice for Coordination

Aspect-oriented programming is a programming paradigm that is often praised for the ability to create modular software and separate cross-cutting concerns. Recently aspects have been also considered in the context of coordination languages, offering similar advantages. However, introducing aspects makes analyzing such languages more difficult due to the fact that aspects can be recursive - advice from an aspect must itself be analyzed by aspects - as well as being simultaneously applicable in concurrent threads. Therefore the problem of reachability of various states of a system becomes much more challenging. This is important since ensuring that a system does not contain errors is often equivalent to proving that some states are not reachable. In this paper we show how to solve these challenges by applying a successful technique from the area of software model checking, namely communicating pushdown systems. Even though primarily used for analysis of recursive programs, we are able to adapt them to fit this new context.