

Asynchronous ASMs

Multiple Reads One Write
(non-exclusive and exclusive resource sharing)

Egon Börger

Dipartimento di Informatica, Università di Pisa
<http://www.di.unipi.it/~boerger>

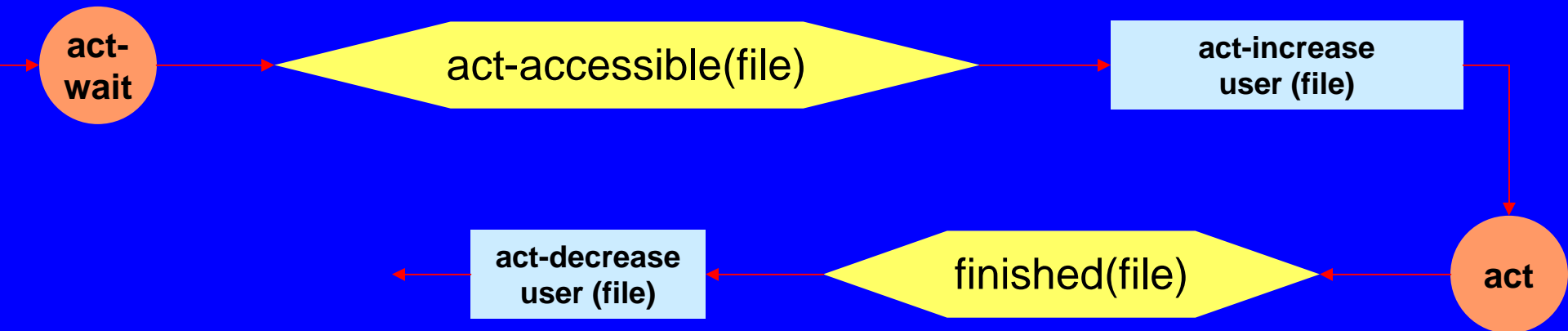
Multiple Reads One Write : problem statement

- Goal: Design a distributed algorithm allowing at each step one agent to start a read or a write operation in a given file, up to $\text{max-read} > 0$ simultaneous reads but only 1 write (not overlapping with any read)
 - Below we define agent rules for read/write access to files which in a distributed run of an async ASM (of agents all of which are equipped with those rules) reach the requested goal
 - NB. In the example we do not define the partial order for such distributed runs.

Multiple Reads One Write : Agent Signature

- **Agent** : set of agents which are allowed to access files for read/write operations
 - **finished:File** \rightarrow Bool indicating whether an agent has finished his current file operation
- **File**: set of files equipped with
 - **user** : Nat indicating the number of agents which are currently reading or writing the file
 - **max-read,max-write** maximal number of agents allowed to simultaneously use the file for reading/writing

Multiple Reads One Write: agent rules for act=read/write



act-accessible(file) \circ
 $\text{user (file)} < \text{max-act (file)}$

$\text{max-write(file)} = 1$
write-in/decrease user(file) \circ
 $\text{user (file)} := \text{user(file)} \pm 4$

$\text{max-read (file)} = 4$
write-in/decrease user(file) \circ
 $\text{user (file)} := \text{user(file)} \pm 1$

If instead of 1 one wants to allow multiple simultaneous file access attempts,
max-read becomes a **cumulative** counter to have the expected overall effect

read-accessible (file) becomes $\text{user (file)} + \text{newUsers} \leq \text{max-act (file)}$

where newUsers indicates the number of users attempting to access the file for reading

References

- W.Reisig: Elements of Distributed Algorithms. Modeling and Analysis with Petri Nets.
 - Springer 1998, Ch.24
- E. Börger, R. Stärk: Abstract State Machines. A Method for High-Level System Design and Analysis Springer-Verlag 2003, see <http://www.di.unipi.it/AsmBook>
 - See Chapter 6.1