AD-A252 445

INTEGRATED INFORMATION SUPPORT SYSTEM (IISS)
Volume II - Project Overview
Part 1 - Technical Summary

M. Foster, C. Hedge, J. Althoff

Control Data Corporation
Integration Technology Services
2970 Presidential Drive
Fairborn, OH 45324-6209

DTIC
SELECTED
JUN 0 3 1992
S B D

September 1990

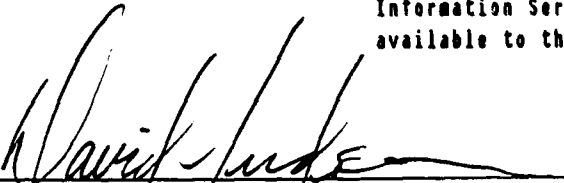Final Report for Period 1 April 1987 - 31 December 1990

92-14512

92 6 02 013

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever, regardless whether or not the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data. It should not, therefore, be construed or implied by any person, persons, or organization that the Government is licensing or conveying any rights or permission to manufacture, use, or market any patented invention that may in any way be related thereto.
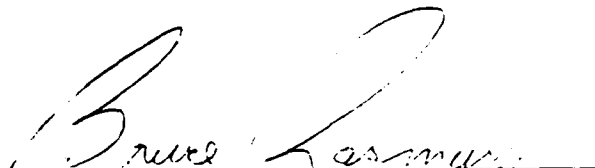
This technical report has been reviewed and is approved for publication.

This report is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations

DAVID L. JUDSON, Project Manager
WRDC/MTI
Wright-Patterson AFB, OH 45433-6533

DATE

FOR THE COMMANDER:

BRUCE A. RASMUSSEN, Chief
WRDC/MTI
Wright-Patterson AFB, OH 45433-6533

DATE

If your address has changed, if you wish to be removed form our mailing list, or if the addressee is no longer employed by your organization please notify WRDC/MTI, Wright-Patterson Air Force Base, OH 45433-6533 to help us maintain a current mailing list.

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified		1b. RESTRICTIVE MARKINGS		
2a. SECURITY CLASSIFICATION AUTHORITY		3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for Public Release; Distribution is Unlimited.		
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE				
4. PERFORMING ORGANIZATION REPORT NUMBER(S) FTR620300002		5. MONITORING ORGANIZATION REPORT NUMBER(S) WRDC-TR-90-8007 Vol. II, Part 1		
6a. NAME OF PERFORMING ORGANIZATION Control Data Corporation; Integration Technology Services	6b. OFFICE SYMBOL (if applicable)	7a. NAME OF MONITORING ORGANIZATION WRDC/MTI		
6c. ADDRESS (City,State, and ZIP Code) 2970 Presidential Drive Fairborn, OH 45324-6209		7b. ADDRESS (City, State, and ZIP Code) WPAFB, OH 45433-6533		
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Wright Research and Development Center, Air Force Systems Command, USAF	8b. OFFICE SYMBOL (if applicable) WRDC/MTI	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUM. F33600-87-C-0464		
8c. ADDRESS (City, State, and ZIP Code) Wright-Patterson AFB, Ohio 45433-6533		10. SOURCE OF FUNDING NOS.		

11. TITLE (Include Security Classification) See Block 19	PROGRAM ELEMENT NO. 78011F	PROJECT NO. 595600	TASK NO. F95600	WORK UNIT NO. 20950607

12. PERSONAL AUTHOR(S) Control Data Corporation: Foster, M., Hedge, C., Althoff, J.				

13a. TYPE OF REPORT Final Report	13b. TIME COVERED 4/1/87–12/31/90	14. DATE OF REPORT (Yr.,Mo.,Day) 1990 September 30		15. PAGE COUNT 62

16. SUPPLEMENTARY NOTATION WRDC/MTI Project Priority 6203				

17.	COSATI CODES		18. SUBJECT TERMS (Continue on reverse if necessary and identify block no.)		
FIELD	GROUP	SUB GR.			
1308	0905				

19. ABSTRACT (Continue on reverse if necessary and identify block number)

This document describes the technical enhancements made to the Integrated Information Support System (IISS) during the final project reporting period.

```
Block 11 - INTEGRATED INFORMATION SUPPORT SYSTEM (IISS)
Vol II - Project Overview
Part 1 - Technical Summary
```

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED x SAME AS RPT. DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified		
22a. NAME OF RESPONSIBLE INDIVIDUAL David L. Judson	22b. TELEPHONE NO. (Include Area Code) (513) 255-7371	22c. OFFICE SYMBOL WRDC/MTI	

DD FORM 1473, 83 APR
    EDITION OF 1 JAN 73 IS OBSOLETE

FOREWORD

This technical report covers work performed under Air Force Contract F33600-87-C-0464, DAPro Project. This contract is sponsored by the Materials Laboratory, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio. It was administered under the technical direction of Dr. Walter H. Reimann, Branch Chief, Manufacturing Technology Directorate, through Mr. David L. Judson, Project Manager. The Prime Contractor was Integration Technology Services, Software Programs, Professional Services Division, of the Control Data Corporation, Dayton, Ohio, under the direction of Mr. W. A. Osborne. The DAPro Project Manager for Control Data Corporation was Mr. J. P. Maxwell.

The DAPro project was created to continue the development, test, and demonstration of the Integrated Information Support System (IISS). The IISS technology work comprises enhancements to IISS software and the establishment and operation of IISS test bed hardware and communications for developers and users.

The following list names the major Control Data Corporation subcontractors and their contributing activities:

SUBCONTRACTOR	ROLE
Arizona State University	Responsible for test bed operations and support in Arizona.
Control Data Corporation	Responsible for the overall Common Data Model design development and implementation, IISS integration and test, and technology transfer of IISS.
D. Appleton Company	Responsible for providing software information services for the Common Data Model and IDEF1X integration methodology.
International Business Machines	Responsible for studies in Enterprise Integration Framework.
Northrop Corporation	Responsible for studies in Enterprise Integration Framework.
ONTEK	Responsible for defining and testing a representative integrated system base in Artificial Intelligence techniques to establish fitness for use.
Simpact Corporation	Responsible for Communication development.
Structural Dynamics Research Corporation	Responsible for User Interfaces, Virtual Terminal Interface, and Network Transaction Manager design, development, implementation, and support.
Universal Technology Corporation	Responsible for test bed operations and support in Dayton.
Northrop Aircraft Division	Responsible for studies in Enterprise Integration Framework.
TCS Consulting	Responsible for OSI communications development.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

SECTION 1

INTRODUCTION

1.1 Background

Since 1979, projects have been supported to design and develop a prototype of an integrated computer system called the Integrated Information Support System (IISS). The Data Automation Processor (DAPro) project was created to continue the development, testing, and demonstration of IISS. The objective of the DAPro project is to establish and operate a Test Bed, to validate the concept of integrated applications supported by IISS, and to maintain and enhance IISS.

DAPro is a continuation of technical results from Project 6202 and its predecessor Project 6201. The initial objective of Project 6201, ICAM Integrated Center (ICENT) Manufacturing Control Material Management (MCMM) System, was to establish the Requirements Definition and Detail Design of an ICENT-level MCMM System. As work proceeded, it became apparent that accomplishing this objective required an integrating mechanism that allowed the integration of higher-level planning systems, with lower-level shop floor control systems.

Work under ICAM Project 3101, Computer-Based Information System (CBIS), established a number of principles as guides in formulating solutions for the near term that are extendible for the long term. The solution to the ICENT integration issue appeared to be generic to the entire spectrum of system and data integration. It was apparent that undertaking IISS would ultimately yield not only the solution to integrating high-level and shop floor systems but to the entire spectrum of integrating existing enterprise systems and future systems.

Project 6201 and 6202 objectives were to establish and operate a Test Bed to validate the concept of integrated applications supported by establishing an IISS. In addition, the projects established a standards guideline document called the "Interim Standards and Procedures" to guide the design of the IISS and to provide guidance to other ICAM projects. A set of requirements and a migration strategy were established as the basis for enhancement direction to the IISS.

The DAPro objective, under the prime contractor Control Data Corporation, is to establish, operate, and enhance a Test Bed containing services provided under Project 6201 and 6202. This project will provide technical and educational support for IISS users representing several manufacturing technology centers and disciplines, including integration technology, sheet metal, machinery, composites, electronics and assembly. Each center could be supported by the Test Bed, which includes system development, tests, production emulation, and technology transfer environments.

1.1.1 Today's Non-Integrated Environment

Today, factories are characterized by a multiplicity of discrete information systems that have been designed to serve individual users or activities within an organization. Since many of these activities receive information from and generate information for many other activities, an extremely complex non-integrated information system has been developed.

Additionally, today's environment is characterized by a number of different computers that do not easily communicate, have different keyboard access methods, and require expertise in order to use them. These conditions lead to a reluctance to implement other types of software and equipment.

While capital spending for computer equipment and software is soaring, and information costs are mounting rapidly, the current environment results in the poor treatment of data and a waste of information, which is a valuable corporate resource. This situation becomes more complex and difficult to contain as more discrete systems are installed.

The Air Force's Manufacturing Technology Directorate at Wright-Patterson Air Force Base (WRDC/MTI) recognized the advantages and requirements for integrated computer systems. The Air Force realized that an integrated computer system would reduce the cost of these manufacturing processes. Therefore, the Air Force has been supporting the design and development of a prototype of an integrated computer system, the IISS, since 1979. The DAPro project was created to continue this effort of the development, testing, and demonstration of IISS.

1.1.2 What is the Integrated Information Support System?

IISS is a software system that provides the user with a single view of data, even though that data might reside on one or more databases on one or more computers. The system under IISS appears as a single computer system with one central repository of data. Users and applications access this data without regard to its location, the type of terminal available, or the format of the stored data. Under IISS, the user does not have to be familiar with a specific database management system or a specific vendor's hardware and software to access data. IISS provides the tools to allow the business enterprise to define and control data, enforce data integrity, and provide data shareability, data quality, data timeliness, and ease of use.

IISS is an evolutionary approach to the design of an information system. This approach allows new computers and software systems to be added or implemented within the organization as required.

IISS uses the three-schema ANSI/SPARC concept developed for integrated environments. Using this concept, data is available to users regardless of the system environment.

1.1.3 Test Bed

To develop the IISS software, prove the concepts, and demonstrate the system's use, a combination of hardware, software, and communications have been established. This combination is called the IISS Test Bed.

The Test Bed hardware currently consists of two interconnected computers: a VAX and an IBM. The VAX is used as the main computer for user interface and central database access. The computers are interconnected using a local area network and wide area communication facilities. The wide area communications include leased lines, dial-up lines, multiplexors, and modems. The Test Bed allows access for system development, technology transfer, and implementation of advanced technology centers, which expect to use this environment to validate production prototypes.

1.2 Goals of DAPro and IISS

It is estimated that in large U.S. corporations most of the existing computer applications will be redesigned over the next 10 to 20 years. It is further expected that, because of the rapidly changing computer technology, the construction techniques and operation modes of new applications will bear little resemblance to those of existing systems.

Because of the complexity of these new systems, integrating them becomes even more important. However, data integration must be accomplished using new application techniques and designed for interaction in conjunction with existing applications. The IISS software architecture allows for an integrated application process that can access data existing on one or more databases and computers.

IISS development and the DAPro project are required to meet several goals:

1. Provide a testing facility for separate Computer Integrated Manufacturing (CIM) software products.

2. Demonstrate initial integration of CIM products

   * Data integration via the CDM and IDEF modeling methodology.

   * Techniques for more extensive integration of program functions.

3. Provide a site for demonstration and evaluation of CIM products

   * Applications.

   * Methodologies.

   * Information support system.

4. Reduce risk to subsequent users of CIM products by providing an environment to test the concepts of IISS.

5. Provide standards, guidelines, and procedures

   * For development of CIM products.

   * For evaluation/adoption by industry.

6. Demonstrate strategies evolving from current application processing and development methods and techniques that will subsequently reduce cost and increase system flexibility.

1.3 Summary of Benefits of IISS

DAPro is a support service that assists others in achieving the benefits of an integrated environment using IISS technologies and concepts. Other projects using the concepts can implement CIM systems faster and with less risk by providing a test environment to prove applications concepts in an integrated environment.

The design of IISS provides the following capabilities:

- Portability between computer systems.

- A common data management system that provides one common method for defining and accessing data from the system regardless of location.

- A common data definition language to define data within the system.

- A common data manipulation language to allow standard access to data without regard to location, access system, or format.

- A common user interface to allow one central type of access to the system from multiple terminal types.

- Computers to communicate throughout the IISS environment. Data access is not bounded by hardware or software systems.

- Data that is managed by IISS software as it is transmitted throughout the system. This allows IISS to be responsible for data transmission instead of the application program.

- Computers that appear in the IISS environment as one computer. IISS is responsible for accessing different hardware systems. It is not the responsibility of the programmer or the user to know the characteristics of each computer being accessed.

Benefits of IISS include:

- Distributed heterogeneous systems, distributed data, and distributed processing is provided by IISS.

- Independence of application data from considerations of actual internal storage organization and database management system access techniques is available.

- Reduced and controlled data redundancy.

- Automated data validation, assertions, triggers and constraint checking through the Common Data Model.

