

PRODUCT PROJECT DESCRIPTION

**COPYRIGHT (c) 1987 BY
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.**

**Digital Equipment Corporation - CONFIDENTIAL AND PROPRIETARY
FOR INTERNAL USE ONLY**

Change History:

Revision No.	Created	Author
0.1	6/9/87	Robert Bismuth Based on 6/9/87 meeting with Benn Schreiber, David Ballenger and Lou Perazzoli.
0.2	6/12/87	Robert Bismuth Based on 6/11/87 meeting with David Ballenger, Benn Schreiber, Tom Miller, Chris Saether, Jeff East, and Lou Perazzoli.
0.3	6/17/87	Robert Bismuth Misc. typo corrections. Incorporating discussion with John Gilbert and the above list of engineers on 6/17/87.
0.4	7/13/87	Robert Bismuth Incorporate comments from review meetings of 6/19/87 and 6/24/87.
0.5	7/17/87	Robert Bismuth Include MICA discussion section.
0.6	9/10/87	Robert Bismuth Close DECnet, installation, and character cell terminal issues Update for 32 bit support Include recent changes in workgroup dependency Clarify installation goal for DB platform Clarify reliability goal for MICA Replace naem CRDK with QUARTZ Close dev. environment for CXO issue Modify DB platform RPC requirements
0.7	9/21/87	Robert Bismuth Incorporate Myles Connors' comments regarding AIA
0.8	11/3/87	Robert Bismuth Incorporate review comments from Benn Schreiber and Jeff East

TABLE OF CONTENTS

1 Introduction	3
2 MICA Overview	3
2.1 Requirements	3
2.2 Goals	3
2.3 Non-goals	4
2.4 Description	4
2.5 Dependencies	5
2.5.1 Internal Dependencies	5
2.5.2 Corporate Dependencies	5
2.5.3 External Dependencies	5
2.6 Outstanding Issues	6
2.7 Resolved Issues	6
2.8 Impact on Previous MICA Design	6
3 Database Platform Product: Cheyenne	6
3.1 Requirements	7
3.2 Goals	7
3.3 Nongoals	8
3.4 Description	8
3.5 Major Dependencies	8
3.5.1 Internal Dependencies	8
3.5.2 External Dependencies	9
3.6 Outstanding Issues	9
3.7 Resolved Issues	9
3.8 Impact on Previous MICA Design	10
4 Compute Server Product: Glacier	11
4.1 Requirements	11
4.2 Goals	11
4.3 Nongoals	12
4.4 Description	12
4.5 Major Dependencies	13
4.5.1 Internal Dependencies	13
4.5.2 Corporate Dependencies	14
4.5.3 External Dependencies	14
4.6 Outstanding Issues	14
4.7 Resolved Issues	15
4.8 Impact on Previous MICA Design	16
5 Bibliography	16

1 Introduction

This paper outlines two products which the OS Group is building. Briefly, these are:

1. Cheyenne:

A high performance, highly available database machine built out of the QUARTZ database software and a MICA subset operating system running on ROCK systems. \Note that in this document the MICA subset is referred to as the database platform.\

2. Glacier:

A high performance, tightly coupled compute server for DIGITAL's hardware/software base.

Both of products are described here with the assumption that the database platform forms the basis of the compute server.

Both these products are assumed to be built around hardware which implements the PRISM architecture and a base system derived from the current MICA design. This paper also briefly outlines some of the constraints on and design of MICA.

2 MICA Overview

2.1 Requirements

MICA has two basic requirements:

- 1. Immediate requirement:** provide a common subset on top of which the Cheyenne and Glacier products may be built.
- 2. Long term requirement:** provide a state-of-the-art, portable operating system for future DIGITAL computer systems in the mid 1990s. This path is to be evolutionary.

2.2 Goals

- 1. Meet the above two requirements.**
- 2. Robustness and reliability.** In particular MICA must deal with all hardware and software errors in a reliable and predictable fashion.
- 3. Quality:** meet the expectations for a new operating system.
- 4. Schedule.**
- 5. Performance:** goals required for both short and long term requirements.
- 6. Ease of installation and management.**
- 7. Eventual support for all PRISM implementations,** both 32 and 64 bit systems. Immediate support for the MORaine/ROCK processors and packaging.
- 8. Provide a flexible I/O software architecture** which support all ROCK hardware configurations defined as products.
- 9. Provide a flexible system interface** which allows MICA to be extended without modification or addition of component software executing in kernel mode.

