*IN-61-CR*
*243103*
*P-15*   *552479*
*26p.*

# Transferring Data Objects
# A Focused Ada Investigation

## Sue LeGrand

*SofTech, Inc.*

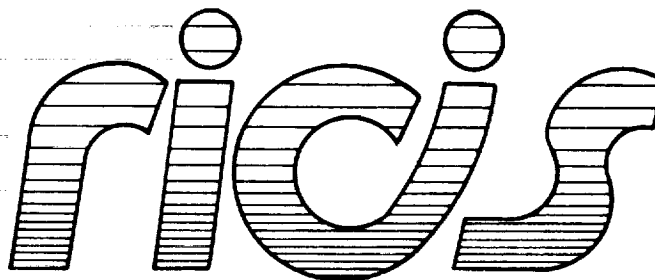April 30, 1988

**Cooperative Agreement NCC 9-16**
**Research Activity No. SE.17**

*ricis*

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

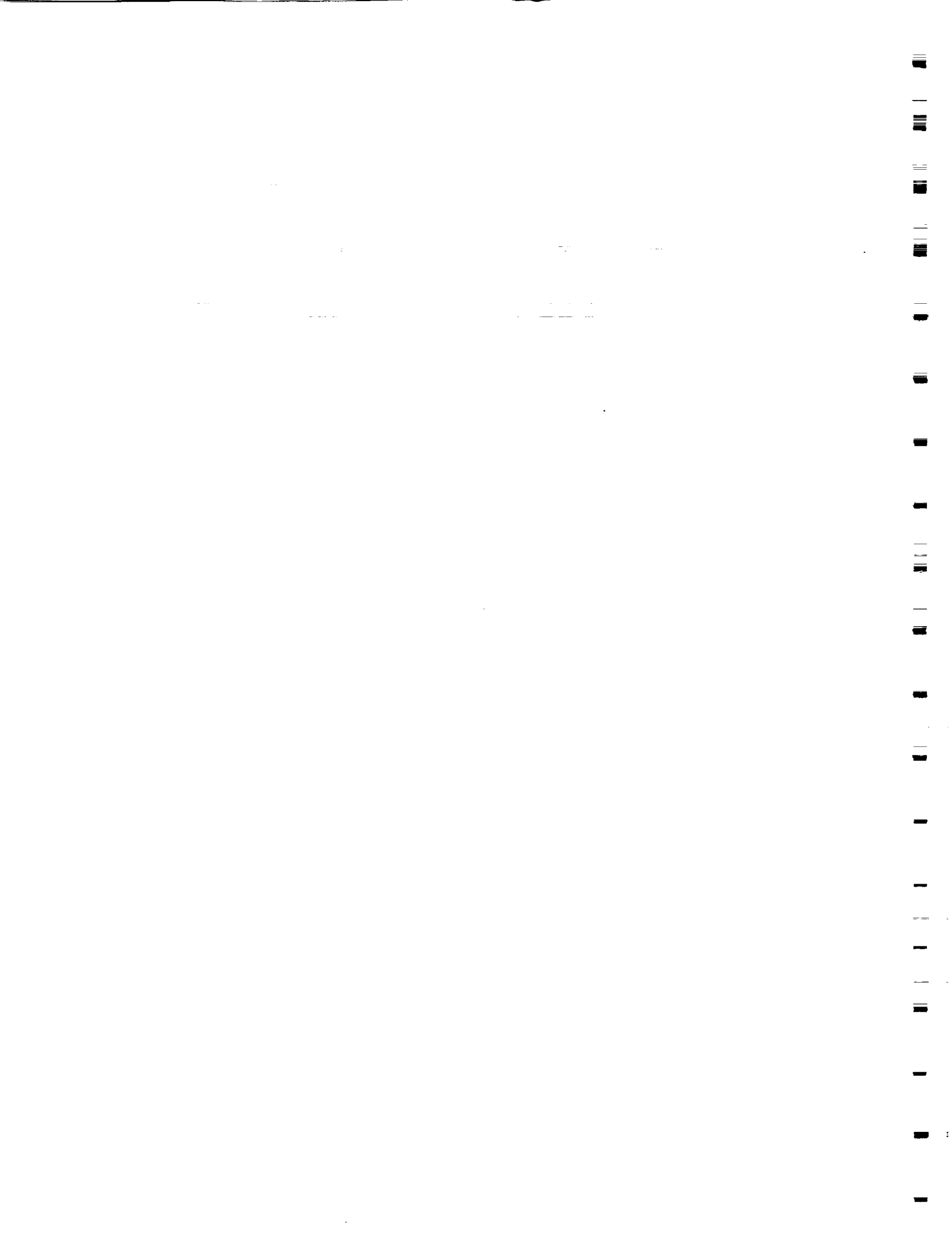# T·E·C·H·N·I·C·A·L    R·E·P·O·R·T