- Transaction-oriented applications.

- Standardized user interface (similar menu construction for all applications, standard user "HELP" procedures, standard error messages, etc.).

- Control of execution in a consistent manner of processes on different computers with different operating systems.

- Facilitation and control of passing data and messages between processes on the same or different computers.

- Consistent error handling throughout the system.

- System-wide control of startup, shutdown, restart, and recovery.

- Application programs written with relational database languages referencing non-relational databases.

- Independence of application program from the characteristics of the terminal on which it will be used.

- System-supported translation of information formats to host-specific representations.

- Standardized electronically generated documentation.

1.4 List of Tasks and Current Subtasks

The following list of tasks will be used as the basis for subject content reporting. The Statement of Work (SOW) tasks have been broken down into project numbers to facilitate a specific delineation of the work.

TASK NO.	SOW. NO.	DESCRIPTION
1100	4.1	Project Management
1115	4.1	Principal Investigator
1130	4.1	MT-NET Management
1293	4.6	CDM IDEF1X
1294	4.6	IRDS
1295	4.6	IISS/PDES
1400	4.4	Tech Transfer
1410	4.4	System Architecture
1415	4.4	Enterprise Integration Framework
1420	4.3	Workshop Preparation
1421	4.5	Workshop Delivery
1430	4.4	IDEF Users Group
1453	4.4	CALS Expo '88
1454	4.4	CALS Expo '89
1460	4.4	Public Relations
1471	4.5	Navy Support
1478	4.5	NIST Install/Support
1479	4.5	ALD Support
1490	4.5	ML-NET Network and Maintenance
1491	4.5	ML-NET Travel System
1492	4.5	ML-NET User Support
1493	4.5	Electronic Documentation System
1494	4.5	Device Drivers
1496	4.5	NTM/OSI
1500	4.3	Project Documentation
1510	4.3	Monthly Status Reports
1520	4.3	Interim Technical Reports
1530	4.3	Final Technical Reports
1610	4.2	Operate and Maintain Test Bed - ASU
1615	4.2	Operate and Maintain Test Bed - UTC
1710	4.3	Testbed Configuration Management
1720	4.3	Testbed Database Administration
1730	4.3	Testbed Shutdown

1825	4.6	Release 2.4/3.0 Specification
1890	4.6	Release 2.4/3.0 Documentation

PLMM		SOW 4.7

2010	System Analysis and Program Management
2011	Project Management
2012	Functional Analysis
2013	Information Analysis
2020	Enhancements Design
2030	Implementation
2031	Initial Conversion
2032	Build and Test
2033	Full Conversion
2034	User Training
2040	Documentation
2041	User Documentation
2042	Maintenance Documentation
2050	Project Accounting and Reporting

ATF-NET	SOW 4.20

3110	Project Management
3120	Hardware, Software Acquisition
3130	Maintenance
3140	Network Installation
3150	On-Site User Support
3210	Training - Standard Products
3220	Training - MIS System
3310	Analysis of Network Requirements
3320	Network Implementation Consulting
3410	ML-NET Tech Transfer
3420	Other Tech Transfer
3510	ATF-MIS - Review Prior Studies
3520	ATF-MIS - Requirements Specification
3530	ATF-MIS - Data Modeling
3540	ATF-MIS - Design/Prototype
3550	ATF-MIS - Productivity Analysis (CBAM)
3710	Engineering Support Analysis - OSI
3720	Engineering Support Analysis - CALS
3730	Engineering Support Analysis - Applications
3740	Engineering Support Analysis - Enhancements
3750	CSRD Support
3810	J & A Support
3910	Documentation Support

NAVY	SOW 4.5.5
4110	Project Management - NADOC
4410	Tech Transfer - NADOC
5110	Project Management - OFIS
5410	Tech Transfer - OFIS
6110	Project Management - Cherry Point
6410	Tech Transfer - Cherry Point

SECTION 2

IISS SUBSYSTEM OVERVIEW

IISS consists of four major subsystems:

- Common Data Model (CDM)
- User Interface/Virtual Terminal Interface (UI/VTI)
- Network Transaction Manager (NTM)
- Communications (COMM)

The UI is hosted on the VAX and IBM. The UI enables the user to access information anywhere in the system and addresses the problems of ease of use and the user's recognition of data. The UI is connected to IISS through the NTM. The NTM subsystem, which is located on all IISS computers, controls and manages activities between the subsystems and IISS applications. The COMM subsystem, also located on all IISS computers, allows communications between computers and addresses the problems of data timeliness and passing of data.

The CDM provides the capability to merge the enterprise's database resources to form an integrated, common source of information that supports decision-making.

The CDM runtime module, shown in Figure 2-1, is the module responsible for accessing the data stored on any database in the IISS environment. It is located on all IISS nodes. This section provides a detailed explanation of these subsystems.
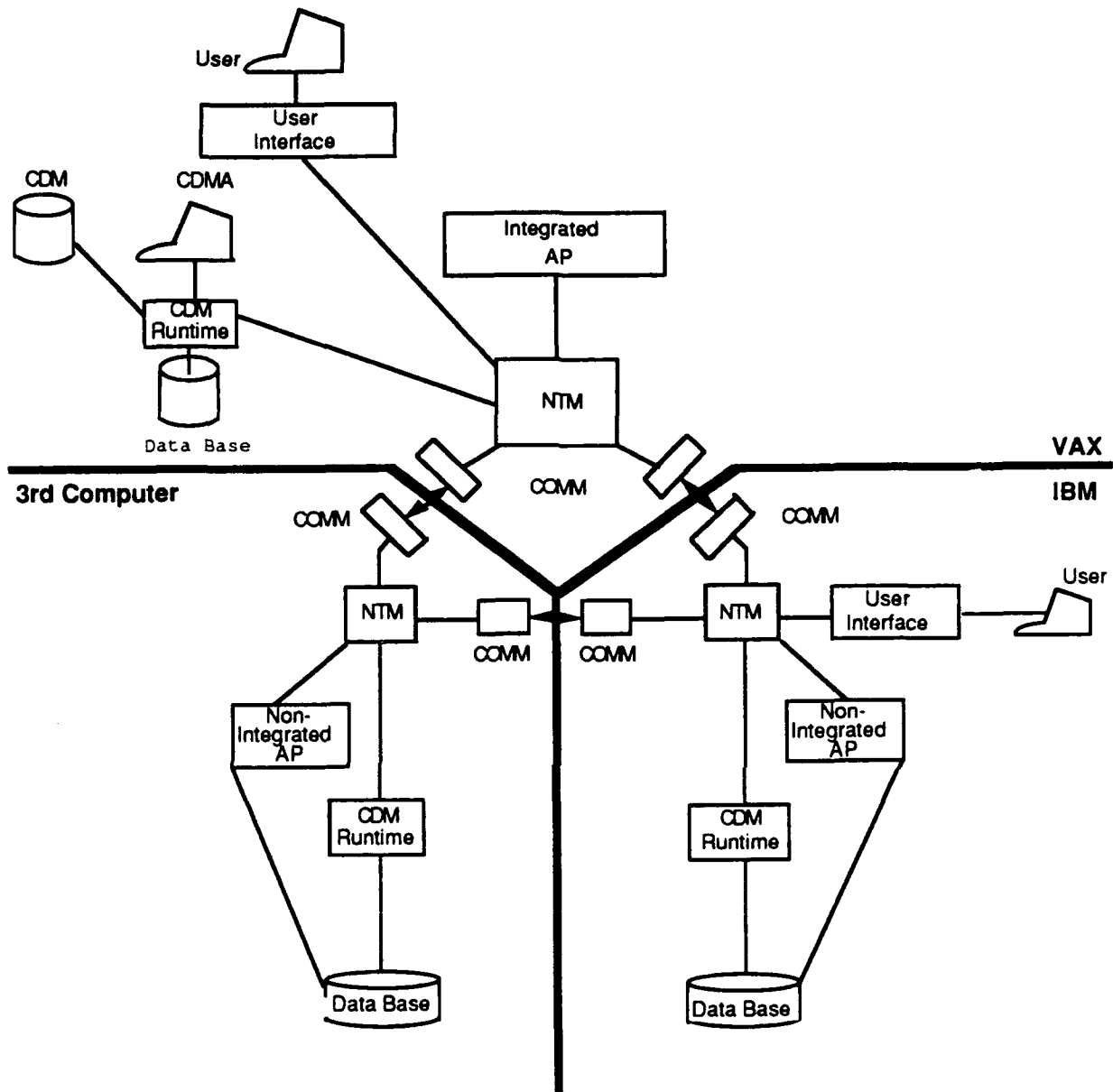
Figure 2-1. Runtime Relationship of IISS Subsystems

2.1. Common Data Model

The CDM makes the information on different databases and computers appear as if it were stored in a central data repository. To accomplish this, the CDM provides a data dictionary/directory that stores information about all of the data existing within the IISS system.

The CDM contains information about the location, format, and definition of data. The CDM uses the ANSI/SPARC three-schema architecture, consisting of the external schema (user view), conceptual schema (data model), and internal schema (physical view). See Figure 2-2.

The information contained in the CDM is defined by the Neutral Data Definition Language (NDDL). NDDL defines the three schemas and inter-schema mappings. Users access data defined to the CDM using a SQL-like Neutral Data Manipulation Language (NDML).

2.1.1 Three Schema Architecture

A key to implementing effective data-oriented environments lies in a framework that is called the Three-Schema Architecture. This approach was proposed in the mid-1970s then developed, and finally published in 1977 in a report from a committee of the American National Standards Institute - "The ANSI/X3/SPARC DBMS Framework: Report of the Study Group on Data Base Management Systems." The basic concepts proposed in the report have the power to lead us to more effective information resource management. They are implemented in the CDM.

The Three-Schema Architecture is based upon several fundamental facts:

• Computers and users need to be able to view the same data in different ways

• Different users need to be able to view the same data in different ways

• It is (more or less) frequently desirable for users and computers to change the ways they view data

• It is undesirable for the computer to dictate or constrain the ways that users view data.

Thus, it is necessary to be able to support different types of views of a data resource. Users need to be able to work with logical representations of data, which are independent of any physical considerations of how the data are actually stored and managed on computer facilities. Users view data in terms of high-level entities; e.g., staff members, tools, vehicles, products, orders, and customers. Meanwhile, computer facilities; e.g., access methods, operating systems, and DBMSs, need to be able to work with more physical representations. They view data in terms of records and files, with index structures, B-trees, linked lists, pointers, addresses, pages, and so forth. These requirements lead us to conclude first that there are two fundamentally different types of data views: logical and physical. The logical views are user-oriented, while the physical views are computer-oriented. A second conclusion is that there must be a mapping or transformation between the logical and physical views. After all, the ultimate objective is to enable user. This mapping might be simple if there were only one user view and one database, but that is not the real-world situation. Rather, there are multitudes of user

views and commonly many (sometimes hundreds or thousands) databases in an enterprise. Each user view could be mapped directly to the underlying databases.

This solution suffers, however, when change is introduced in either type of view. If a physical database is restructured on a disk to provide more efficient performance, then the mapping to each of the user views that references that database can be affected. If a logical view is revised to present information in a somewhat different way, then the mapping to each of the referenced databases may be affected. Independence of logical and physical considerations would not have been achieved, and we would find that physical computer factors would constrain the ways that users logically view their data. This is undesirable. Using three-schema architecture terminology, "external schemas" represent user views of data, while "internal schemas" represent physical implementations of databases. Schemas are metadata, i.e., they are data about data. As a simple example, CUSTOMER-NAME and CHARACTER (17) are metadata describing the data value CHRISTOPHER ROBIN.

To enable multiple users to share a data resource that is implemented on potentially many physical databases, we insert between the users' views and the physical views a neutral, integrated view of the data resource. This view is called a "conceptual schema" in three-schema architecture terminology. Others sometimes call it an "enterprise view." As the vehicle for data integration and sharing, the conceptual schema also carries metadata for enforcement of data integrity. It is extensible, consistent, accessible, shareable, and enable the data resource to evolve as needs change and mature. The following diagram illustrates the relationships between the three types of schemas. The schemas and the mappings between them are the mechanism for achieving both data independence and support of multiple views. An internal schema can be changed to improve efficiency and take advantage of new technical developments without altering the conceptual schema.

