

Parallel and Distributed Systems Group (PDS)
Department Software and Computer Technology
Faculty EEMCS
DELFT UNIVERSITY OF TECHNOLOGY

LAB EXERCISES

DISTRIBUTED ALGORITHMS (IN4150)

Exercise 2c

Implementation of detection of global states
according to Chandy-Lamport in Java/RMI

D.H.J. Epema

January 31, 2012

Assignment

Implement Chandy's and Lamport's algorithm for detecting global states in a distributed system with Java/RMI. The implemented program should be **truly distributed** in that it can be demonstrated to run across multiple physical machines. The assignment can be split up into the following three parts.

First day

Write the remote interface and the global framework of the `Component` class implementing the components of the distributed algorithm. In addition, create the framework for the `Main` class that will create the `Component` objects and their threads on a single host. It must be possible to specify the number of these components. Include into `Main` and `Component` the functionality of registering and looking up components.

Second day

Include into `Component` the functionality for sending and receiving messages, and for maintaining the local state.

Third day

Include into `Component` the algorithm for recording the global state. It can be assumed that a single designated process starts the algorithm. Make sure that the output of the algorithm makes it possible to check its correct operation.

Hints

- Simulate a complete network.
- Let the messages that processes send to each other be numbered sequentially, and include random delays between send operations.
- Denoting the number of processes by n , let the state of a process consist of $2(n - 1)$ integers, which indicate the numbers of the last message sent to and received from every other process.
- Let the state of a channel be the sequence of the numbers of the messages sent but not yet received along the channel.