# 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.

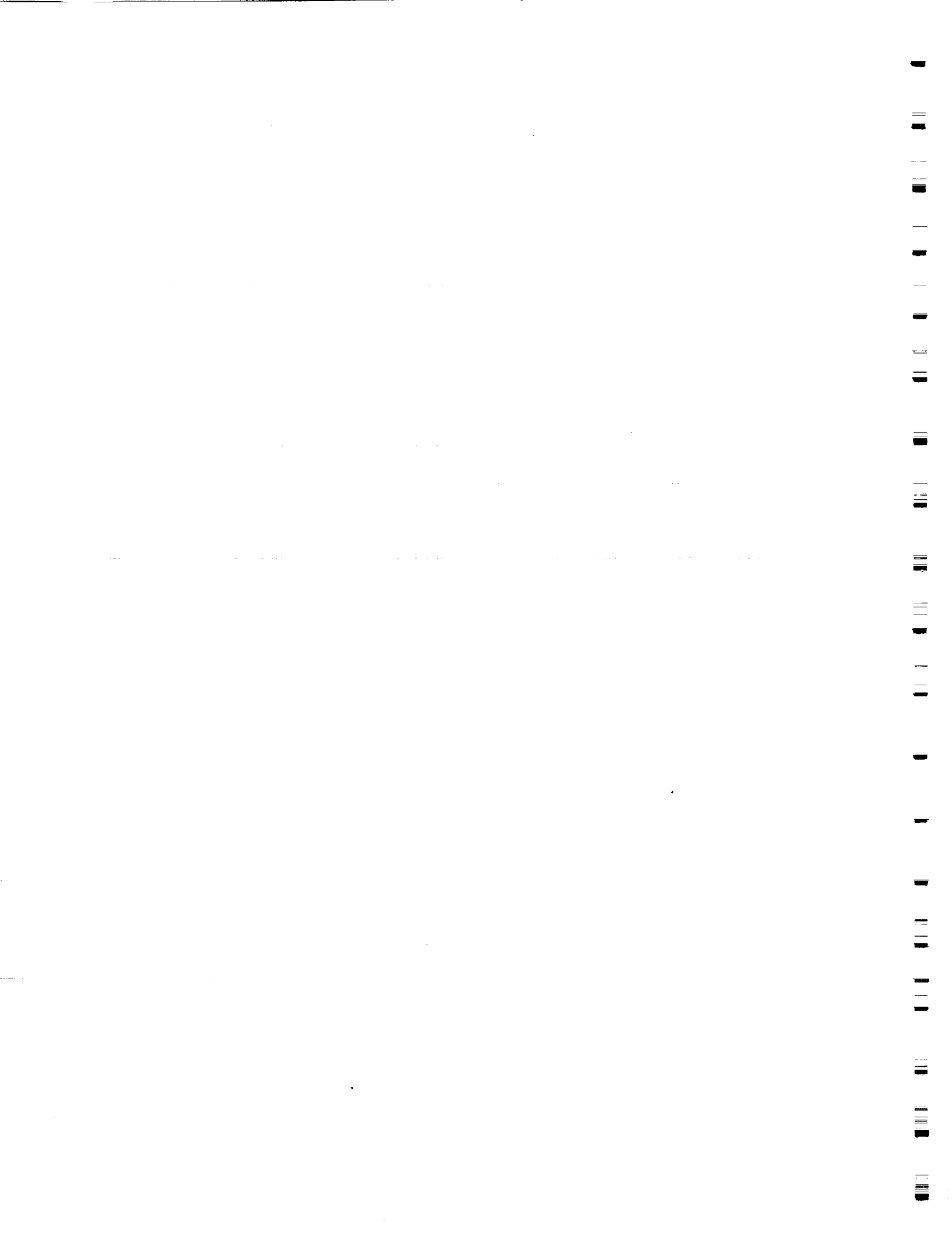# *Transferring Data Objects*
# *A Focused Ada Investigation*

# Preface

This research was conducted under the auspices of the Research Institute for Computing and Information Systems by Sue LeGrand of SofTech, Inc. under the direction of John McBride, also of SofTech, Inc. Charles McKay, Director of the Software Engineering Research Center at the Univeristy of Houston-Clear Lake served as technical representative for RICIS.

