43115
p- 8

Programming in a Proposed 9X Distributed Ada

REPORT 1

(NASA-CR-188831) PROGRAMMING IN A PROPOSED 9X DISTRIBUTED Ada Report No. 1 (Research Inst. for Advanced Computer Science) 8 p

N91-32825

CSCL 09B

Unclas

G3/61 0043115

Raymond S. Waldrop
Richard A. Volz
Stephen J. Goldsack

Texas A&M University

September 10, 1990

Cooperative Agreement NCC 9-16
Research Activity No. SE.35

NASA Johnson Space Center
Engineering Directorate
Flight Data Systems Division

ricis

Research Institute for Computing and Information Systems
University of Houston - Clear Lake

T·E·C·H·N·I·C·A·L R·E·P·O·R·T

The RICIS Concept

The University of Houston-Clear Lake established the Research Institute for Computing and Information systems in 1986 to encourage NASA Johnson Space Center and local industry to actively support research in the computing and information sciences. As part of this endeavor, UH-Clear Lake proposed a partnership with JSC to jointly define and manage an integrated program of research in advanced data processing technology needed for JSC's main missions, including administrative, engineering and science responsibilities. JSC agreed and entered into a three-year cooperative agreement with UH-Clear Lake beginning in May, 1986, to jointly plan and execute such research through RICIS. Additionally, under Cooperative Agreement NCC 9-16, computing and educational facilities are shared by the two institutions to conduct the research.

The mission of RICIS is to conduct, coordinate and disseminate research on computing and information systems among researchers, sponsors and users from UH-Clear Lake, NASA/JSC, and other research organizations. Within UH-Clear Lake, the mission is being implemented through interdisciplinary involvement of faculty and students from each of the four schools: Business, Education, Human Sciences and Humanities, and Natural and Applied Sciences.

Other research organizations are involved via the "gateway" concept. UH-Clear Lake establishes relationships with other universities and research organizations, having common research interests, to provide additional sources of expertise to conduct needed research.

A major role of RICIS is to find the best match of sponsors, researchers and research objectives to advance knowledge in the computing and information sciences. Working jointly with NASA/JSC, RICIS advises on research needs, recommends principals for conducting the research, provides technical and administrative support to coordinate the research, and integrates technical results into the cooperative goals of UH-Clear Lake and NASA/JSC.

Programming in a Proposed 9X Distributed Ada

REPORT 1

Preface

This research was conducted under auspices of the Research Institute for Computing and Information Systems by Raymond S. Waldrop and Richard A. Volz of Texas A&M University and Stephen J. Goldsack of Imperial College, London, England. Dr. E.T. Dickerson served as RICIS research coordinator.

Funding has been provided by the Engineering Directorate, NASA/JSC through Cooperative Agreement NCC 9-16 between NASA Johnson Space Center and the University of Houston-Clear Lake. The NASA technical monitor for this activity was Terry D. Humphrey, of the Systems Software Section, Flight Data Systems Division, Engineering Directorate, NASA/JSC.

PAGE ___ INTENTIONALLY BLANK

PROGRAMMING IN A PROPOSED 9X DISTRIBUTED ADA

REPORT 1

by

Raymond S. Waldrop[1]
Richard A. Volz[2]
Stephen J. Goldsack[3]

1 Introduction

This is the first report from the joint project[4] of Texas A&M University and Imperial College to study the proposed Ada 9X constructs for distribution. The goal for this time period was to select suitable test cases to help in the evaluation of the proposed constructs. The examples were to be considered according to the extent to which they

- required real-time operation.
- required fault-tolerance at several different levels.
- demonstrated both distributed and massively parallel operation.
- reflected realistic NASA programs.
- illustrated the issues of configuration, compilation, linking and loading.
- indicated the consequences of using the proposed revisions for large scale programs.
- covered the spectrum of communication patterns: predictable, bursty, small and large messages.

The first month of the project was spent identifying possible examples and judging their suitability for this project.

[1]Texas A&M University
[2]Texas A&M University
[3]Imperial College, London, England
[4]This work is supported by NASA subcontract #074 Cooperative Agreement NCC-9-16.

2 Examples Considered

There were four examples considered. They were:

- A set of three real-time program scenarios being prepared by IBM for the Navy that are typical of naval platform operation.
- The Ada implementation of the NASREM architecture that has been done by NIST.
- The distributed tele-autonomous robot system developed at the University of Michigan (currently being ported to Texas A&M University).
- Software systems being developed by NASA and which could be made available for study.

Consideration of these examples led to the following conclusions:

- There were no suitable software systems currently available at NASA in Ada.
- The distributed tele-autonomous robot system would require a significant amount of effort to convert to Ada for use in this study and should therefore be eliminated from consideration in favor of systems requiring less effort.
- The Ada implementation of the NASREM architecture would require a significant amount of effort to become familiar with the code before modifications could be undertaken. Also, there was some question about the availability of the actual Ada code for this study.
- The IBM real-time scenarios provide a good start for the specification of suitable test programs. Of the three scenarios, the submarine CIC scenario was the most completely specified and therefore was chosen to be the basis of our example.

3 Current status of the chosen scenario

The specification of the submarine CIC scenario is largely devoted to the description of the 5 time-critical functions required by the scenario and the size and frequency of the data passing between those functions. As it stands, the CIC scenario meets the following selection criteria:

- It requires real-time operation.
- It demonstrates distributed operation.
- It reflects a realistic system.
- It will illustrate the issues of configuration, compilation, linking and loading.
- It will indicate the consequences of using the proposed revisions for non-trivial systems.

- It includes predictable, bursty, small and large messages.

At present, though, the CIC scenario merely mentions that the system should be fault-tolerant but does not adequately address the issue. We feel that it will not be difficult to modify the scenario specification to address fault-tolerance and configuration issues at several levels such as:

- node replacement, i.e. replacement of all functions on a single physical processor. An example of when this feature would be used would be when one of the system computers either failed or had to be taken down for maintenance.
- partition replacement, i.e. replacement of one or more functions on a single physical processor. This feature would be useful for reconfiguring the system to provide better load balancing or for providing fault tolerance in the event of the failure of a single function rather than the processor itself.
- mode changes, i.e. changes in system functionality in response to changes in the environment. An example of such a change would be a switch to a heightened state of defensiveness due to enemy targets in the area.

In addition, we believe that these modifications will provide insight into the proposed Ada 9X constucts' interactions with fault tolerance issues such as consistency of system state after fault recovery, data sharing by modules involved in fault recovery, internal state of modules affected by fault recovery, and methods of replacement of faulty modules. This latter issue concerns the degree to which a backup module is immediately able to take over processing for the module it is replacing. This can range from a condition where the backup module has been monitoring communication with the primary module and can immediately take over when a fault occurs (hot backup), to the condition where the backup module has no information concerning the primary module's status when it was replaced and other modules must send unfulfilled requests for service to the backup module.

The next goal for the project is to revise the specification of the CIC scenario to include features which illustrate these issues.