**Title:** Abstraction-based approaches to correct reactive software

**Author:** S. Purushothaman Iyer

**Performing Organization:**
North Carolina State University

**Sponsoring/Monitoring Agency:**
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

**Abstract:**
This final report on our efforts to make abstraction-based reasoning of reactive software possible. Concurrency Workbench, a tool for designs of reactive concurrent systems was enriched to handle programs. The enrichment depends upon notions of abstractions, a technique used in compilers. Furthermore, work was also done on enriching the Concurrency Workbench to deal with probabilistic information regarding errors during a systems execution.
MEMORANDUM OF TRANSMITTAL

U.S. Army Research Office
ATTN: AMSRL-RO-BI (TR)
P.O. Box 12211
Research Triangle Park, NC 27709-2211

☐ Reprint (Orig + 2 copies)  ☐ Technical Report (Orig + 2 copies)
☐ Manuscript (1 copy)  ☐ Final Progress Report (Orig + 2 copies)
☐ Related Materials, Abstracts, Theses (1 copy)

CONTRACT/GRANT NUMBER: DAAG55-98-1-0309

REPORT TITLE: Final report on Abstraction-based approaches to correct reactive software

is forwarded for your information.

SUBMITTED FOR PUBLICATION TO (applicable only if report is manuscript):

[Signature]

Sincerely,
Final Report: ARO Project DAAG55-98-1-0309
Abstraction-based approaches to correct Reactive Software

S. Purushothaman Iyer
September 17, 2001

1 Summary of goals

The goal of the research supported by the grant DAAG-55-98-1-0309 from ARO was to increase the ability of the Concurrency Workbench, a tool for specifying and reasoning about reactive systems, to deal with programs, and not just design notations. With this goal in mind we considered the following two main problems:

- Is it possible to build an interface to the Concurrency Workbench such that it accepts C programs as inputs? The motivation for considering this problem is so that the reach of model-checking and simulation relations checking can be extended from finite state designs to programs.

- Can the Concurrency Workbench be extended to handle design notations that have probabilistic components in them?

Furthermore, given that we wish to extend model-checking type techniques to programs, we also considered the problem of simulating concurrent programs.

2 Summary of Important Results

The important results achieved during the project are as follows:

1. C2CCS – a system to abstract C programs based on user input has been constructed. This system, which borrows ideas from work on Compilers and data flow analysis, allows an user to input a C program and instructions on what aspects should be abstracted, and how. For instance, the user could specify that the only meaningful value of a program variable are whether it is positive or negative. With such specifications about all variables in a program the C2CCS converter can build a finite state model of the program that is being analyzed. The output of the converter is a specification in CCS that can be input to the Concurrency workbench, which can then be reasoned...
about. As of the writing of the report the tool is almost done, and is being tested on protocol specifications such as the i-Protocol. The principal implementor, Dan DuVarney, is also currently finishing up his dissertation and will defend it very soon.

2. One of the main strengths of Process Algebras, and its implementation in the Concurrency Workbench, is compositionality, i.e., the ability to reason about a system by reasoning about its subsystems. In the presence of probabilistic information, such as the error rate of communication medium, properties can not be established with certainty but with certain probability. We have designed enriched version of the specification language used in the Concurrency Workbench and an enriched version of mu-calculus, the requirements language used in the workbench. This work is ongoing and is supported by a new grant from ARO.

3. With colleagues from France, the PI investigated how abstractions can be used for simulating programs. While simulations can be used to find the presence of bugs and not the absence of bugs, it forms a major component in any program development environment. To extend the reach of our techniques we investigated how symbolic representations could be used to drive the simulation faster, and get it to cover larger parts of a program’s execution. The main result is that a class of regular expressions can be effectively used in computing the interactions among a set of processes that use buffered communication, as in network protocols.

3 Listing of Publications

Peer Reviewed Journals


Refereed Workshops and Conferences


Papers under consideration


4 List of participating personnel

The following people participated in the project:

1. Dan DuVarney, Graduate Student, North Carolina State University, Raleigh, NC 27695-7534. ABD. Will submit and defend his PhD dissertation by November 2001.

2. Murali Narasimha, Graduate Student, North Carolina state University, Raleigh, NC 27695-7534. Graduated with a PhD 1999. Currently at Ericsson R&D, RTP, NC

3. Rance Cleaveland, Professor, Department of Computer Science, State University of New York, Stony Brook, NY.

4. S. Purushothaman Iyer, Associate Professor, North Carolina State University, Raleigh, NC 27695-7534.