The views and conclusions contained in this report are those of the author and should not be interpreted as representative of the official policies, either express or implied, of NASA or the United States Government.

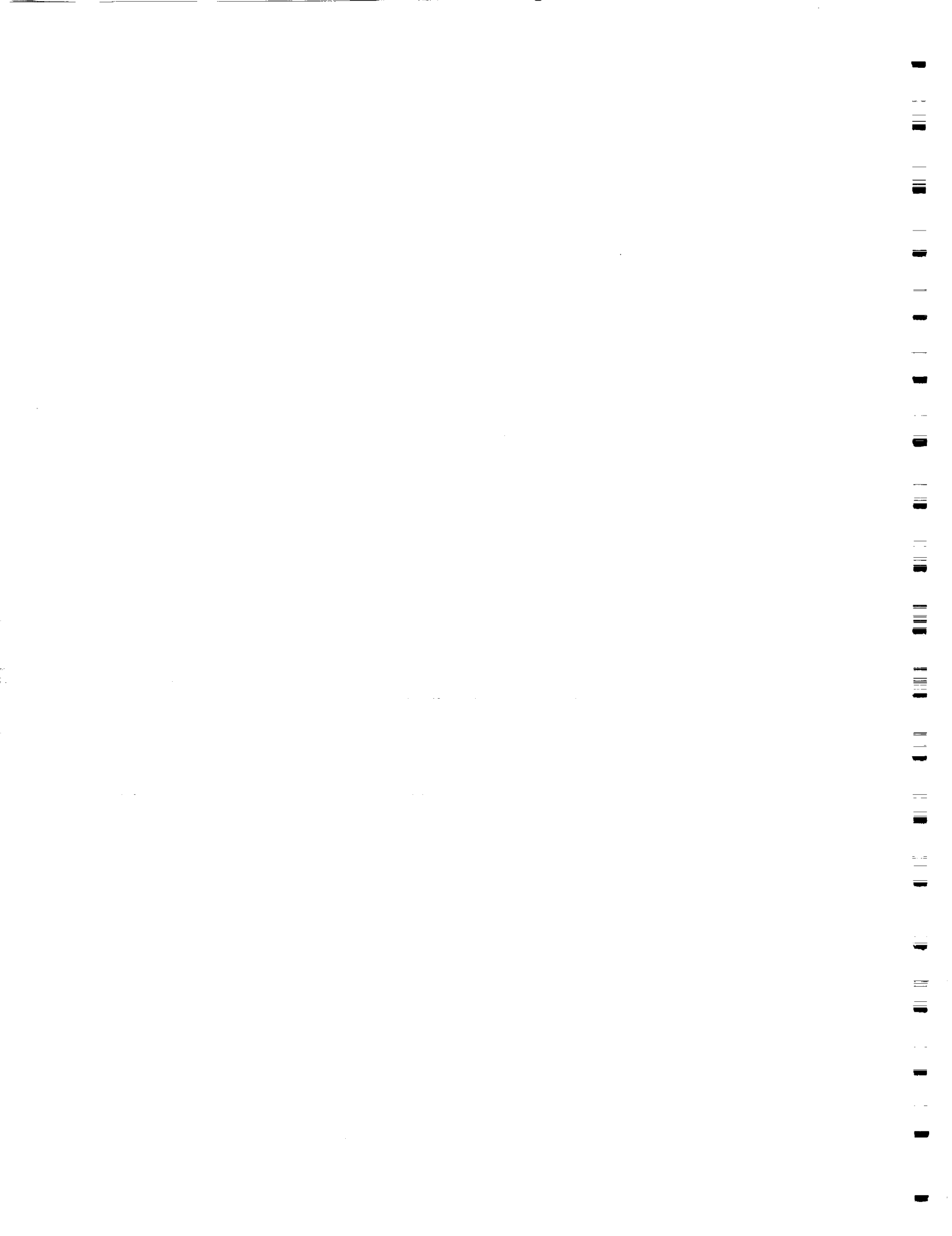Transferring Data Objects

A Focused Ada Investigation

April 30, 1988

RICIS Report, Contract SE.17

Prepared for

NASA Space Station Program Office
NASA Headquarters

Prepared by

SofTech, Inc.
1300 Hercules Drive, Suite 105
Houston, TX  77058-2747

# Transferring Data Objects

## Preface

The use of the Ada language does not guarantee that data objects will be in the same form or have the same value after they have been stored or transferred to another system. There are too many possible variables in such things as the formats used and other protocol conditions. Differences may occur at many different levels of support. These include:

    Program level
    Object level
    Application level
    System level.