\Note that device support and object support are considered exceptions to this.\

10. Provide a programming environment in which DIGITAL's Application Integration Architecture may be implemented as the user applications' interface to MICA.

\This would then eventually give a standard applications interface to MICA which results in source level compatibility for applications between MICA and other DIGITAL operating systems supporting this standard interface.\

2.3 Non-goals

1. Provide tools, documentation and features for user modification of the MICA executive and/or kernel.
2. Constrained by any of DIGITAL's existing operating systems.

\For example, provide direct system service compatibility with any of DIGITAL's existing operating system products.\

2.4 Description

MICA has been designed as an object oriented system with a rich set of services available from user mode (note that most of these services will eventually be accessed via the Application Integration Architecture library). It is designed to support both the 32 and 64 bit PRISM architectures and should require minimal effort to move from a 32-bit only implementation to a 32/64-bit implementation when 64-bit systems become available. Currently, key points in the design are:

1. Priority based pre-emptive schedule with provision for class scheduling.
2. Flexible memory management system which supports all allowed PRISM memory management implementations.
3. Multiple threads of execution within a single address space (i.e. "light weight processes").
4. A layered I/O architecture for the support of physical devices, file systems, and concepts such as volume shadowing, volume striping, virtual terminals, etc.
5. A centralized ACL based security architecture for all objects.
6. Protected subsystems: user processes which act as servers, with amplified security profiles and /or privileges, on behalf of client processes, charging back resource usage to those clients.
7. DECNET Phase 5 support.
8. Workgroup support: a new distributed computing environment which provides file, record, security profile and batch/print queue sharing. \Note that this is initially just support of DFS.\
9. Implemented almost entirely in the PILLAR language which provides block structure, strong typing, structured condition handling and has been designed as a portable system implementation language.

The design of MICA is described in the MICA Working Design Document. This document is currently being developed, given the above requirements.

2.5 Dependencies

\Dependencies are listed mainly for the short term.\

2.5.1 Internal Dependencies

1. The PRISM SRM.
2. The PRISM emulators and their configurations.
3. The XMI Prism development system.
4. Suitable 32-bit hardware and configurations. \The MORAINE/ROCK hardware. Note we eventually need 64-bit configurations.\
5. The 32-bit PILLAR compiler and an eventual 64-bit PILLAR compiler.
6. Continued support and extensions for the 32-bit SIL language until the 32-bit PILLAR compiler is available.
7. A PRISM C compiler and runtime library.
8. A development strategy that allows transition from our current 32-bit PRISM emulator/SIL compiler environment to the 32-bit XMI-system/PILLAR compiler environment, and from there to the MORAINE/ROCK hardware.
9. A testing strategy.

2.5.2 Corporate Dependencies

1. The PRISM Calling Standard: immediately the 32-bit standard.
2. DECNET Phase 5 architecture: both the architecture and the phased development of it for other DIGITAL products.
In addition to these development issues, MICA must be validated for DECNET support.
3. Corporate distributed security strategy, both for workgroup support and for its effect on MICA's security architecture.
4. The FILES-11 ODS-2 specification. \Note that we have an ECO proposed and previously accepted for this but are will require VMS to support this ECO prior to our FRS for either of the two products.\
5. The Applications Integration Architecture. \MICA will need a native implementation of this for software development both internally and externally. This implementation will be phased over various future versions of MICA as parts of AIA become defined.\
6. The corporate RPC. \Required for interface to protected subsystems.\

2.5.3 External Dependencies

1. RTL and debugger support from SDT.

2.6 Outstanding Issues

1. Testing:

Current MICA project plans have not been specific regarding testing strategies. Given the goals of the initial products based on MICA, rigorous testing is a now even more important a requirement of the development plan.

2. PILLAR RTL:

Since PILLAR has very little in the way of built-in functions (such as I/O), an RTL is necessary. This RTL is for internal use only (i.e. during development) and may eventually be replaced by the superset functionality of a native Applications Integration Architecture.

2.7 Resolved Issues

1. Relationship with ULTRIX:

We have resolved that there is no direct ULTRIX compatibility designed into MICA. This does not preclude adding limited compatibility in the form of run time libraries at some later date.

2. Failure modes and effects:

We have planned a document, following after our design, which details the failure modes and effects of our products.

