Specification and Analysis of a Reliable Broadcasting Protocol in Maude

Grit Denker[1], J.J. Garcia-Luna-Aceves[2], José Meseguer[1], Peter Csaba Ölveczky[1], Jyoti Raju[2], Brad Smith[2], Carolyn L. Talcott[3]

[1]Computer Science Laboratory

SRI International, Menlo Park, CA

[2]Computer Engineering Department

University of California, Santa Cruz, CA

[3]Computer Science Department

Stanford University, Palo Alto, CA

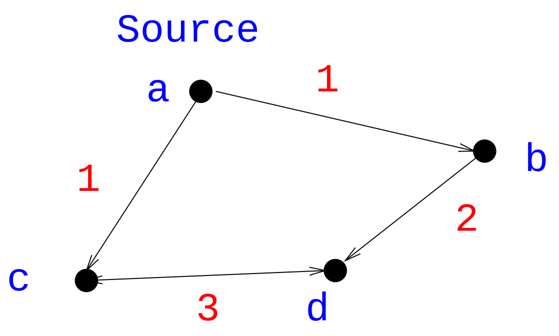
Reliable Broadcast Protocol

- performs reliable broadcasting of information in networks with dynamic topology, including failure and mobility

- aim: ensure that all nodes that satisfy certain connectedness criteria receive the information within finite time, and that the source is notified about it

- advantages of RBP: low latency and as few messages as possible

RBP intuitive



a is parent of b and c
b and c are siblings
b is parent of d
c and d are siblings

Node a snd Msg to neighbors b and c
Node b snd Msg to neighbor d
Node c snd Msg to neighbor d
Node d snd Msg to neighbor c

Limitations of Pseudocode Specification

- not executable
- no modeling and analysis tools available
- incomplete case coverage due to incomplete description of state information
- key assumptions on communication not explicit

Advantages of Using Maude

- get formal executable model
- can subject the model to formal analysis and verification
- the specification process reveals incompleteness in the original specification
- execution uncovers bugs and inconsistencies
- formal analysis of different concurrent behaviors can uncover subtle design errors such as race conditions, deadlocks, and unexpected behavior
- new validated and complete pseudocode can be derived from the Maude specification (this could be automated)

Value Added – Static Case (I)

- *Formalization:* Made essential, but implicit, knowledge explicit. *Example: No message overtaking on the path between two nodes*

- *Formalization:* Ambiguities in the pseudo-code description were clarified.

Value Added – Static Case (II)

- *Validation using default interpreter:* Let us discover errors in the specification as early as possible at no cost and led to a clear formalization of the main aspects of the given informal protocol.

- *Analysis using exhaustive search:* Showed an "unexpected" deadlock state which invalidated some of the main aspects of the protocol, resulting in a substantially revised Maude specification.

Value Added – Dynamic Case (I)

- *Formalization:* Formal and complete definition of all data structures to store information for each nodes. Issues clarified: Partially defined data structures, default values, monotonicity of data structures, consistency conditions between different data structures.

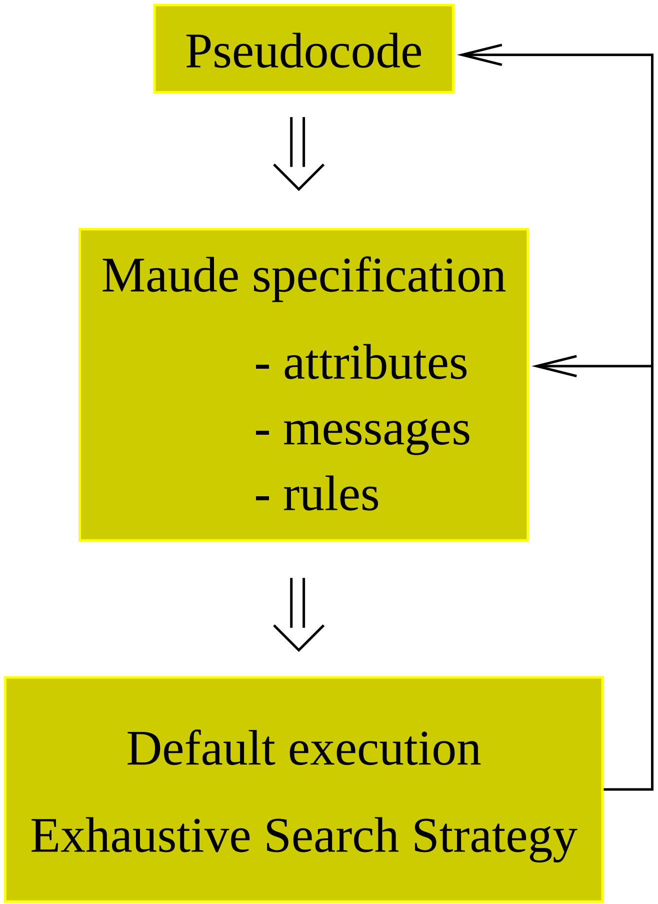
- *Formalization:* Determining a complete initialization scheme for valid network states, including initial configurations and initializing new nodes.