A standard language is only one aspect of making a complex system completely homogeneous. Many components must be standardized and the various standards must be integrated.

## Overview

According to a report on A Study of System Interface Sets for the Host, Target and Integration Environments of the Space Station Program [3], the principal issues in providing for interaction between systems are of exchanging files and data objects between systems which may not be compatible in terms of their host computer, operating system or other factors. A related concern is for supporting backup and archiving of data, insuring that a future system upgrade or replacement will not invalidate the data which has been archived. A typical resolution involves at least a common external form, for data objects and for representing the relationships and attributes of data collections.

The issues here are closely related to the issues of interoperability between tools whether they are co-located, concurrently executing, or even different generations of the same tool. Interoperability refers to the ability to exchange information, where information is not only the data involved, but also the proper interpretation of this data. The difficulty is that information is interpreted at many levels in its lifetime. Useful exchange of information requires standards of representation at all levels. Consider the following:

*   At the program level, the design dictates how the information which is to be worked upon will be reflected as a collection of certain data declarations. These alternatives in representation are at the highest level and are dependent on the application domain. As an example, the definition of an intermediate representation for an Ada program is an important design characteristic of Ada compilers which differs substantially in current implementations.

*   At the object level, in its representation as data objects in the program, the information is interpreted by the compiler into a particular representation in the target hardware which is to execute the program.

1
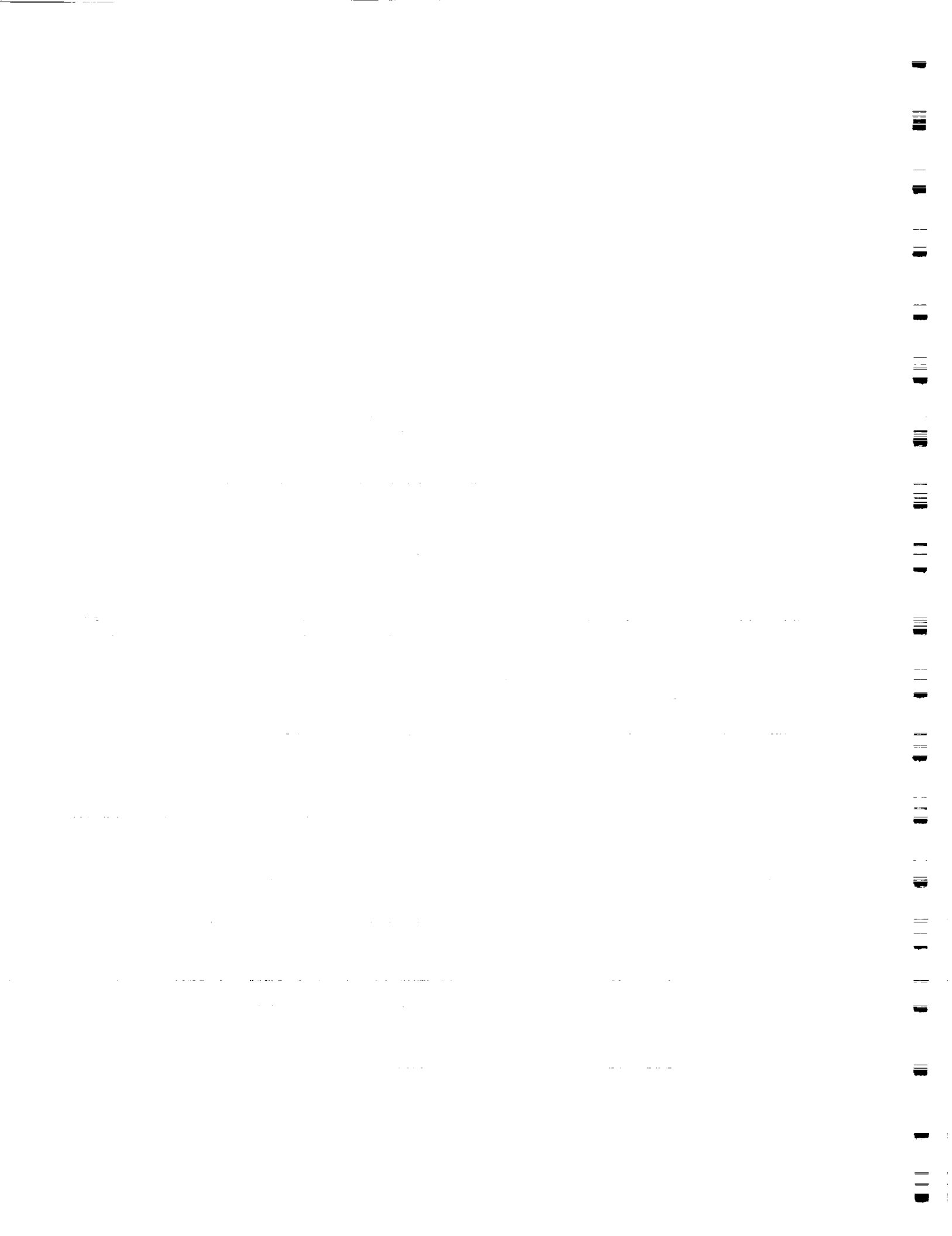
* At the application level, for information exchange, the data must be written out using the standard Ada in/out services. One issue in interoperability is that current standards for external representations do not provide any information about the data structures which have been written (non-self-descriptive file formats). The information cannot be retrieved without implicit knowledge of these data structures and the application which created them.