2.8 Impact on Previous MICA Design

Given the current ordering of requirements for MICA is changing the previous design. Each of the two products defined below require only subsets of MICA. A fully functional, general purpose MICA is not required until such time as the corporation needs MICA as a product in its own right.

However, while Cheyenne and Glacier allow us to remove some functional component of MICA from the project, they both require extra functionality, specific to each product, be implemented by MICA.

This is covered below separately for each product.

Finally, MICA will need to support both 32 and 64-bit PRISM architecture machines with a minimal of development effort once 64-bit systems become available.

3 Database Platform Product: Cheyenne

3.1 Requirements

The only requirement of the database platform is that it provide the host environment required by the CXO QUARTZ project. In particular, this implies:

1. Possibly new operating system interfaces and facilities.
2. Specified performance goals, both I/O and operating system overhead.
3. Specified availability goals.
4. The end product must fit in with DIGITAL's OLTP offerings.

3.2 Goals

1. Quality, robustness and reliability.
2. Support of the QUARTZ database software with a clean interface between MICA and the QUARTZ software.
3. Operation of Cheyenne (i.e. platform and QUARTZ software) as a "black box", i.e. a "box" with a concisely defined and specified hardware and software interface for database transactions. Cheyenne must appear as a product, not a collection of products.

\This does not imply that the platform would be built out of a single ROCK system. This "box" might in fact be composed of several ROCK systems connected by a high performance interconnect. This helps to insure availability goals and also an extensible system with a reasonable price/performance scale.\

4. "Online" software installation or upgrade: replacement or modification of component parts while the entire system is running with minimal service interruption.
\\The current vision is that system software upgrades will take place while the system is running: to use the upgrade, a ROCK will need to be rebooted.\
5. Performance as required for the QUARTZ product: capable of supporting up to 600 tps in a fully configured system.
6. A host environment that gives the reliability, availability, and fault tolerance required for the QUARTZ product. \\Figures presented for QUARTZ are: 99.95 availability.\
7. Large amounts of disk storage: QUARTZ has the goal of simultaneous support of up to 2100 HDAs giving 300 Gbytes total storage: one third large disks, two thirds small disks, all configured as 150 Gbytes of shadowed storage.
8. Simplified system management, including remote system management.
9. The Cheyenne is easily expandable from a basic hardware configuration to the top end configuration which meets or exceeds the above performance, availability and storage capacity goals.
10. Eventual expansion to include a TP monitor.

3.3 Nongoals

1. Access or use without a client system.
2. Access to processes on the platform other than database system processes. \Note that database system processes also include processes required for system management.\
3. Use of the underlying system as a compute server.

3.4 Description

The primary design center of the database platform is QUARTZ. The platform supplies all the operating system services which the QUARTZ software requires.

The platform is a MICA-subset which must minimally be capable of:

1. Supporting free running server processes.
2. Supporting the security model required by QUARTZ.
3. Implementing the specific network transport and/or protocol required for communication with QUARTZ processes.
4. Supporting a diagnostic environment for hardware diagnostics.

In addition, QUARTZ requires that MICA support a common logging facility and high performance interprocess communications. Note that these facilities must function in a multi-ROCK Cheyenne configuration between the individual ROCK systems.

Minimally, in addition to the QUARTZ requirements, MICA will need to provide remote system management, and high speed disk/file access.

The current design model consists of a Cheyenne system composed of multiple ROCK systems interconnected by high speed communication links. Through MICA, QUARTZ would distribute its work load via these links and thus meet the performance, availability and connectivity goals. Communication from client systems to the Cheyenne system would have no application visible knowledge of the internal organization or topology of ROCK systems making up the configuration.

3.5 Major Dependencies

3.5.1 Internal Dependencies

1. MICA: the subset needed for this product, and all its dependencies.

3.5.2 External Dependencies

1. QUARTZ.
2. Front-end software for access to QUARTZ servers. \The QUARTZ software will implement data management and a protocol for data access via LAN. To sell this as a product, there clearly need to be front end packages which run on client systems and use the QUARTZ/MICA system. Currently no group in DIGITAL is developing such packages. It is our responsibility to make sure that plans are in place for this area so that DIGITAL has a complete product.\
3. The corporate RPC standard. \This is required by the system management model.\
4. DECnet Phase 5: the QUARTZ network interface.
5. DECnet performance over the CI to/from VMS client systems.