EXTERNAL SCHEMA
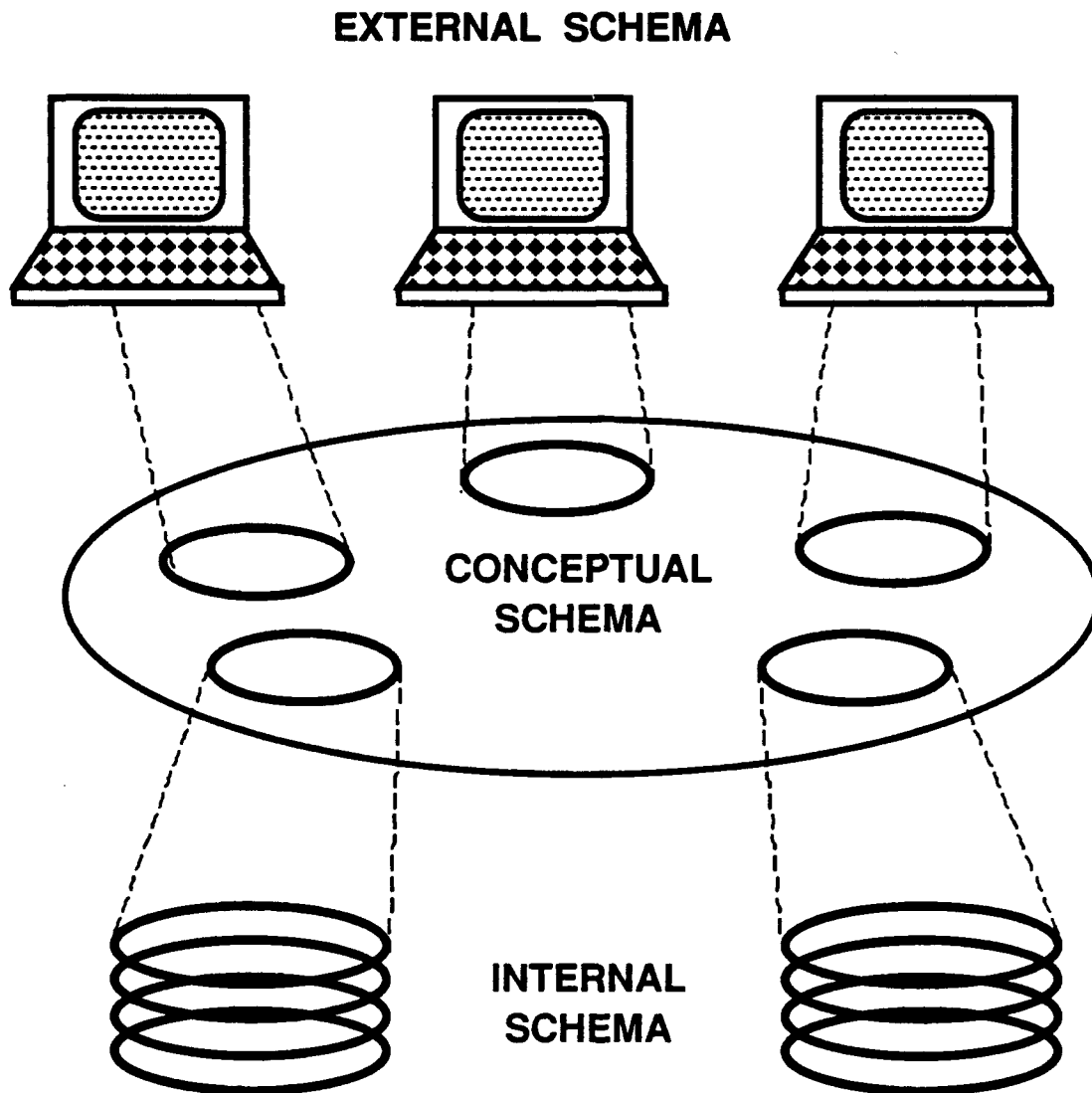
Figure 2-2. Three Schema Architecture

The conceptual schema represents knowledge of shareable data. There may be access controls and security restrictions placed upon these common data, but they are not restricted to access by only one user. The conceptual schema does not describe personal data.

The scope of the conceptual schema expands through time. The conceptual schema extension methodology continually expands the conceptual schema to include knowledge of more shared data. The external-conceptual mappings protect the external schemas and the transactions/programs that depend on them from most modifications incurred in evolving the conceptual schema.

Adding data to the integrated, common resource does not start over in defining the data resource, nor does it create another stand-alone database. Rather, development of its database must examine questions of how those data relate to what is already known by the conceptual schema. The result will be an integrated data resource whose scope is

expanded gradually. It is absolute folly to approach integration of the data resources of an organization all at once; the job must be taken on piecemeal. The conceptual schema is the integrator. The CDM contains all three types of schemas, as well as the interschema mappings. It not only documents these metadata, but also supplies appropriate metadata to support transaction processing.

2.1.1.1 Representation of Three Types of Schemas

In the IISS, the Three-Schema Architecture is implemented through the CDM facilities to store each of the three types of schemas and the interschema mappings. An appropriate representation mode has been selected for each of the three types of schemas. The conceptual schema is represented by an IDEF1X model. The CDM stores this model in terms of entities, attributes, and relations. The external schemas are represented by tables. The user views the common data resource in terms of flat, simple tables. The mappings between these tables and the IDEF1X model of the conceptual schema are part of the CDM database.

The internal schemas are represented in terms of physical database components, including record types and inter-record relationships. The CDM Processor routines convert the users' data access requests, which are phrased in terms of tables, into requests against the conceptual schema IDEF1X model, then into requests against the physical database structures described in the internal schema part of the CDM.

The CDM design has developed from a simple representation of a CDM nucleus and extremities to a more detailed decomposition of the functions to be performed in the system, resulting in a less clear mapping of functions as nucleus and extremity. Another design decision that has evolved is the question of runtime versus precompiler logic for implementing the NDMLs. The approach that has been selected is to develop a Precompiler to decompose the NDMLs. This is as opposed to running "interpretive," or compiling the NDML statements at runtime. The primary reason for a precompiler approach is for performance. The approach is compatible with an eventual runtime compilation that would be used in an ad-hoc query environment.

The Common Data Model (CDM) can be viewed in a way that groups the software modules into the following three major components:

1.   CDM Database and CDM Maintenance, which is the utilities and access to the CDM

2.   CDM Application Development, which is the CDM and application definition/development

3.   Distributed Database Manager, which is the runtime modules

The local data base schemas are input into the Schema Integration Methodology. The results of this, along with the data necessary to load the system tables, become the input to the CDM maintenance routines. These routines load the information into the CDM. This comprises the maintenance cycle of the CDM.

2.1.2 The CDM Database

The CDM database is the database dictionary of the IISS. It captures knowledge of the locations, characteristics and interrelationships of all shared data in the system. The CDM database is implemented as a relational database. The most significant feature of the CDM database is that it implements the ANSI/X3/SPARC concepts of the three-schema approach to data management. These three types of schemas are the conceptual schema, the internal schemas, and the external schemas.

The information in the CDM Tables is populated and maintained using the Neutral Data Definition Language (NDDL). NDDL is an interpretive language that serves two basic purposes. It is a modeling support tool that enforces IDEF1X rules and it is a dictionary definition and maintenance tool.

The definition of the conceptual, internal and external schema objects along with their inter-schema mappings enable schema transformations to be performed. The precompiler generates software modules and code directly into the user's application to do these schema transformations.

The precompile cycle begins with the language source including NDML statements. These NDMLs reference an external schema in the CDM. This is input into the precompiler. The precompiler retrieves CDM schema information, transform information, data locations, etc. The NDML is transformed to the conceptual schema or enterprise view. It decomposes the request into many requests against local data bases according to the internal schemas which go together to make up the conceptual schema and formulates the staging and aggregation rules and conceptual/external transform rules to be used at runtime. It builds the commands for the request process generator from generic DMLs to enable it to generate DBMS-specific instructions. It eventually outputs language source along with the above mentioned rules.

This language source is then put through the system compiler. The instructions for staging, aggregation, and query process generation are sent to their appropriate hosts, where code is compiled to accomplish the specific tasks. At execution time, the precompiled application process generates messages which are sent directly to the proper hosts, which trigger execution of the appropriate query processor. The output from the request processor is sent to the aggregators and stagers interactively until the conceptual response is built. This response is sent to the originating host, where the conceptual/external transformer reformats it into the application-specific format. The data are then returned to the application.
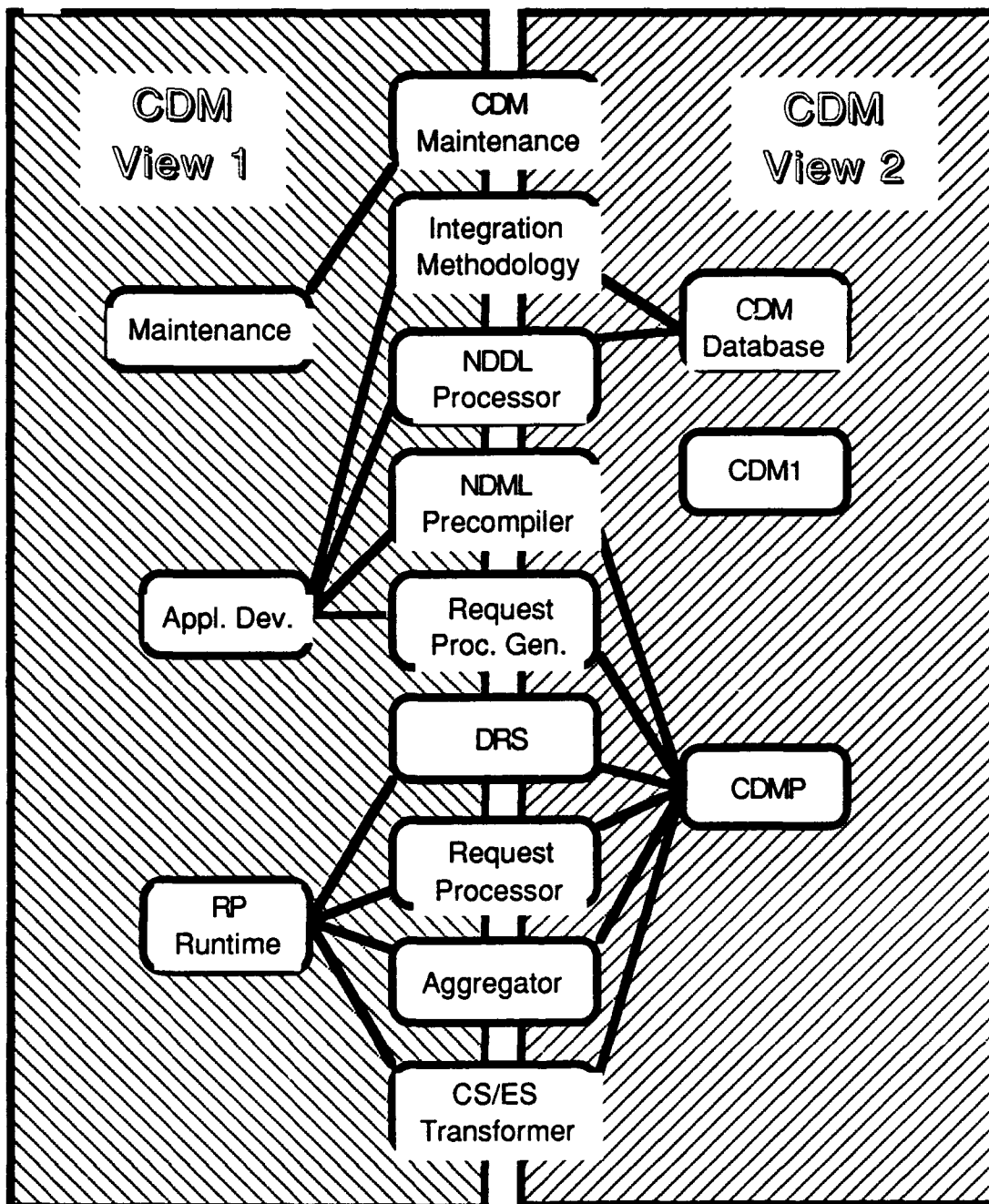
Figure 2-3. Two Views of the CDM Structure

2.1.2.1 Overview of the IDEF1X Methodology

The methodology for building the initial IISS conceptual schema from an existing data base is the foundation for the complete schema integration methodology for a 3-schema architecture. The IDEF1X methodology includes the following activities to build the first version of the conceptual schema:

1.   Identify and define entities.

2.   Identify and define relationships between entities.

3.   Identify and define keys of entities

     a.   Refine non-specific relationships

     b.   Define key attributes

     c.   Migrate primary keys

     d.   Validate relationships and keys

4.   Identify and define attributes

     a.   Develop an attribute pool

     b.   Establish attribute ownership

     c.   Define non-key attributes

     d.   Validate and refine the data model
        The available CDM reports are:

- ENTREP -   The Entity Report displays all entities, their tags and corresponding datatype names, types, sizes, and number of decimals as contained in the CDM. This is useful in examining the Conceptual Schema (CS).

- DFREP -   The Data Field Report displays all records/tables, their columns/fields, and corresponding datatype names, types, sizes, and number of decimals as contained in the CDM. This is useful in examining the Internal Schema (IS).

- CSISMAP -   The CS/IS Mapping Report displays all entities in the Integrated-Model, their tags, and the corresponding database column it is mapped to. The report lists what database and record each column resides in, and also the preference number of the mapping. This report is useful in examining all mappings from the CS to IS that are tag to datafield mappings.

- VIEWREP -   The View Report displays all views, their dataitems, and corresponding datatype names, types, sizes, and number of decimals as contained in the CDM. This report is useful in examining the External Schema (ES).

- DOMREP -   The Domain Report lists all domains, their datatype names, and corresponding, types, sizes, number of decimals, and whether the datatype is

standard or user defined as contained in the CDM. This report is useful in examining all domains.