* At the system level, even without consideration for a self-descriptive file format, the compiler, operating system and physical media all define individual aspects of representation. The individualistic aspects generally prevent any other combination of compiler, operating system, and device from accessing the information, even when using the original source code.

It is recognized that true exchange of information requires a staggering extent of standardization, and that a standard which addresses one level cannot solve all of the problems. A set of standards working cooperatively are needed, either to define system wide uniformity which allows native mode transfer of data, or to define interfaces to which all systems will connect.

A partial list of issues have been raised dealing with the transfer of data objects. It is shown in Table 1. Consideration must be given to how these issues may be handled using the Ada language.
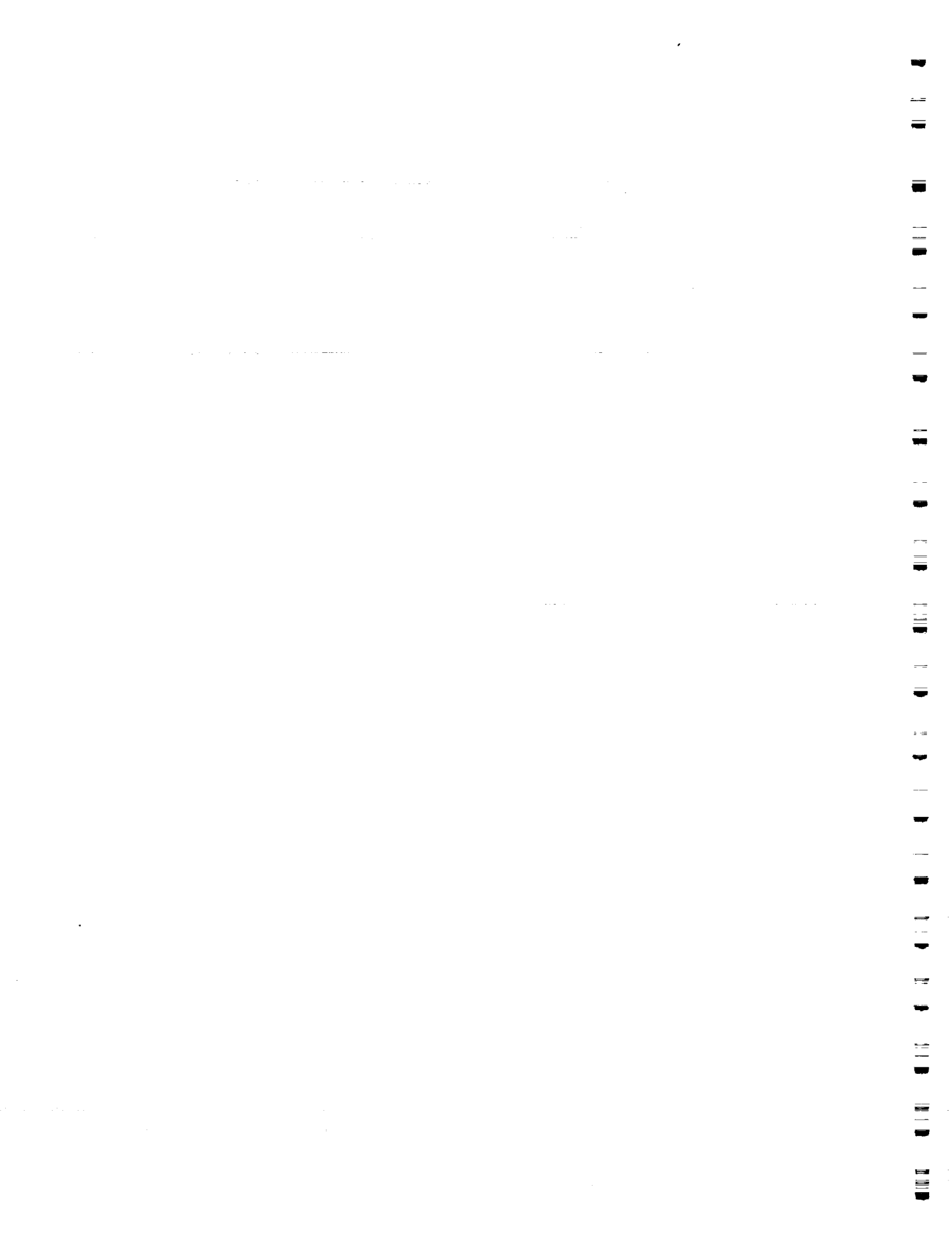
Table 1