3.6 Outstanding Issues

1. System management:

System management is carried out via a remote system management interface from a client system. The details of this and the role of the system console in this are in design. That QUARTZ will use the system management interface to activate utilities. The interface must be extendable in this respect.

2. Security:

What is the QUARTZ security model and how does it relate to the corporate distributed security model?

3. Transaction processing platform:

It has been claimed that the database server should be the basis of a future transaction processing (TP) platform. Since TP has particular requirements and constraints, we should investigate current TP models to ensure that our base system design does not prevent a platform from being built in the future.

\There are examples in VMS today of original design decisions for the base system which later made it impossible or difficult to implement efficient TP systems on VAXes.\

3.7 Resolved Issues

1. Development environment for CXO:

The development requirements for CXO have been identified and plans made to deliver the required environment in the right time frame.

2. High speed interconnect for use between ROCK systems in a single Cheyenne configuration:

The CI has been selected as the interconnect device.

3. Files-11:

QUARTZ will use Files-11 volumes for files. MICA shadow set support will also be used.

4. Backup and restore facilities:

The QUARTZ group will provide backup/restore/rebuild facilities for database files.

5. Network support:

DECnet will be the network software implemented for Cheyenne and used by QUARTZ.

6. Transport protocols:

The only transport protocols used by Cheyenne are DECnet protocols and SCA.

7. Client systems:

DECwest has no currently planned role in any client software or client systems.

8. IPC:

High performance interprocess communication is essential to the QUARTZ design. We are currently designing this facility to meet their needs.

9. Fault tolerance:

We are dealing with the requirements of fault tolerance by providing the capability of multiple ROCK systems within a Cheyenne configuration. ROCK systems have the following requirements:

1. 100% data integrity.
2. Every system failure is detected and the source identified. \ "System" here includes hardware and software.\
3. Software must provide failover, reconfiguration and graceful degradation.

3.8 Impact on Previous MICA Design

The following facilities and areas of MICA are not believed to be needed for the Cheyenne MICA-subset operating system:

1. All utilities (both VMS and ULTRIX), with the exception of:
 1. The linker and PILLAR compiler.
 2. Some SET/SHOW/MONITOR functions. \Perhaps just an RPC server for these on the database platform with the actual command utilities integrated on the client systems.\
 3. BACKUP. \Not for database recovery - for entire volume backup and for installation.\
 4. Some system management utilities.
2. Terminal driver (except console support).
3. DCL, shells or any command interpreter.
4. Job controller: no batch or print management, very little (if any) local job management because QUARTZ processes are managed by QUARTZ and not by users directly. The only other processes in a Cheyenne system would be system management processes which are also very limited and controlled.

It is very likely that the Process Architecture should be reviewed in light of the QUARTZ model.

5. Local VMS or ULTRIX compatibility libraries as defined for MICA today.

6. Vector support.
7. SDT languages and a large part of the RTLs.
8. Workgroup/DFS support.
9. RMS ISAM support.

MICA is not a highly available system. It does meet certain goals of behavior in failure modes and it also guarantees data integrity. MICA must provide the basis to allow QUARTZ to be made highly available by using multiple MICA systems. The fundamental tool for this is a high speed inter-rock CI based IPC mechanism.

4 Compute Server Product: Glacier

4.1 Requirements

We are expecting more specific requirements to be generated by our Product Management Group.

However, one requirement currently identified and assumed in this document is that compute servers and clients must be members of the same Workgroup.

4.2 Goals

1. Quality, robustness and reliability.
2. Seamless integration with client systems:
 1. Application compatible programming interfaces between client operating systems and Glacier's MICA operating system.
 2. Provide transparent image activation on the compute server from client systems. \This should make Glacier essentially an extension of a client.\
3. Client systems are VAXes, in particular workstations. \There is some discussion of other future client systems, VAXmates for example.\
4. The client system interface to the compute server is designed for future support by multiple operating systems.
5. The programming interface for Glacier is DIGITAL's Application Integration Architecture.
6. Performance: both processor, memory access/addressing and I/O speed/capacity. \Note that no specific goals have been identified in this area.\
7. Glacier provides parallel and vector processing capabilities.
8. Provide the basis from which a full functionality MICA operating system will be implemented as part of our new-architecture long range goal.
9. Provide an easy to use, simple, and consistent user interface for system management.
10. Layer support for the Glacier entirely on top of a client operating system with no internal changes. \This may be in conflict with goal 2 and tradeoffs may occur.\

4.3 Nongoals

1. Access to or use of the compute server without a client system. \Note that we do not include a LAT Server as a potential client system.\
2. Full participation in local or wide area networks as a node distinct from any client nodes.
3. Implement MICA system services that are completely compatible with VMS or ULTRIX system services.
4. A DECwindow server distinct from any client system. \This is a non goal because DECwindow services are provided by client systems, not the compute server itself.\
5. Making MICA system services directly available to applications.