- ENTDSC - The Entity Description Report provides essential information about an entity. It forms the basis for an entity glossary which furnishes information about entities to the Common Data Model Administrator (CDMA), data modelers, application programmers, database administrators (DBA), and end users. It allows the user to either access entity descriptions to aid in data modeling or assigned keywords for cross reference purposes. Report parameterization is used to limit entities included to a single entity, the entities belonging to a specific model, or a group of entities identified by a keyword.

- ATTDSC - The Attribute Description Report provides essential information about an attribute. It forms the basis of an attribute glossary which furnishes information about attributes to the CDMA, data modelers, applications programmers, and end users. It allows users to access attribute aliases and keywords for cross reference purposes, to determine information about which entities use the attribute, or to ensure that attribute information is correct. Report parameterization is used to limit attributes included in the report to a single attribute, attributes belonging to a single attribute, attributes belonging to a specific module, or a group of attributes identified by a keyword.

- TAGUSE - The Tag Use Index R port lists the attribute name(s) identified by the tag name for the CDMA, data modelers, application programmers, and end users. It forms an index for the attributes. Report parameterization is used to limit tags included in the report to tags of a specific model, a group of tags of a specific model identified be a keyword, or a group of tags identified by a keyword.

- CIMAPC - The Conceptual Schema to Internal Schema Mapping Report shows the CDM Administrator (CDMA) how a database management system (DBMS) is referenced by the CDM from the conceptual schema. This report allows the CDMA to review these critical mappings for correctness and to maintain them. Only objects from the Integrated Model can be mapped. The Integrated Model is the only model within the CDM that is always maintained in a normalized and integrated form. Report parameterization is used to limit the output of the report to a single entity, a single record, a group of entities identified by a keyword, or a group of records identified by a keyword.

- RELDSC - The Relation Description Report allows data modelers and the CDMA to create a list of all relations for a particular entity within a particular model where the entity is either dependent or independent. It cardinality of both entities involved and provides various descriptions associated with each relationship. If the user enters a model name without specifying an entity, the report generates descriptions of all relations involving every entity contained within the model. Report parameterization is used to limit entities included in the report to a single entity, entities belonging to a specific model, or a group of entities identified by a keyword.

- CVMAPC - The Conceptual Schema to View Usage Report provides information about entities and attributes used in a particular view. The report also provides information about views in which an entity participates. This report can be used by the CDMA, data analysts, application programmers, or end users. Report parameterization is used to limit entities included in the report to a single entity, a group of entities identified by a keyword, a single view, or a group of views identified by a keyword.

- MDRPT -   The Data Model Associated Entities Report displays all of the entities associated with a given CDM-defined data model. Report parameterization is used to limit the entities included in the report to entities belonging to a specific model, the entities of a specific model identified by a keyword, or a group of entities identified by a keyword.

- TBLSMC -   The Internal Table Summary Report shows the CDMA and the DBA how the Conceptual Schema is implemented in the DBMS. It displays information about columns that make up each table and the database containing the table. Report parameterization is used to limit tables included in the report to a single table, tables belonging to a specific database, or a group of tables identified by a keyword.

- ENTSMC -   The Entity Class Summary Report provides the CDMA, data analysts, and end users with a listing of all attributes are located within a model and which attributes describe a particular entity. The report displays data type and length of each attribute and whether the attribute is used as part of a key of an entity. The report also shows which attributes were migrated into a particular entity from other entities. Report parameterization is used to limit entities included in the report to a single entity, the entities belonging to a specific model, or a group of entities identified by a keyword.

2.1.3 Neutral Data Definition Language

2.1.3.1 NDDL Functional Summary

The Neutral Data Definition Language (NDDL) is a language used to maintain information in the Common Data Model (CDM) of the IISS System database. It provides the user with three modes of operation:

1.   Batch Mode allows NDDL command files to be executed;

2.   Interactive Mode allows the user to enter NDDL commands at a terminal; and

3.   Forms Mode allows the user use of the IISS forms processor to display input an NDDL commands.

The NDDL language can be divided into five basic groups of commands which perform the definition and maintenance functions for the Conceptual Schema, Internal Schema, External Schema, Schema Mappings, Model Integration. The command groups are as follows:

Conceptual Schema Command Group

The following NDDL commands are used to define and maintain the CS:

Create
Alter
Drop
Describe
Copy

The following verbs supply information about the following CDM objects:

Domains
Models
Entities
Attributes
Relations
Keys

Internal Schema Command Group

The following NDDL commands are used to define and maintain the IS:

Define
Alter
Drop
Describe
Copy

The following verbs supply information about the following CDM objects:

Hosts
Database Managers
Databases
Records
Fields
Sets

External Schema Command Group

The following NDDL commands are used to define and maintain the ES, and to automatically define and maintain the mappings between the external schema and the conceptual schema.

Create
Drop
Describe
Copy

The "Views" is the CDM object about which these verbs supply information.

Schema Mappings Command Group

The NDDL commands used to define and maintain the schema mappings are:

Create Map
Alter Map
Copy Map
Define Algorithm
Drop Algorithm

The Map commands apply to conceptual schema to internal schema mappings only, and are used to define and maintain mappings between:

Entities and Records
Attributes and Datafields
Relation Classes and Sets

The Algorithm commands supply information about user-written transformation algorithms. These algorithms can be written to handle complex mappings between: Data Items and Attributes, Attributes and Datafields or Records.

Model Integration Command Group

There is a set NDDL commands and optional phrases, often to assist modelers with the comparison, validation, and integration of development models with the common Integrated (enterprise) Model. These commands are intended to be used in conjunction with the CDM report and impact analysis utilities. The commands are:

Compare Model
Check Model
Copy/Combine Entity
Alter Attribute
alter ownership
Alter Entity
alter key
Copy/Merge Model

2.1.4 Utilities

Access Control of all CDM maintenance functions is strict and enforced by passwords and user names. Data Access is then provided for CDM Data Entry, Data Update, and Data Edit functions.

2.1.4.1 DDL to NDDL Translator

The NDDL Translator is a utility for translating the native Data Definition Language (DDL) for DATABASE 2 and TOTAL data base management systems into the IISS Neutral Data Definition Language (NDDL) for the creation of the Internal Schema (IS) specification for the Common Data Model (CDM).

2.1.4.2 CDM Impact Analysis

The CDM Impact Analysis Utility identifies and reports which software modules are affected by a change to the CDM and also identifies and reports affected external schemas used by these software modules. Changes to the CDM may require:

- Modification to the application programs to work with the new CDM model

- Reprecompilation of Neutral Data Manipulation Language (NDML) software modules.

- Revision of the NDDL commands causing the CDM changes.

Whenever changes are to be made to the CDM, a CDM Impact Analysis should be run to generate reports giving information necessary as to what additional action must be taken.

2.1.4.3 Common Data Model Compare

The CDM Compare Utility is used to report the differences between two versions of a CDM. Comparisons will be made on the objects within the internal, conceptual and external schemas as well as the objects in the conceptual-internal, conceptual- external schema mappings and complex mappings.

The CDM Compare utility queries the database tables of the CDM and presents its report to a terminal, a file, or a hardcopy device. Neutral Data Manipulation Language (NDML) is used to obtain the required information from the CDM during the extract phase. The user must have access privileges to the IISS environments containing the CDM versions to be compared.

2.1.4.4 CDM Reports

The CDM Reports is a utility that is useful in examining the contents of the Common Data Model (CDM). These utilities and the Neutral Data Definition Language (NDDL) "COPY" commands give a comprehensive picture of the CDM contents to the Common Data Model Administrator (CDMA) and to any other users of NDDL.

The output format desired by the utility user will determine which utility (reports, NDDL "COPY" commands) should be executed. The CDM Reports provide hard copy that list the entire CDM contents for CDM objects in table form. The NDDL "COPY" commands provide the user the CDM contents in legal NDDL syntax format.

2.1.5 Neutral Data Manipulation Language

The Neutral Data Manipulation Language provides the capability of posing requests to the heterogeneous distributed data bases of the IISS as if they were a single data base. The NDML is intended for use by both data processing professionals and manufacturing personnel who have little or no knowledge of computers or programming.

The NDML includes verbs that support retrieval and update of IISS data bases. The four fundamental commands are:

1.   SELECT, which allows the user to specify inquiry of field values from one or more relations, with optional clauses to restrict the retrieval to only those rows that meet specified qualifications.

2.   INSERT, which allows the user to add new rows to a relation.

3.   DELETE, which allows the user to remove old rows from a relation.

4.   MODIFY, which allows the user to modify the column values of an existing row in a relation.

The NDML is designed for embedding in a host language program; e.g., in a COBOL or Fortran program. Embedded usage is suitable for repeated production requests.

A user employing the NDML as a stand-alone language receives rows of data values as complete tables. By contrast, a programmer using the NDML in a COBOL or Fortran program receives rows of data values one at a time. Both forms of the NDML are set-oriented, but the constraints posed by COBOL and Fortran compilers force the embedded form of the NDML to present data to a program in a record-oriented manner.

2.1.5.1 NDML Roots

The NDML has its roots in the two major languages of the relational data base world: SQL (formerly Sequel, from IBM's System R) and Quel (from University of California, Berkeley's Ingres). Nearly all relational, non-procedural, set-oriented languages are patterned after SQL and/or Quel. Examples of the most prevalent commercially available relational DBMSs are SQL, used by Oracle (Relational Software, Inc.) and SQL/DS (IBM) and Quel, used by Ingres (Relational Technology, Inc.).

Both the SQL extensive theory to support their utility. They provide complete data manipulation capabilities. Additionally, they have "closure," i.e., all operators act upon relations, and all operators produce relations as their results. Closure facilitates language implementation and proofs of language correctness and completeness.

2.1.6 Distributed Database Manager

The Distributed Request Supervisor is the mechanism that determines the order of aggregating intermediate results of accesses to distributed databases, interactions among the AP/RP/DRS/Aggregators during updates.

There is a Distributed Request Supervisor program for each site or host in the IISS network. An instance of the Distributed Request Supervisor (DRS) program runs as a subroutine to each user AP that contains NDML commands. This DRS takes on the role of master control program for all the transactions from that user AP. Instances of Distributed Request Supervisor at other sites become the master control programs for transactions initiated at those sites.

The Distributed Request Supervisor has responsibility for run-time scheduling of activities comprising distributed database accesses and updates. It initiates local Request Processors and receives replies when they have completed. It sends subtransactions to appropriate application clusters, initiates file transfer requests to transmit intermediate relations, and initiates corresponding Aggregators to perform join, union, and outer join operations on intermediate relations.

2.1.6.1 Aggregator

The overall objective of the Aggregator is to perform relation join, union, outer join and not-in-set operations upon intermediate results of a multi-database transaction. It, along with the DRS, Request Processors, and local DBMS modules, performs the run-time evaluation of commands presented to them by application processes.

The Aggregator sorts the files containing the operands, creates the file which will contain the resultant table, reads the operand files, performs the join, union, outer join or not-in-set operation and stores the results in the resultant relation file. Finally, it deletes the operand files, sends a completion message to the DRS or user application program which created it, and terminates.

2.1.5.2 File Utilities

There are two file utilities. These are invoked by the Distributed Request Supervisor. The first utility is File Transfer. This is used to move a result file from one machine to another so that the file may be aggregated with another file. The second utility is File Delete. This is used to delete results files as they are aggregated.

2.2 User Interface

2.2.1 User Interface Structure

The subsystem which makes each user's terminal able to access information anywhere in the system and addresses the problems of ease of use and how the user will recognize the data is called the User Interface (UI). The software to provide this capability consists of the following components:

- User Interface Development System (UIDS)

- User Interface Management System (UIMS)

- Electronic Documentation System (EDS)
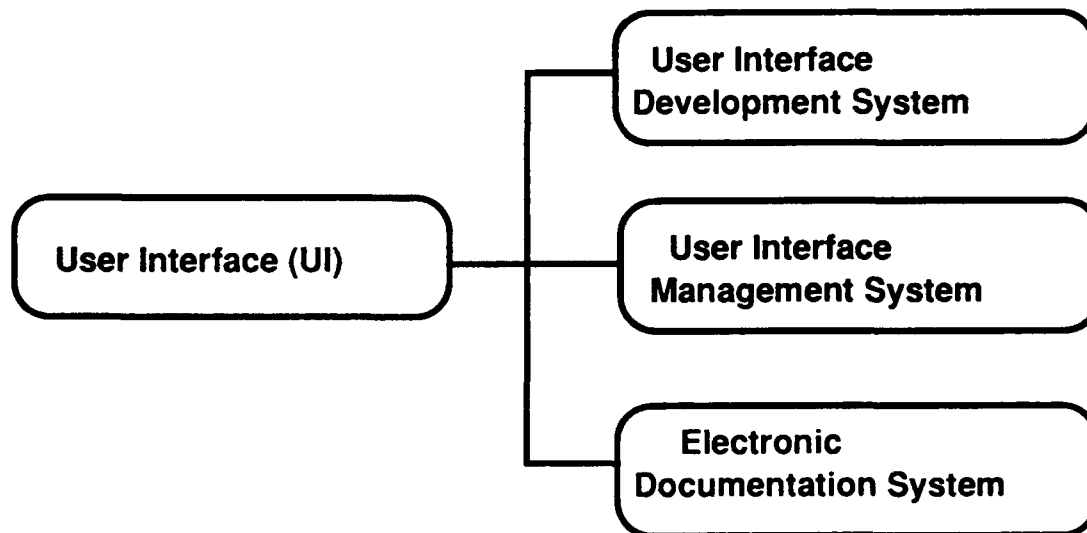
Figure 2-4. User Interface Modules

2-16

2.2.2 UI in the IISS Environment

It is convenient to consider the UI in terms of interactive and run-time environments. The function of the UI is threefold. First, the UI provides an environment that not only allows but encourages good user interface design. Secondly, the UI provides a run-time environment that supports interactive dialog. These two UI subsystems are called the User Interface Development System (UIDS) and the User Interface Management System (UIMS). The third function is the Electronic Documentation System (EDS)

The User Interface Development System contains three development components. The Forms Editor (FE), the Text Editor (TE), and the Application Generation which has two parts, the Report Writer (RW) and the Rapid Application Generator (RAP). The Report Writer (RW) and Rapid Application Generator (RAP) can be viewed as IISS Applications.