Value Added – Dynamic Case (II)

- *Validation using default interpreter:* Led to full case coverage. Due to underspecification (missing cases), running the Maude specification terminated in states which were not final.

- *Analysis using exhaustive search:* Produced a revised version of protocol dynamics due to unexpected behaviors. The analysis revealed an implicit assumption about the duration of existence of messages in the network (Message packets that are not receivable should be dropped).

Analysis Method



RBP – specification

Pseudocode

N^i = set of direct neighbors

ST^i_j = state (active/passive) of node i wrt to source j

SQ^i_j = sequence number

...

Start (i=j):

source node j checks if ST^j_j = passive. If yes, then

- send latest msg for source j to every k in N^j
- ST^j_j := active
- SQ^j_j := SQ^j_j + 1
- ST^j_{jk} := active for all k in N^j
- s^j_j := j

Maude specification

attributes

```
class Node | nbs : OidSet,
             states : Map[Oid,Status],
             seqNos : Map[Oid,Int],
             ...
op active : -> Status .
...
```

messages

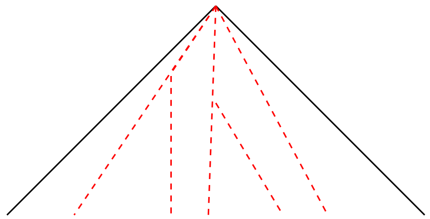
```
msg msg_To_From_Src_ :
    Number Oid Oid Oid -> Msg .
msg multimsg_To_From_Src_ :
    Number OidSet Oid Oid -> Msg .
```

conditional rule

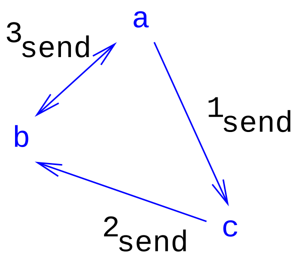
```
crl [Start] :
 < J:Node | nbs : OIDSET,
            states : (J,passive) u MAP,
            seqNos : (J,N) u MAP'
            ... >
 => < J:Node | states : (J,active) u Map,
               seqNos : (J, N+1) u MAP',
               ... >
    multimsg N+1 To QIDSET From J Src J
 if OIDSET =/= mt .
```

Finding a Deadlock through Formal Analysis

Executing the specification using exploratory strategy



delivered a deadlock




```
c cannot ack a (needs ack from child b)
b cannot ack c (needs ack from sibling a)
a cannot ack b (needs ack from child c)
```

node a is grandparent and sibling to node b

New "Initialization" Pseudocode

Initialize: for all nodes i do:

ST^i_j := *passive* for all $j \in (N^i + i)$
ST^i_{jk} := *passive* for all $k \in N^i$ and all $j \in (N^i + i)$
SQ^i_j := 0 for all $j \in (N^i + i)$
s^i_j = nil
all other fields are undefined!

Original and New "Start" Pseudocode

Start: source node j sends a message to every $k \in N^j(0)$ where N^j is the neighbor set of j. Update $ST^j_{jk}(0) \leftarrow$ *active* for all $k \in N^j(0)$.

Start: source node j checks if ST^j_j = *passive*. If yes, then
send a message (the latest one) for source j to every $k \in N^j$
ST^j_j := *active*
$s^j_j := j$
ST^j_{jk} := *active* for all $k \in N^j$
$SQ^i_j := SQ^i_j + 1$

Original ”NewLink” Pseudocode

For new link (i, b):

i sends $\text{mesg}_{SQ_j^i}$ to b

$ST_{jb}^i \leftarrow$ *active*

$ST_j^i \leftarrow$ *active*

$S_j^i \leftarrow i$

New "NewLink" Pseudocode

For new link (i, b): For all nodes j for which i has a defined value SQ^i_j do:

if ST^i_j = *active* and SQ^i_j is defined

    $N^i := N^i + b$

    ST^i_{jb} := *active*

    send message SQ^i_j for source j to b

    if ST^i_b is not defined

        ST^i_b := *passive*

    if SQ^i_b is not defined

        $SQ^i_b := 0$ /* the initialized state

    if ST^i_{bk} not defined

        ST^i_{bk} := *passive* for all $k \in N^i + b$

if ST^i_j = *passive* and SQ^i_j is defined

    ...

Conclusion

- Maximum benefit via early insertion of formal modeling

- Simplicity and intuitive appeal of the formalism
  "Rewriting Logic"

- Executability of Rewriting Logic: simulation, testing, debugging

- Formal analysis: highly concurrent and nondeterministic behavior of communication protocols becomes feasible

...and it is all in the web!

http://maude.csl.sri.com/

- Maude documentation, tutorial, and examples
- Maude executables
- Documentation of mid-size case studies with overview of the case study, executable Maude specifications, Maude strategies for analysis, sample runs and analysis scenarios, and related papers for:
    1. Reliable Broadcast Protocol
    2. Cryptographic Protocol Analysis
    3. CCS operational semantics and Hennessy-Milner-Logic