4.4 Description

The current model of Glacier is a system which derives most of its programming environment and some of the programming interface and from its client systems. Glacier itself has system service capabilities and also uses services in its clients to complete requests from application images running on the compute server. This model is described in:

```
docd$: [MICA.papers.rpc-compute-server] rpc-compute-server.paper
```

The associated files in that directory discuss specific aspects of the model. What follows is a brief description of this model.

Glacier is based on a PRISM/MICA system that is connected to its client systems by a high-speed LAN. The server is accessed by client systems and largely acts as a slave to those systems, becoming an extension of each client system. A process, created by a client system, in the compute server, is always associated with the job in the client system that created it. Such processes are called bound processes.

Each bound process can make use of two types of system procedures: those which use native MICA system services and those which make RPC calls back to the client system. In this way, a bound process is given control over MICA and PRISM specific facilities (such as memory management, multiprocessing, scheduling, etc.) while having capabilities and context of its associated job/client system available (such as logical name context, command information, user information, etc.).

In order to meet the goal of application transportability between client and server, bound processes may only use services provided by DIGITAL's Application Integration Architecture. AIA is designed to provide a system-independent implementation of system services/facilities. These include most of the basic services an operating system normally directly provides, plus facilities like DECwindows. This implementation of the AIA is built on top of the two possible types of system procedures described above.

Key to the Glacier design is that activating an image on a compute server should be indistinguishable from activating an image on a client. Therefore, a bound process has the following properties:

1. Its maximum life span is that of the process which created it.
2. It has the same effective security profile as the job/process which created it. \The FRS product will use DECnet proxies for this.\

3. It is associated with a specific process running on the client system in the associated job. That process forms the bound process' RPC server for nonlocal system procedures.

Thus, from a client's point of view, Glacier is a tightly coupled extension of its computing resources. In the current model, no client is aware of any other client's use of the compute server: bound processes on the compute server are unaware of each other.

From Glacier's point of view, in addition to bound processes (which are essentially slaved to client jobs) there are also free-running processes. These are necessary to implement server processes for facilities such as system management and workgroups. Free-running processes have the following important characteristics:

1. All the native MICA system procedures are available to them but they have no remote client procedures available since they are not associated with any particular client.
2. While they may be started via a special mechanism from a client, they have a life which is not implicitly linked to any job on any client. \Note that this makes the management and failure detection of these processes nontrivial.\
3. They are not automatically tied to the security profile of any client job.
4. Specific privileges are required for a client to create them.

The only interface to Glacier is via a client system. Client systems access the compute server via the RPCs' transport and via workgroup support (i.e. DFS for FRS). The RPC transport is bi-directional: client systems use it to create processes and compute server processes use it to make calls to client systems. Glacier has no directly connected terminals (other than the system console), so that all terminal usage by compute server processes is via the DECwindows component of the AIA or via callback RMS.

File access for a compute server is either provided by the local file system to locally attached disks, or via DFS support to disks located on client systems. It is implicit in this model that the compute server is in the same workgroup as its clients. DFS will provide file access services for FRS and is eventually expected to provide both record services, and the complete workgroup distributed security environment.

A compute server may also make its local disks available to processes running on client systems via the workgroup.

DFS and RPC transports are the only networking requirements which explicitly need to be implemented on the compute server. Other LAN/WAN access by applications on the compute server is obtained via callbacks to client systems or via using the DECnet facilities provided by MICA.

4.5 Major Dependencies

4.5.1 Internal Dependencies

1. MICA: the subset needed for this product.
2. DFS architecture implementation.

4.5.2 Corporate Dependencies

1. A satisfactory RPC standard and the tools to use and implement that standard.
Satisfactory means: capable of being used in our model. This may mean greater involvement in the review/specification of the corporate RPC standard.\
2. DIGITAL's Application Integration Architecture.
This is currently the biggest unknown quantity. Its central role in our model implies our future participation in its design and specification. For Glacier, it is partially a native MICA issue and partially a distributed issue.