Form Processor (FP), the Virtual Terminal Interface (VTI), the Application Interface (AI), as well as several IISS applications associated with the UI such as the User Interface Services (UIS) comprise the User Interface Management System. Device Drivers in the UIMS are the only processes that can be started directly by user. All other processes are initiated by the UI through calls to the Network Transaction Manager (NTM). The Monitor is the portion of the FP that manages this process and the communication between the AP's and the Form Processor. The FP also provides text-editing capabilities through the Text Editor which are available to any application that runs in the IISS environment, without any special code being written.

The User Interface Services are applications that aid programmers in defining and controlling this environment.
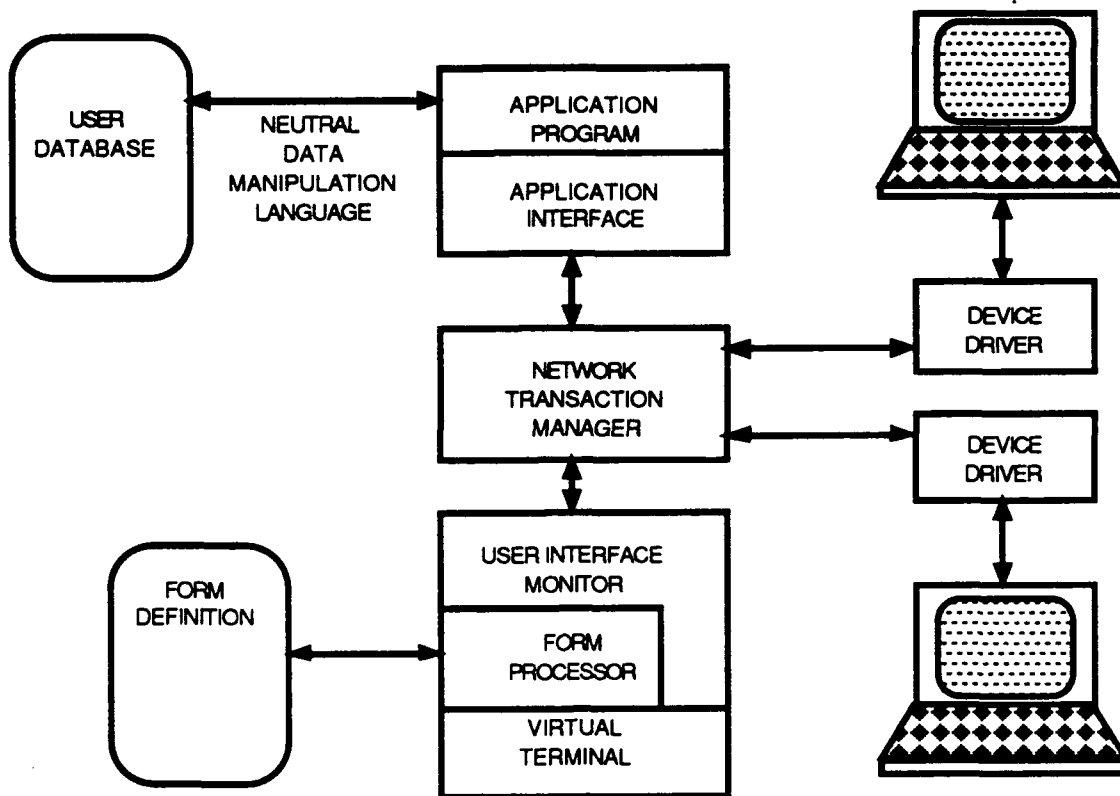
Figure 2-5. User Interface Environment

The Application Interface is a library that contains routines similar to the FP and
VT routines. The difference is that the Applications Interface Communicates through the
NTM to the UI Monitor to perform the actual calls to the Forms Processor or Virtual
Terminal rather than performing the actual function directly. This allows the UI to be
hosted on a different machine than the AP. Since the Application Interface looks exactly
like the FP or VT routines to the AP, the AP can run "stand alone" or in the IISS
environment without changing code. The AP must simply be linked with the appropriate
library.

2.3 Network Transaction Manager

The Network Transaction Manager (NTM) provides the operating environment the
system issues such as the decomposition of the system, performance, portability, and
transaction processing definitions are intimately related to the NTM design. The top level
functional responsibility of the NTM is to manage messages, manage processes, and
maintain operability.

2.3.1 NTM Structure

The NTM is divided into four major architectural components:

- Monitor - primarily involved with the maintain operability function. It controls startup/shutdown of the IISS, monitors operation, and provides an interface to the IISS Operator.

- Message Processing Unit (MPU) the main module for managing messages and processes.

- NTM AP Services the set of service routines that is bound to the APs. It provides a subroutine call interface for all NTM services accessible to the AP. The NTM Service Routines then dialog with the MPU.

- File I/O Primitives the set of routines to provide a neutral input/output access.

```
                                    ┌─────────────────────────┐
                                    │        Monitor          │
                                    └─────────────────────────┘

                                    ┌─────────────────────────┐
                                    │ Message Processing Unit  │
  ┌───────────────────────┐        └─────────────────────────┘
  │ Network Transaction   │
  │   Manager (NTM)       │        ┌─────────────────────────┐
  └───────────────────────┘        │      NTM Services       │
                                    └─────────────────────────┘

                                    ┌─────────────────────────┐
                                    │    File I/O Primatives   │
                                    └─────────────────────────┘
```
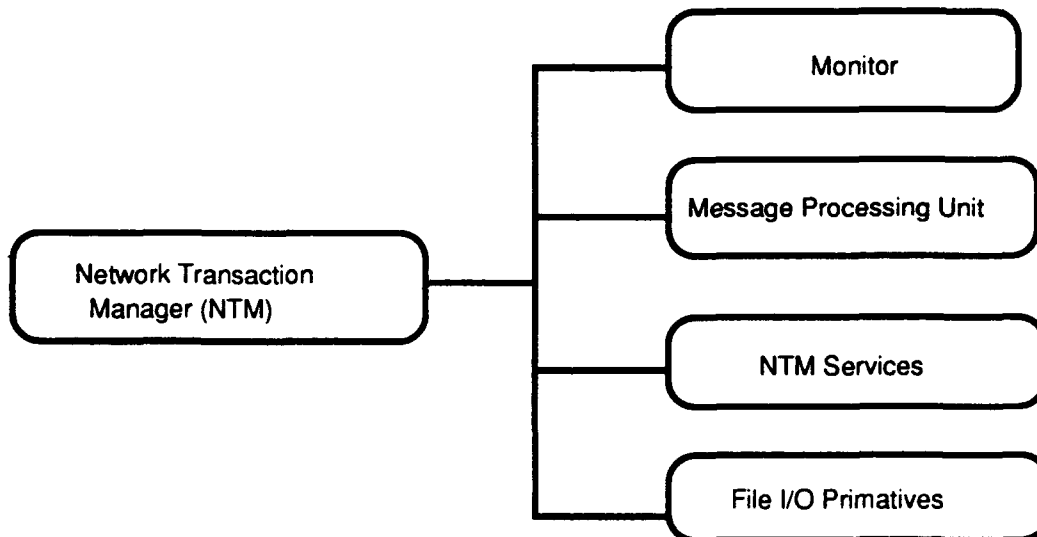
Figure 2-6. IISS Network Transaction Manager Modules

The Monitor component of the NTM is comprised of two modules. The first module is the operator interface to the IISS system. This module accepts and processes operator commands and delivers status. The second module is the monitors the IISS status.

The IISS architecture is based on the concept of cooperating Application Process Clusters (APC). Each APC is a collection of Application processes (AP's) that are uniquely addressable but form a subsystem or application from the user's prospective. Each APC has a NTM component called the Message Processing Unit (MPU) that supports the application Process clusters AP's by providing process management and message services. Certain APCs support IISS components including Communication, Common Data Model Processor and User Interface. These system Components, in combination with the NTM, cooperate to provide transaction processing, communication, and data integration services to users at IISS terminals and to application programs.

The NTM APC components form a distributed executive for the IISS that provides the coordination, communication, and housekeeping functions required to integrate the application processes across heterogeneous computers and databases into the IISS. The MPU supports message communications between APC clusters and between APs and the MPU through "mailboxes". Each MPU has two input mailboxes: one for high-priority messages and the other for low-priority messages. The APs associated with the MPU's cluster and the MPUs associated with other clusters send their messages to MPU's mailboxes.
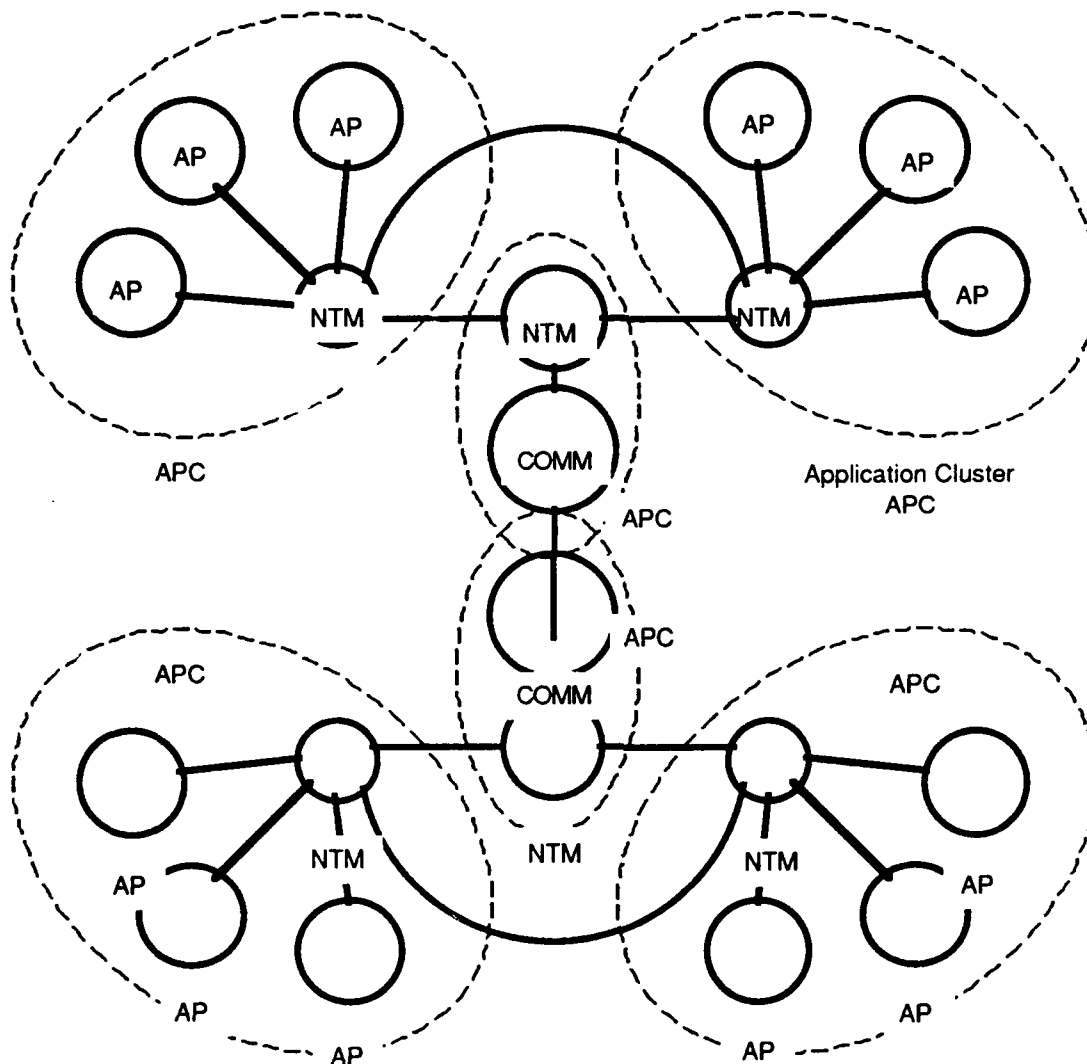
Figure 2-7. IISS NTM Architecture