List of Issues

1. Data structure

   a. What should be the order of representation?  This includes first listing
      least or most significant bits and rows or columns of data.
   b. How many significant digits should be used to represent a number?
   c. How many bits are available to provide this number accuracy?
   d. Unsigned numbers - How should unsigned numbers be represented and
      manipulated?
   e. There is no standard for representing characters beyond the ASCII set.
      How should other characters be represented?  More than 8 bits are
      required to provide for just the addition of European characters.

2. Resource management

   a. Overloaded queues can be the result of inadequate in/out handling or
      excessive processing requirements.
   b. Do communicating systems use compatible versions of the same resource?
   c. Is archived data usable with the current version of a resource?

3. Data represented in native mode vs. conversion of data

   a. Can one mode be dictated for the entire computer network?
   b. Would a standard system interface require too much elapsed time, CPU
      power and/or memory for processes?

4. Software based data conversion

   a. Will Ada implementations have enough power?
      Can data structures be defined as Ada types?
      Are Ada calls to special functions well enough provided?
      Are Ada machine representation specifications adequate?
   b. Does Ada provide binding to special software?

5. Hardware based data conversion

   a. Can an implementor rely on hardware or specialty system (black box) for
      data object conversion?
   b. Does Ada provide binding to special hardware?

## Approach

The list of issues was sent to each of the known researchers for comment. Each one indicated which issues he was investigating and made comments on his progress. Each one also gave information about his project which is reflected in the updated list of Research Activities in Appendix A. All researchers said that NASA could obtain more information from the contact.

## Summary

The focus of the researchers is shown in Table 2. The numbered columns correspond to the issues listed in Table 1. Data structure was the area of the least amount of work. A few researchers remarked that there is a wide variety of structures among the implementations. A NASA standard should be defined after studying the styles and capabilities of the implementations being considered for use in the Space Station Program. A useful reference is the Transportability Guide, produced by the KAPSE Interface Team (KIT) and due this summer. It includes a list of pragmatic guides for data collection form in order to achieve transportable code. The list contains recommended maximum numbers for such things as:

> Length of identifiers, labels and attributes
> Units in a program library
> Simultaneously active tasks
> Nesting depths

Resource management studies did not include the issue of overloaded queues. NASA will need research in in/out handling and in balancing the processing load. This may be the area needing the most research, since it contains the most unanswered questions. There is a computer testbed at the Avionics Systems Division, Johnson Space Center that may be useful for this. The other issues of resource management did not seem to be a problem to the researchers.

Data representation seemed to be an area of great interest. A slight majority favored converting data, but no one had any assurance that this would work for large amounts of transmitted information. NASA should use the information available from the research activities listed and test the theories with large data loads.

All of the researchers were studying software based data conversion. Many were investigating the impact for good or bad of the special features that Ada provides. This information should prove valuable to the Space Station Program.

