

Title	Actor-based Protocol Composition Framework for Rapid Prototyping
Author(s)	Rattanaponglekha, Nuttapong
Citation	
Issue Date	2011-09
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/9926
Rights	
Description	Supervisor:Associate Professor Xavier Defago, 情報科学研究科, 修士

Actor-based Protocol Composition Framework for Rapid Prototyping

Nuttapong Rattanaponglekha (0910210)

School of Information Science,
Japan Advanced Institute of Science and Technology

August 10, 2011

Keywords: protocol composition, actor-model, rapid prototyping, composer, expressiveness.

Nowadays, the development of distributed systems and applications are larger and more complex. In one application, we have many protocols inside that have interactions together. Protocol developers not only consider their own protocol but they also have deep knowledge in other protocols. In each protocol, there are many hindrances, complexity that distract the developing application, prototyping the algorithm and evaluating the performance in each algorithm. The thesis has contribution in protocol composition framework. The protocol composition framework should provide facilities to make protocol programming and protocol composition easier. The idea of protocol composition framework is separated into 2 views, *composer* and *protocol programmer*. Composers no need to have deep knowledge of the composed protocol that includes details such as the cumulative state of the protocols to protect, or the handlers in which new threads are launched. On the other hand, protocol programmers only consider their own protocol logic, have no deep knowledge in other protocols. Protocol composition framework should provide facilities to make protocol programming and protocol composition easier.

Our protocol composition framework is based on actor model. By using actor model, we target to loose decoupling between protocols as much as

possible and make algorithm code be more compact and expressive. We implement some protocols in agreement problems such as consensus algorithm, atomic broadcast algorithm. We use *boilerplate code* and *effective code* as criteria, compare the protocols on our framework with other framework and pseudocode. In the evaluation, we evaluate the expressiveness of our protocol in terms of line of code. We express the analysis of writing style compared with pseudocode and other framework. In terms of performance overhead, we evaluate the initialization time, the execution time and availability.