The APs have an input mailbox configuration of from zero to three mailboxes depending upon the AP's message processing requirements. If an AP does no message "sends" or "receives" during its IISS processing, the AP will have no input mailboxes. If the AP sends messages to other IISS APs but will receive no messages, then it requires only an acknowledgment (ACK) mailbox. The ACK mailbox is a small special-purpose mailbox used by the MPU to send MPU acknowledgments to an AP for the messages that

an AP has sent. If the AP will only receive messages from other IISS APs and does not require synchronous high priority status messages from the NTM, then the AP will have a Low-Priority Mailbox and ACK Mailbox. The final case is where the where the AP can receive messages from both IISS APs and the NTM and requires all three mailboxes.
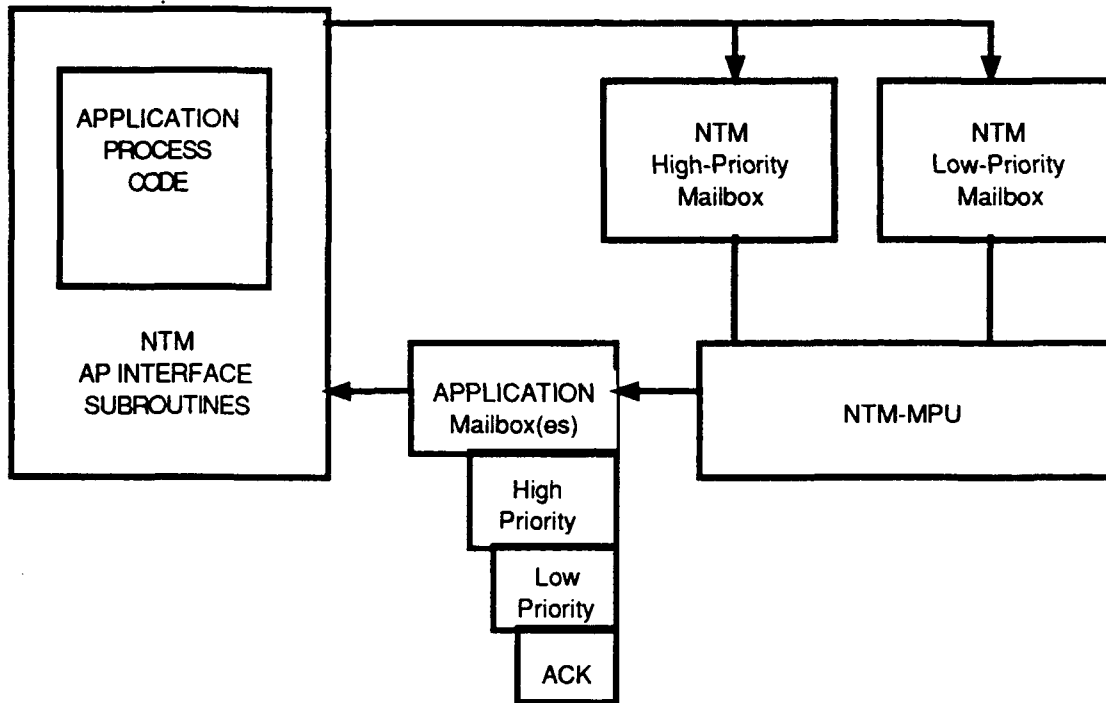
Figure 2-8. APs and the NTM

2.4 Communications

The function of the Communications Subsystem (COMM) is to transfer accurately and in an orderly way messages between two computers or between two tasks running on one computer. It is expandable to enable the addition of other computers to IISS at a later time.

2.4.1 Communication Requirements

The software constraints on the Communications Subsystem are:

1.   Modifications to the operating systems of the computers is not allowed.

2.   Writing or modification of I/O drivers is not allowed.

3.   IISS on the IBM runs under CICS.

4.   The system-dependent code must be contained in a small set of routines.

        The hardware constraints are:

1.   A broadband Local Area Network must be used to carry the message traffic between the computers.

2.   The interface between the computer and the LAN must use standard, vendor-supplied hardware.

It was decided in 1982, that the state of the art of LAN was such that its hardware and software would be undergoing rapid modifications and improvements for the next several years. Also, committees, such as the International Standards Organization (ISO) and the National Bureau of Standards (NBS), were beginning to design protocols for communications that would become standards. Thus, the decision was made to choose a hardware interface and a software protocol that could be replaced without affecting the remainder of IISS.

The emergence of General Motors' MAP, built upon the ISO Open Systems Interconnection (OSI) model, and the impact it is having on vendors of computer and communication hardware and software supports the correctness of this decision.

The hardware chosen was the standard, asynchronous, RS232C terminal interface. The protocol, implemented in software, was a combination of Bisync and layers one, two and four of the ISO/OSI standards. Layer three, network, was not needed because the connection between any two computers was to be point-to-point using permanent virtual circuits. The protocol was designed and implemented in-house.

2.4.2 Communication Structure

The subsystem which allows communications between computers and addresses the problems of data timeliness and passing of data, is called Communications (COMM). COMM contains both hardware and software. The hardware consists of a local network which connects several computers. There is a copy of the COMM software for each link on each computer. Each copy is responsible for communications with one of the other computers. The majority of the software is the same on all computers. A small amount of software called the Interhost Communication Primitives (IHC) are responsible for interfacing to each computer. These are machine dependent and must change for each different computer implementation. The transfer of messages between tasks on the same computer is accomplished through Interprocess Communication Primitives (IPC).
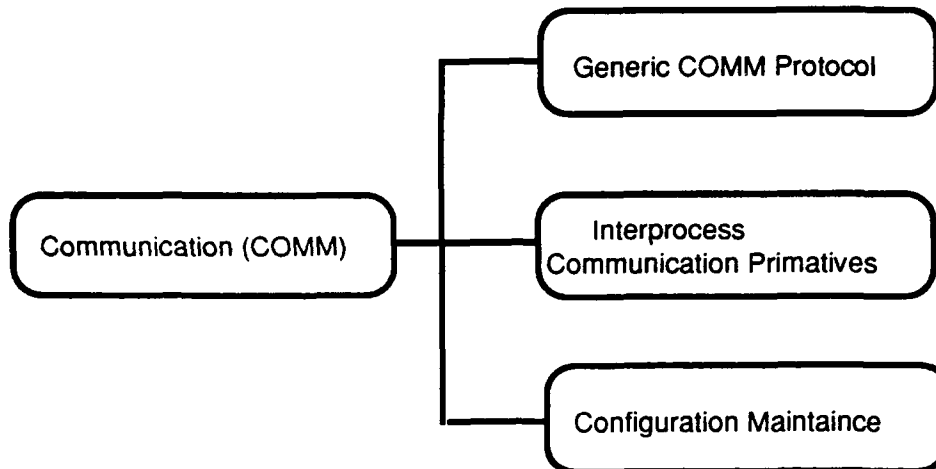
Figure 2-9. IISS Communication Modules

COMM communicates with the NTM via the IPCs. Messages are sent in each direction between the two subsystems. Each message contains message data and a message header. The COMM uses the following fields in the header:

1. Destination AP Name

2. Message Type

3. Binary/Native Flag

4. Priority Flag

Messages between NTM and COMM are control messages or data messages.

2.4.3 Protocol Characteristics

The system-dependent code that interfaces to the terminal I/O driver in each operating system is restricted to a set of routines called the Interhost Communication Primitives (IHC).

The Communications Subsystem design contains the following communications protocol characteristics.

1. It is a contention system with one of the computers designated as primary to resolve collisions.

2. The master/slave relationship is determined by whichever side successfully bids for the line at the start of a session.

3. Once a session is established, the message transmissions are interleaved, half-duplex.

4. Error detection is performed through a Longitudinal Redundancy Check. Correction is done by retransmission of the message.

5. Communication blocks are variable in length.

6. Large messages are segmented and then reassembled on the receiving side.

7. There is no prioritization of messages. They are transmitted on a first in, first out basis.

8. Only the ASCII character set, less the control characters, is used in message transmission. Control characters are transmitted through byte stuffing and character transformation.

9. Binary data is transmitted by transforming it to the ASCII equivalent of its hex value, then converting it back upon reception.

COMM contains functionality specified for ISO/OSI layers 1, and some of 4. The NTM contains functionality for layers 3, 4, and 7.

2.4.4 Interprocess Communications

The set of routines to support communications between two tasks on one computer exist. These are called the Interprocess Communication Primitives (IPC). They consisted of the primitive processes:

1. Create a mailbox

2. Send a message to another program

3. Post a receive for a message from another program

4. Get a message from another program

5. Delete a mailbox

6. Start a timer

7. Cancel a timer

8. Wait for a event to occur

9. Terminate a program

10. Release an event block

2.5 The Relationship of the Subsystems

The CDM is used to define the data in the IISS environment. Programs use the CDM and its associated tools to access common data definitions. These common definitions in turn access the actual data stored on various databases and computers.

The IISS user requests access to data using the VTI. The VTI allows a variety of terminal hardware types to access the IISS system. The UI provides the capability of designing screens in a fast, user-friendly manner. It also provides the interface to the NTM, which coordinates these requests. The NTM passes the message to the COMM modules for communications between host computers and other nodes. The data is accessed on the individual databases and returned to the user via COMM and NTM.

2.6 Interfaces

This section describes the system-level interfaces between the principal IISS software subsystems. The major software components in the IISS environment are:

1. Integrated Application Programs

2. Non-Integrated Application Programs

3. Common Data Model

4.   Distributed Database System Processes

5.   Local Database Management System

6.   Network Transaction Manager

7.   User Interface

8.   Communication Subsystem

9.   IISS Monitor

The following diagram shows the interfaces between the components. There are several layers or levels of interfaces and there are protocols between the components.
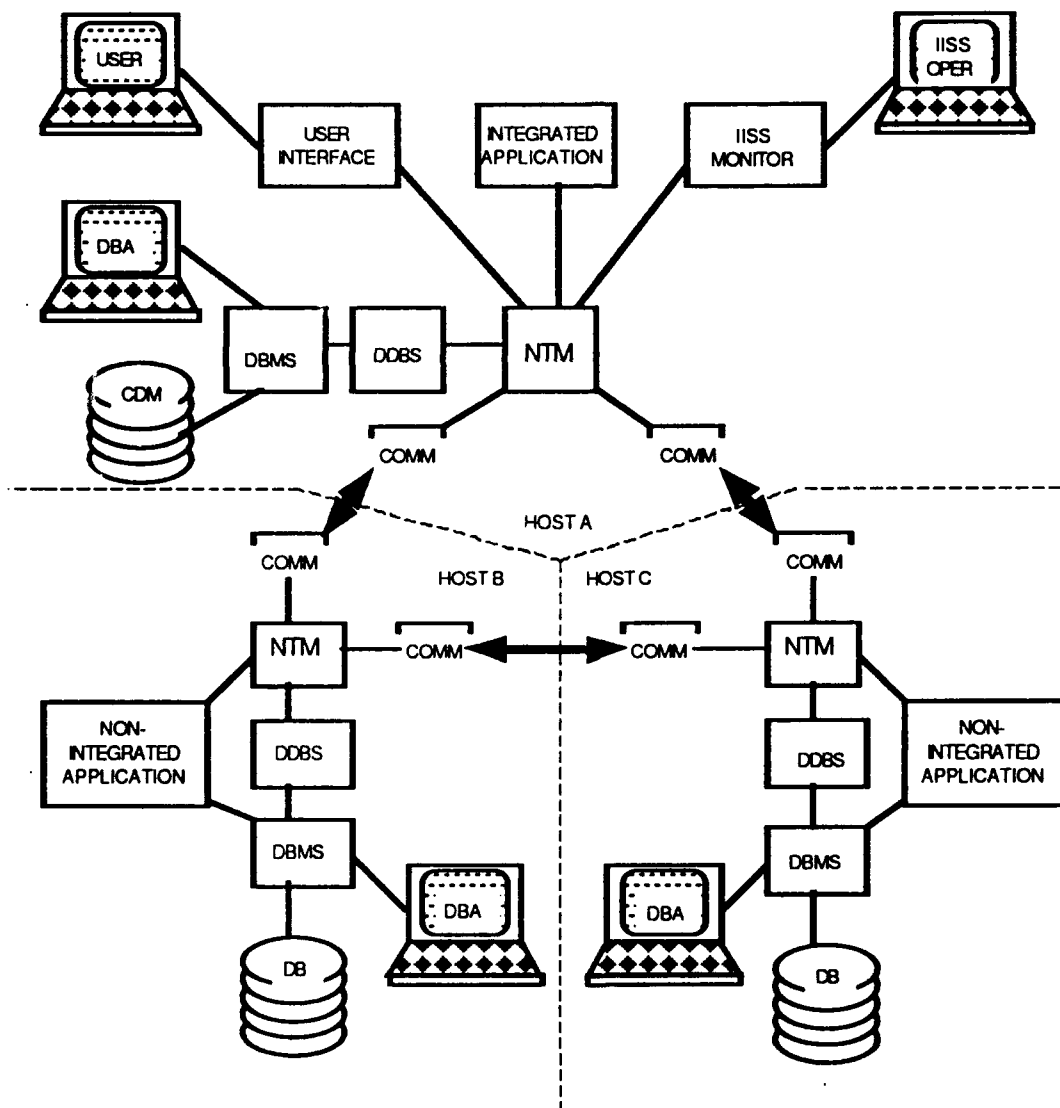
Figure 2-10. IISS System Overview

2.6.1 Services Provided

The services which are provided by a software component are typically a subset of the functions of the component. For example, the NTM is called to send messages between application processes. This is a service. The NTM must also validate the message header information and route messages to their destinations, but these are not considered services in this context.