Hardware based data conversion was out of the scope of all of the research. Dr. Volz commented that it would have serious memory implications. It would be valuable to the Space station Program to do a market survey to see if any attempt has been made to provide off-the-shelf black boxes for data conversion.
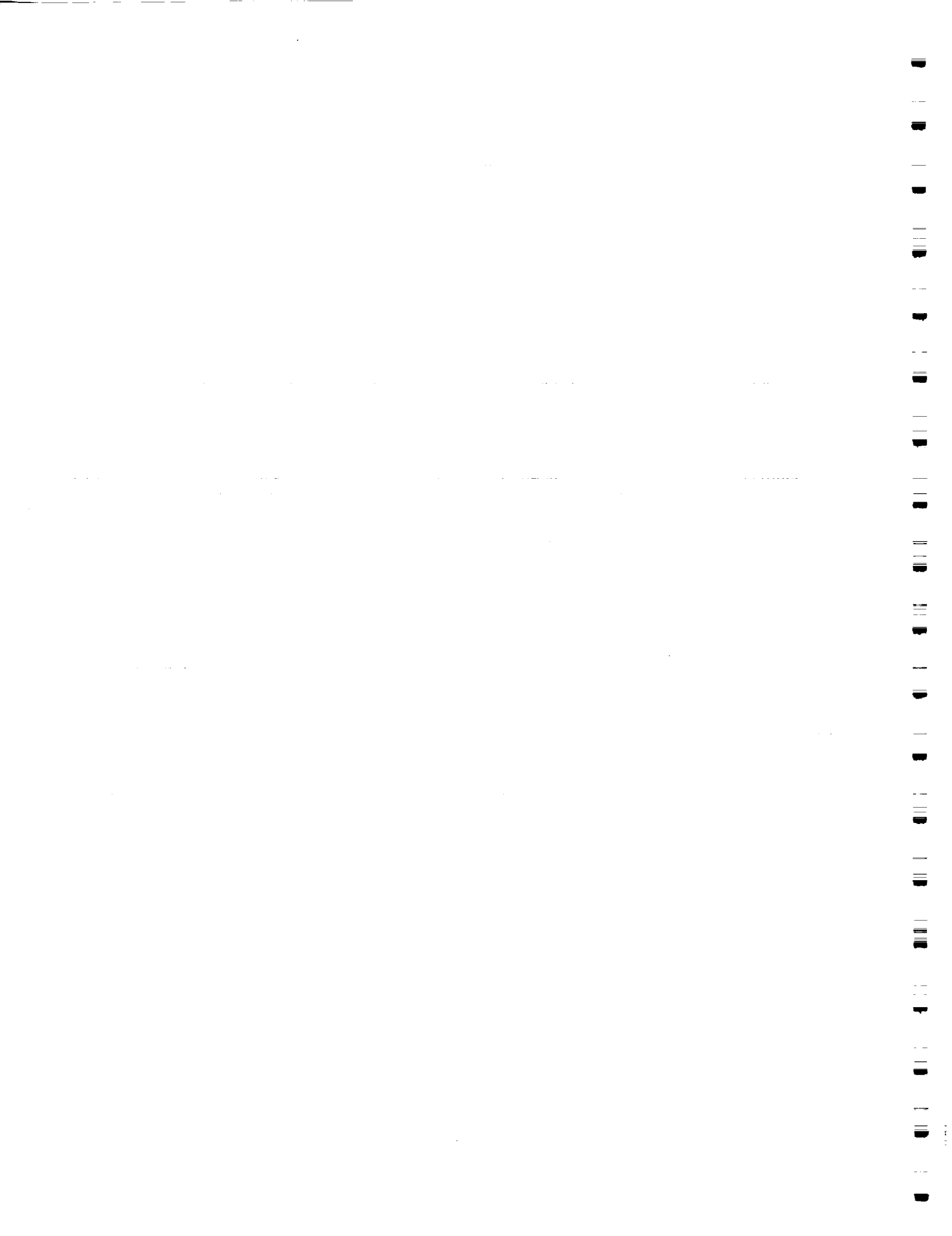
# Table 2

## Issues Being Studied

| Group/ /Issues | 1a | 1b | 1c | 1d | 1e | 2a | 2b | 2c | 3a | 3b | 4a | 4b | 5a | 5b |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Lockheed S. Allen | X | | X | | X | X | | | X | | X | | | |
| MIT B. Liscov | | | | | | | X | X | X | X | | | | |
| MITRE R. Munck | | | | | | | | | | X | X | X | | |
| MITRE D. Emery | | | | X | X | | X | | | X | X | | | |
| SofTech R. Thall | | | | | | | X | X | X | X | | | | |
| SofTech G. Macpherson | | | | | | | | | | | X | | | |
| Tartan Labs E. Ploedereder | | | | | | | | | X | X | X | | | |
| Telesoft K. Bowles | | | | | | | X | X | X | X | X | | | |
| TRW Judy Kerner | | | | | | | | | | X | X | | | |
| U. Mich. R. Volz | | | | | | | X | | X | X | X | | | X |

Appendix A

Research Activities

The following activities have been identified that are of interest to those interested in the issues of transferring data objects with Ada.