The MICA group has the opportunity to help specify this architecture. However, the architecture's success depends on management commitment within other development groups at DIGITAL. We must monitor this.\
3. Corporate distributed security strategy: critical for post FRS workgroup support.
This is now a corporate wide issue under the direction of Butler Lampson with the major authors from SRC.\
4. Corporate ULTRIX strategy. **This item has far ranging consequences** since it is unclear what exactly the future strategy is. However, it is now clear that ULTRIX systems will be clients of Glacier, post FRS.\
5. DFS. As DFS V2 becomes a reality, we will need to track and implement it, together with any changes which come out of DRG review.

4.5.3 External Dependencies

1. Support of DFS by other client operating systems.
2. Distributed debugger support from SDT.
3. SDT compilers. **Not for development of the product, but for availability at FRS.**\
4. The DFS product for post FRS workgroup distributed record and file access services.

4.6 Outstanding Issues

The following are the open issues surrounding our current model:

1. Division of system services between local and remote:
Some analysis of this has been done, but little is understood about the relative efficiency versus implementation costs per service. This needs to be looked into, possibly via examining sample applications which might be ported to Glacier or by prototyping some services once the tools required to implement RPCs are in place.
2. Changes in client operating systems:
As yet, no detailed analysis has been made of how to implement the requirements of this compute server model in any potential client operating systems. It is possible that compute server support can be layered on top of any client system, as desired. If not, any changes required in client operating systems must be identified and a plan agreed upon with other development groups for their implementation.

\DECwest owns the "total system" problem for this product.\

3. Quotas and accounting:

Should quotas for a bound process originate from the client system or should they be defined on Glacier? Is accounting information only returned to the client system when a bound process exits? Is it captured on Glacier?

4. Application availability:

Currently no applications running on DIGITAL supplied systems use the Application Integration Architecture. Given that this architecture is currently being design and is expected to take some time to be available to customers, it is unlikely that there will be many applications available at FRS of Glacier which use this architecture.

Thus, to provide some initial attraction for the product, it may be necessary to provide support for some VMS or ULTRIX system services via the RPC techniques described above. To resolve this issue, applications typical of those which would be run on Glacier should be studied for their requirements in terms of host services.

5. Failure modes and behavior:

Little in-depth study has been made of dealing with server and network failures. These should be completely characterized for the product and documented. The question of failover (thought to be too large a problem for FRS) should be at least characterized and future product goals identified.

6. RTL versus AIA:

It is unclear exactly what AIA will include. Currently, AIA is known to include the SMG\$ and parts of the LIB\$ VMS libraries. It may also include the general PILLAR RTL.

We must resolve what general RTL support is required. If AIA includes what currently is defined as our RTL, then implementation and definition will continue as a part of the MICA AIA project. If not, we must plan and produce the RTL.

4.7 Resolved Issues

1. The availability of general RPC support for user written applications:

The FRS product will not make general RPC support available to user written applications. RPC support will be design with that goal in mind for some future release.

2. Transport protocols:

The only protocols implemented for Glacier are DNA protocols and the DFS protocols.

3. Network support:

DECnet will be implemented on Glacier.

4. Model for system management:

The system management model has been selected and is in design.

5. Scaling:

The issue of scaling has been largely postponed to a future release. For now, when a client may use multiple compute servers, the target server will be selected by a fairly simple mechanism (such as a logical name).

Support for application fan out or load balancing is independent of the basic compute server design and thus need not be included for FRS.

6. Character cell terminal support:

Glacier will provide support for character cell terminals. This support will be minimal line I/O, unless requirements show a need for more functionality. The maximum amount of functionality which would be implemented is the SMG library.

7. Client operating systems:

Both ULTRIX and VMS are client operating systems for this product. \Note that there may be only 1 at FRS.\

4.8 Impact on Previous MICA Design

The impact on MICA includes all points mentioned for the database platform. However, Glacier requires the following MICA facilities be implemented on the database platform subset operating system:

1. The Application Integration Architecture library. \Note that the DECwindow part would be RPC stubs.\
2. Workgroup support (note that this implies support of DFS).
3. Vector support.
4. SDT languages, RTLs, etc.
5. The C compiler and C-RTL.

5 Bibliography

A large part of the input for this document comes from the research papers located in:

docd:[MICA.papers...]

on the software development cluster.