2.6.2 Protocols and Messages

The services listed above are implemented in a distributed processing environment like the IISS by exchanging "messages" between programs. When two programs of processes cooperate to deliver services, they must establish a set of rules and agreed-upon messages which, together, are called a "protocol". The message distribution services supported by the NTM, IPC, and COMM subsystems provide a basis for implementing these protocols.

The complexity of IISS process interconnections causes confusion about the distinction between interfaces and protocols. The following diagram shows two views of process interaction. On the left is shown a "macro-level" view of a protocol between two application processes. This protocol is implemented by interfaces with the NTM. A similar protocol exists between the NTM processes and, at the bottom level, a telecommunications protocol implements the actual transfer of the data. On the right-hand side of the diagram is shown a "microscopic" view of the interface between the application and NTM processes at the left. This view depicts a protocol between that application process and the NTM that is implemented by interfaces with the IPC.
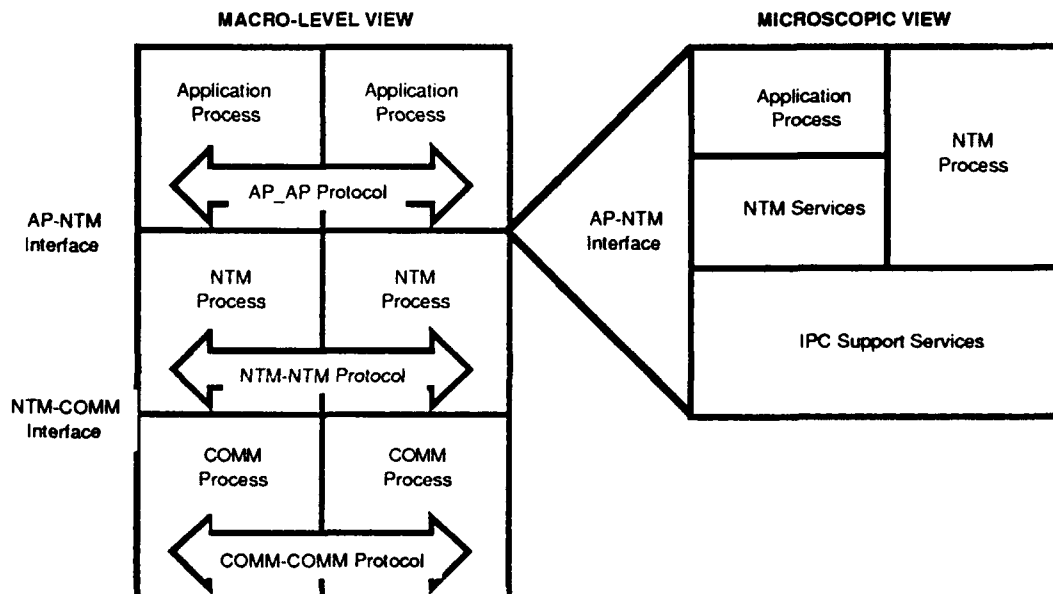
Figure 2-11. Protocols Between IISS Processes

SECTION 3

DAPRO PROJECT ACCOMPLISHMENTS

3.1 Overview

IISS contains the following four subsystems:

- Common Data Model Processor (CDM)
- User Interface/Virtual Terminal Interface (UI/VTI)
- Network Transaction Manager (NTM)
- Communications (COMM)

This section describes the contributions made to these subsystems and to IISS as a whole during this reporting period.

3.1.1 System Integration Components

Computers integrated within IISS are connected for communication processes by the COMM software resident on each host computer. The NTM on each system provides the connection between the subsystems and application processes. The UI and VTI provide a standard interface for all terminals and application programs.

The CDM subsystem provides the enterprise with an integrated view of the information. This view allows a user to identify and access the data without knowing where or how the data is stored. The CDM runtime programs perform the actual retrieval or update of the data on the physical databases.

The IISS software system architecture exemplifies system integration. IISS subsystems must be defined to the integrated environment. Integration testing of subsystems is the primary responsibility of the developers of each subsystem.

All subsystems must use services provided by the NTM to communicate with themselves and with the other subsystems. Application programs must use services provided by the UI and CDM runtime to retrieve and display information.

The next level of system integration testing and operation combines all functions of heterogeneous computer systems so that they work together as a single integrated information support system. This involves coordinating all NTMs and COMMs.

The following paragraphs explain the concepts of each subsystem and highlight the integration steps and enhancements that have been accomplished during this reporting period.

3.1.2 CDM General Requirements

In an effort to define the bounds of the CDM, industry standards were researched and adopted for IISS. The standards project declared interim standards, procedures, and guidelines for the Integrated Information Support System (IISS). This project was driven by the need to bound the IISS functions and, thus, its design. Three major areas: conceptual design decisions for IISS; standards; and procedures and guidelines were specified. All three areas are needed to bound the IISS design and the applications that will function in the IISS environment.

Conceptual design decisions are expressed in terms of rules that set the scope of the IISS software. These rules are summarized as follows:

1. No Central Data Base Site.

    The software is to be designed such that there is no dependence on a single master copy of the data base population.

2. IISS is a Transaction Processing Environment.

    The fundamental processing scheme is called "transaction processing", which may be described by the following characteristics:

    a. Transactions initiate the execution of application programs. These applications programs differ sufficiently from batch to interactive application programs to have their own name - Transaction Processing Routines (TPRs).

    b. Transactions are messages with a general format specifying message type, transaction type, and parameters.

    c. Transactions are addressed to queues. The presence of one or more transactions in a queue causes a TPR to be initiated.

    d. Parameters are pass to a TRP through a commonly addressable communications area or through the programming language parameter-receiving mechanism.

    e. TPRs process one transaction and then terminate.

    f. TPRs are initiated by a Network Transaction Manager as a result of finding a transaction in a queue.

    g. All transactions are statically pre-defined.

3. No Foreign Access Is Allowed

    Only application programs following the rules of IISS and under the control of IISS should access the data bases controlled by IISS. However, other accesses are not prevented.

4. Only Local Node Update Guaranteed

    Update of data at a node other than the node executing the TPR will not be guaranteed.

5. Minimum Common Data Model Control Will Be Maintained

These items represent the minimum control to be exercised by the Common Data Model (CDM)

a. Application programs used to update or query IISS bases will be written in an IISS standard programming language, use a standard Data Manipulation Language (DML), and address a schema written in a standard Data Description Language.

b. The subschema (external schema, or view) addressed by an application program is a derivative of the conceptual schema in the CDM.

6. No User Interaction When the Data Base Is Unstable.

User interaction with a TPR is not allowed while the data base is in an unstable state. An unstable state exists if one or more data base records are locked or if failure to complete the user interaction would result in a data base inconsistency. This is enforced by the native DBMS.

7. The Three-Schema Approach Will Be Used.

The ANSI/X3/SPARC DBMS Framework Report of the Study Group on Data Base Management Systems is the model for discussing, specifying, and designing the relation of application programs, subschemas, conceptual schemas, and internal schemas.

The standards, procedures, and guidelines were declared after conducting a survey of twenty-one subject matter experts. These experts were solicited for recommendations on published standards that would apply to the IISS. This survey yielded the conclusion that few published standards exist, although there are a number of significant efforts underway. As a result, the standards project changed its focus from the selection of existing standards to the declaration of specifications and working practice as interim standards.

Although few applicable were found, critical issues regarding the IISS are thoroughly addressed. These issues are bounded to the extent that failure to adhere to the rules and standards as declared will result in an IISS which is significantly different from that which is intended. Key decisions focus on the issues of data management and language definition. Standards are declared for the Data Manipulation Language, the Common Data Model, and the Data Definition Language. A limited selection has been made of Data Base Management Systems which will be compatible with the characteristics defined for IISS.

Confirmation of the decisions made on these critical issues was provided by a NASA survey of existing DBMS standards. NASA's findings coincided with those of the IISS expert survey in that few published standards were identified. The NASA survey further identified the nature and extent of work in process. ANSI is following the same direction as the IISS interim standards in basing much of its work on the CODASYL Recommendations. This ensures an easy transition from the declared standards to approved standards when they are published.

3.2 CDM Accomplishments

3.2.1 CDM IDEF1X

Active participation in the IDEF User's Group has continued within this reporting period.

In January 1990, the first IDEF framework working group session was held in Dayton, Ohio and Version 1.0 of framework document was edited and produced.

In February 1990, the IDEF framework was presented to the IDEF Steering Committee in Miami at their IDEF User's Group Meeting. Control Data participated in the workshop evaluation of the framework.

During March 1990, the IDEF framework was reviewed by the Dayton Expert Review. The IDEF framework was sent to Universal Technology Corporation for distribution.

3.2.2 PDES Models

The Product Data Exchange using STEP (PDES) program is concerned with developing an integrated definition of data to enable enterprises to share and exchange product data.

Recognizing the need to organize integration and exchange of this information has resulted in major effort within the PDES community to define a central repository to store the myraid forms of product data models. The IRDS standard is proposed as an integration and configuration management mechanism for PDES. A long-term goal for a central repository is to create a unification schema to store any kind of knowledge.

The PDES models used in the CDM-PDES demonstration at the OSD-CALS office, described on the following pages, were all taken from the Tokyo draft of the Integrated Product Information Model (IPIM), the same draft proposed standard circulated for international comment. Although all were documented in EXPRESS, it was quite straightforward to translate the semantics to IDEF1X and then to the NDDL of the CDM. EXPRESS does allow for a number of constructs such as rules and constraints that cannot be defined in IDEF1X, but must be enforced with procedural code. On the other hand, EXPRESS does not explicitly contain any notion of unique keys or entity identifiers (keeping with more traditional object oriented design). To overcome this, a unique integer identifier was invented for each entity. At first, before translating the product structure configuration model, a part number and revision number must be added to each entity. This was not a faithful translation of the semantics. For this effort, the following models were translated:

MODEL	ENTITIES	ATTRIBUTES	RELATIONS
Features	174	56	384
Topology	40	7	67
Miscellaneous Resources	33	24	45
Geometry	<u>76</u>	<u>65</u>	<u>149</u>
TOTAL	323	152	645

All of this meta data was coded in NDDL and loaded in the CDM. Only that data needed for demonstration was populated at the external and internal schema levels. This translation effort represents four of the 28 models (or schemas) integrated in the Tokyo draft of the IPIM. Note that in the table above, IDEF1X entities were counted. IDEF1X and EXPRESS entities do not translate one for one; in general more IDEF1X entities are required. As a comparison, the four models translated 221 EXPRESS entities into 323 IDEF1X entities. In the entire IPIM, 831 EXPRESS entities were counted. Therefore, 27% of the PDES entities were translated.

Two of the models listed above, Miscellaneous Resources and Geometry, along with a portion of Product Structure Configuration Management were used in the CALS EXPO '89 demonstrations.

In May and June 1989, prototype Level III PDES access across distributed databases was demonstrated to the AF Program Management and the Office of Secretary of Defense (OSD), Dr. M. McGrath. This demonstration was a precursor to the CALS EXPO '89 demonstration in Orlando, and provided the Air Force and OSD with the opportunity to review the technical accomplishments required to successfully support the Orlando event in December, as well as provide feedback for the larger demonstration.

The CDM-PDES demonstration utilized a PDES Level-III database scenario interfaced with the SDRC I-DEAS CAD package. In preparation for this demonstration, SDRC built a Constructive Solid Geometry (CSG) representation of a lens cover part to be used in December at the CALS EXPO Demonstration in Orlando. The CSG data was converted to a PDES exchange file format which was in turn converted to a SQL representation and loaded into a SQL database built for that purpose. The PDES entities necessary for the CSG were translated into Neutral Data Definition Language (NDDL) and loaded into the Common Data Model (CDM). The NDDL for internal and external schemas, as well as requisite mappings was developed and loaded, and an application was written against the CDM which extracted the data and organized it in a format acceptable to I-DEAS. The data was then transferred to I-DEAS where it was displayed and modified.

The objectives of the CDM-PDES demonstration were to explore the applicability of the IISS technologies to support the PDES standard, to learn as much as possible about PDES Level III and the data sharing requirements of CAD applications, and to provide a baseline for the CALS EXPO demonstration in December. All of these objectives were

realized. In addition, follow-up work in July included a rewrite of the algorithm used in the OSD demonstration, moving from a record-oriented method of processing to a relational style of set-oriented processing. The new approach reduced the demonstration performance time from 20 minutes to 45 seconds. Work since then suggests that a true production Level-III PDES capability could be built relatively cost-effectively, but that the construct should be object-oriented rather than relational to achieve production-level response times.

3.2.3 PDES to SQL Translator

During preparation for the 1989 CALS Expo in Orlando, Control Data developed a generalized PDES exchange format to SQL INSERT translator. The translator parses valid PDES exchange file syntax and checks data type conformance (integer, float, string array, etc.). This translator is driven by data files describing the PDES entities and attributes that are expected, and the SQL table definitions and the mappings between them. The translator was written in C and tested under the VMS operating system.