**Lockheed**

Project - Network interface between the Space Station subsystems.
Database of Master Measurement Lists of GN&C Test Bed, archival and retrieval. In/out interface handler in Ada. Ada executive handles old FORTRAN programs.

Schedule - First report issued on data over a homogenous network. Second phase begun on organizing data interfaces between subsystems.

Contact:

> Stanley Allen
> 2400 NASA Road 1
> Houston, Texas 77058
> (713) 333-5411

**Massachusetts Institute of Technology (MIT)**

Project - "A Value Transmission Method for Abstract Data Types". Resource management and data conversion are being studied. Ada is not being used at this time.

No Schedule given.

Contact:

> Dr. Barbara Liscov
> MIT Lab for Computer Science
> 545 Main St.
> Cambridge, MA 02139
> (617) 253-5886

**MITRE**

Project - Building a CAIS-A Prototype with multilevel security on a "bare machine" Intel 80386. Data conversion considered.

Schedule - Limited prototype due September, 1988. Next step unfunded.

Contact:

> Bob Munck
> P.O. Box 208 A156
> Bedford, MA 01730
> (617) 271-3671

**MITRE**

Project - Proposed an IR&D project to provide a canonical representation of objects shared by communicating processes. Funding was denied, but Dave had proposed some ideas to develop.

Schedule - One man year effort proposed

Contact:

> Dave Emery
> P.O. Box 208  A156
> Bedford, MA  01730
> (617) 271-2815

**SofTech**

Project - Development of a common external form for CAIS Revision A and a prototype.

Schedule - CAIS-A is in final review.  The prototype is due by the end of 1988.

Contact:

> Rich Thall
> 460 Totten Pond Rd.
> Waltham, Massachusetts  02254-9197
> (617) 890-6900

**SofTech**

Project - Investigate the feasibility of a generic, reusable Internet Protocol in Ada.

Schedule - Begin June 1988 and continue as long as results justify.

Contact:

> George Macpherson
> 5474 Mark Dabling Blvd.  Suite 305
> Colorado Springs, CO  80918-3845
> (719) 528-5155

**Tartan Labs**

Project - Ada compilers and systems development.  Particular attention to description of data streams.

Contact:

> Erhard Ploedereder
> 461 Melwood Ave.
> Pittsburgh, PA  15213
> (412) 621-2210

**Telesoft**

Project - Research for varied Ada compilers.  Remote procedure calls for servers and clients showed need for a pragma interface to the C language.

Schedule - Research completed.

Contact:

> Dr. Kenneth  Bowles
> 10639 Roselle Street
> San Diego, CA  92121-1506
> (619) 457-2700

**TRW**

Project  -  Creating  a  prototype system interface for the NATO Special Working Group on Ada Program Support Environments.

Schedule - Complete prototype due for delivery August 1988.

Contact:

> Judy Kerner
> One Space Park  R2/1020
> Redondo Beach, CA  90278
> (213) 812-0539

**University of Michigan**

Project  -  Telerobot  controller in Ada.  PC-ATs and Appolo on an Ethernet with graphics displays.  Control the robot from any node on the net.  Pack and unpack floating  point  numbers  at  each  node.   Format handled at a low level in/out interface.  Uses a varied record structure.

Schedule  -  Simulation due Fall 1988.  Incorporation of real components due May 1989.

Contact:

> Dr. Richard Volz
> 110 ATL Building
> 1101 Beal Ave.
> University of Michigan
> Ann Arbor, Mich.  48176
> (313) 763-0035

## Appendix B

### Referenced Documents

1.  Allen, S.R. and M.J. Gordon, <u>Transfer of Ada Objects in a Data Communications Network Using a Single Compiler and Instruction Set Architecture</u>, Lockheed Engineering & Management Services Co., December 1987.

2.  Emery, David, "Reliable Object Management for Networked C3 Systems", MITRE, February, 1987.

3.  McKay, Charles, David Auty and Kathy Rogers, "A Study of System Interface Sets (SIS) For the Host, Target and Integration Environments of the Space Station Program (SSP)", University of Houston at Clear Lake, July 30, 1987.

4.  M. Herlihy and B. Liscov, "A Value Transmission Method for Abstract Data Types", ACM TOPLAS 4.4. October, 1982, p 527