The input meta-data files were manually derived from the PDES Integrated Product Information Model (IPIM) and the SQL table definitions of the data base. The EXPRESS tool developed by McDonnal Douglas Aircraft in St. Louis was used originally to generate the SQL table definition. However, it generated too many single attribute tables. The "create table" statements were therefore modified to combine many of the table definitions.

The translator creates a unique entity ID for each row it generates. This number is determined by searching the database for a special table containing the last used entity ID. The program will generate the update statement to change this. The program was not designed for a multi-user environment. The database used is ORACLE, but the code is not highly ORACLE dependent.

The translator can populate supertype entity occurrences if need be and handles lists, arrays and sets properly, generating a unique row for each list or array element.

The output is a file of SQL INSERT's which can be piped to the database manager's interactive interface. The INSERT commands spell out each column name to remain independent of unmapped columns and immune to column ordering in the table definition.

The major drawback at this time is that the translator cannot tell what exchange file data is already on the database and what data actually represents new occurrences. It assumes all of the data is new, which won't be the case for some the entities such as PERSON, ORGANIZATION, DATA, PRODUCT ITEM etc. It also does no data verification, such as checking the references and applying any of the EXPRESS defined rules.

It will not be technically difficult to flag entity definitions in the meta-data file that must be examined for pre-existence. However, this would require definition of their unique symbolic key, which for many IPIM entities is undefined. With the key information, the translator could be tied to the database directly to look for pre-existing data and if found, compare its values to those on the exchange file. In that case, the translator should generate UPDATE statements. If the row was not found, it would continue to generate the INSERT statement.

The translator could be modified to actually execute the generated statements. Other passes could be added to verify the entity-to-entity references found in the exchange file.

Another enhancement would be to add meta-data to the translator's input to dictate which host and which database the entity occurrences are to be stored, providing a rudimentary distributed database capability.

It would also be worthwhile to pursue integration of the translator's meta-data inputs with that found in the CDM. In pursuit of this, a program translator from a EXPRESS to CDM's Neutral Data Definition Language was designed and coded by McConnel Douglas Aircraft. This translator could be adapted to generate the NDDL's internal schema table definition commands, the external schema view definitions and the conceptual to internal mappings as well.

There is some question concerning whether relational technology will be capable of serious implementations of PDES Level III and IV, especially for complicated, interrelated areas of geometry, solid models and Finite Element Models. It does seem possible, however, that relational database will be used for many other, less complex types of data defined in PDES and that relational could be used to store the entire geometric model in a "blob" data type.

3.3 <u>User Interface/Virtual Terminal Interface Access Accomplishments</u>

The architecture of the IISS User Interface (UI) enables it provide the same user interface style across many physically different terminal devices. Essentially, the UI terminal is a neutral terminal. It represents the mechanism by which data is passed from the UI Form Processor (FP), to the actual physical terminal or other output device. The UI terminal is defined as a set of functions that is can perform, the set of attributes that it can support, the set of commands for invoking functions, and a mode of operation. The UI terminal insulates the applications from terminal dependencies and makes all terminals for which there are UI device drivers look operationally similar to both the application developer and the end user.

Terminal specific device drivers translate between Virtual Terminal (VT) commands, which are commands to the UI neutral terminal, and commands for the specific terminal types. Since no single terminal provides all of the functions and attributes of the UI terminal, the Device Driver (DD) simulates missing functions with existing ones where possible (it is not, for example, possible to provide "blinking" characters on a Macintosh personal computer).

As part of the development of the MT/ML Travel System (paragraph x.x), device drivers for the Macintosh and IBM-PC personal computers were developed. The objective in developing these device drivers was to enable end users of the MT/ML Travel System to access the IISS based travel application from their IBM-PC or Macintosh personal computers in a transparent manner.

The Macintosh and IBM PC device drivers were the first device drivers written for "intelligent" terminal devices. This required additional functionality to support user logon from the personal computer to the host computer on which the IISS application resided. Additional logic to distinguish between "host" command sequences and virtual terminal command sequences was also required by these device drivers. The following is a list of some of the features of the IISS Macintosh and IBM-PC device drivers.

Macintosh

- Support for logon in a separate Macintosh window to a host computer
- Support for all Macinstosh window functions (except grow) for the UI window
- Support for the Macintosh mouse - single click for positioning, double click for select
- Support for Macintosh text editing functions for UI input fields (cut, paste)
- Support for configurable and stored terminal settings
- Support for 2-D graphics
- Support for screen dump of UI screen to a local Macintosh printer

IBM-PC

- Support for logon to a host computer
- Support for 2-D graphics
- Support for configurable and stored terminal settings

As part of the development of the PDSS system (paragraph x.x), a UI device driver for the VT320 terminal was developed.

3.4 NTM

The primary objective of this task was to host the NTM on the UNIX operating system and to make the necessary enhancements and modifications to the NTM to support OSI. In addition, a secondary objective was established to host the NTM on an Intel IPCS2 parallel supercomputer and to make modifications to enable the NTM to take advantage of the parallel architecture of the IPSC2.

In June 1989, work began to port the NTM from COBOL to C language so the NTM could run efficiently in the UNIX environment, would be more portable in the future, and could support the OSI protocol. As part of the porting process, some sections of the NTM code were "rewritten" to take better advantage of C language capabilities (e.g. dynamic memory allocation, linked lists). At the end of this reporting period, the status of the NTM conversion to C is as follows:

- All COBOL code converted with the exception of:

  1. Q-Server support
  2. HSTATS service

In preparation for CALS Expo 89, the NTM code was moved to the Intel IPSC2 at the CALS staging area where a limited amount of testing was performed by SDRC. UES also ran a number of the test programs that they had converted from COBOL to C against the C version of the NTM. A number of errors were identified and fixed during this testing period. Additionally, SDRC and CDC worked to integrate the C version of the NTM and the CDM at the CALS Expo and a number of errors were identified and corrected during this period.

As part of the NTM port to the Intel IPSC2, the IISS UI was ported to the IPSC2 301 machine. All UI code was successfully ported and the test program ARTEST was executed to verify that the UI was working properly.

Beginning in October 1989, modifications were made to the NTM low level IPC (system dependent) routines to support the parallel architecture of the IPSC2. The modifications included calling IPSC2 system service routines to allocate, manage, and deallocate IPCS2 nodes so that processes that were under control of the NTM could be started and stopped on the IPSC2 nodes. SDRC worked closely with Intel in the development of these routines. In addition to process management, functions to provide messaging between the NTM Message Processing Unit (MPU) running on the Intel SRM (the front end to the IPSC2) and any process running on a IPSC2 node were developed. Other modifications were made to the NTM to enable configuration information regarding the number of nodes to use to be read in by the NTM. Test programs were developed to test the modifications and the NTM was able to successfully create, terminate, and manage processes running on nodes of the Intel IPSC2 machine.

In October 1989, a TCP/IP Communication Module was developed and tested to provide an alternative protocol for use at the CALS EXPO 89. It was felt that the development of this COMM was necessary due to the uncertainty associated with the availability of a vendor supported OSI stack for the Intel machine.

The TCP/IP COMM was successfully tested on the following operating systems:

- UNIX System V.3 (Intel)
- HP-UX
- VAX/VMS (using Wollongongs TCP/IP package)

Two test programs, a message sender (NTMSND) and a message receiver (NTMRECV) were developed to test the execution of the NTM between two computer platforms. Using the TCP/IP COMM, these programs were executed and the NTM was successfully able to send and receive messages between two UNIX machines (HP and Intel) and between a UNIX (Intel) and a VAX/VMS machine.

At the end of this reporting period, the COMM OSI was not complete due to the unavailability of an OSI stack for the Intel IPSC2 machine.

In August 1989, a DECnet to DECnet Communications Module (COMM) was developed to support the PDSS demonstration. This module supported the transfer of results data generated by the CDC run-time processor between two geographically separate VAX machines.

3.5 PLMM/CMIC

The AFSC/PLMM (now CMIC) office maintains a database that contains information concerning Critical Materials that are used in the manufacturing process of airframes, missiles and the requisite aircraft engines. This database has existed for 11 years on the ASD main computer system within the SYSTEM 2000 data base management system.

SYSTEM 2000 technology became obsolete in the late 1980's. The AFSC/PLMM office was unable to maintain the application. SYSTEM 2000 has not been supported by a commercial vendor for several years and no local support talent was available to upgrade the application. Therefore, AFSC/PLMM desired to upgrade to a modern database management technology.

A study of the Critical Materials database structure was conducted by Control Data. It was decided that if the application were to be converted to operate within the IISS environment under ORACLE, two purposes would be served. The AFSC/PLMM office would have their application brought up to current technology standards, and the DAPRO project would have an application that demonstrated the capabilities of IISS·

The application was successfully converted and improvements were made over the previous SYSTEM 2000 version. The system now operates in an interactive mode instead of the old batch mode. Other improvements were made in the areas of report generation and backup capabilities.

The difficulties encountered in the area of User Interface development caused the project to run past the scheduled completion date.

3.6 ATF

Since 1989 Control Data has been supporting the Advanced Tactical Fighter (ATF) Program by providing modeling and networking expertise at the ATF System Program Office (SPO).

On March 31, 1989 a network was completed that allows the ATF SPO Commander InBox™ electronic mail access to his 27 division directors and their clerical staff. Control Data started with no desktop systems. A full operating network existed 3 weeks after arrival. This effort involved the selection of protocol, laying cable, testing and integrating the cable plant, and unpacking, assembling, testing, and software installation of 75 Macintosh systems.

The network was expanded by June 1989 to include all the Control Data supplied unclassified Macintosh systems in the SPO and installed a File Server for SPO users to use to share data and applications. Additional users were added to the mail system over the next several months as required.

Using the expanded network, analysis and studies were performed and completed in February 1990 to integrate Zenith Z-248 and EVEREX 286 IBM compatible systems into the backbone network. This network has formed the basis of all future connectivity with the SPO and the external world. The design implemented provides maximum flexibility and adaptability for future network expansion, permitting the ATF SPO to interface with any other network. The next step concerns inclusion of the GOSIP and OSI protocols necessary for the successful implementation of the CALS initiatives in the Full Scale Development and Production Phase of the ATF. At this time, InBox access and file and applications sharing access has been provided to all users. A network users manual was provided and delivered and training was provided for the SPO users.

Periodically, Control Data has provided training to the SPO users in the standard SPO software: Microsoft Word 4.0, Microsoft EXCEL, Microsoft Powerpoint, and MacDraw II. These training courses have been developed and perfected in order to deliver with one days notice.

In September 1989, maintenance services were provided for the hardware delivered in March of that year through a sub-contract arrangement with Falcon Microsystems.

In November 1989, a Functional Node Tree was completed for the ATF SPO. The node tree displays all of the SPO activities and how they are interrelated.

Studies and analysis resulting in a recommendation for the ATF SPO to purchase a VAX system to include in the network were completed in December 1989. This purchase would provide the necessary platform for extended file sharing and the technology transfer of the Man Tech IISS software.

In April 1990 software and and support was provided to the ASD Computer Center (ASD/SC) to link the ATF SPO with other AREA B systems. This required that hardware installation and proper gateway permissions were loaded into the Building 676 systems. The ATF SPO now has more efficient access to the AMS systems in the Computer Center, while also having access to the Defense Digital Network (DDN). As a result, the SPO user can communicate with the other bases around the CONUS to exchange files and electronic mail.

In April 1990, the Aperature software was evaluated and demonstrated. Technical support was provided for implementation. Aperature is a floor plan software application that will allow the SPO to keep track of furniture, personal computers, Macintoshs, software and local area networks. It is a valuable configuration management tool, as well as facilities management system.

In April 1990, a Preliminary Report presenting a study was conducted and completed for the YFE, Engineering Directorate. This study included interviewees perceptions, lists of operational activities and aggregates that represent priority needs for the interviewees, conclusions and resulting recommendations.

In May 1990, software was obtained and installed to expand the InBox electronic mail system into a fully integrated mail system. Previously, only InBox generated messages could be seen by the user in InBox. With this gateway addition, all mail, regardless of origin (for example, DDN, AMS) can now be read and answered by a SPO user with InBox.

In June 1990, a study was performed and completed to expand the current network to accommodate SPO Facility moves, both within Building 50 and to Building 50A. This work is continuing to generate an ease of expansion capability and maximum flexibility for the SPO in any future networking and connectivity efforts.

From March 1990 to June 1990, specifications were prepared for the ATF and ATFE request for proposal (RFP) Work Breakdown Structure (WBS) element 6200. These specifications concerned integration of the Airframe, Engine, SPO, SPM, and other government facilities using the CALS initiatives and existing networks. The meetings with four potential contractors consisted of Control Data providing expert consulting in matters of data formats, communications protocols, generating an IDEF0 model of the WBS 6200 element and capturing the essence of the basic integration requirements. At this time, the In-Process Review (IPR) with the contractors is continuing. Significant progress has been made in the RFP WBS elements.

In January 1990, the ATF SPO Information Model was completed. This model expresses the relationship between the management information.

Access Tracking System (ATS) software and user manual were completed June 1990 for organization YFMX, the Security Program Division, at the SPO. This system

enables the personnel at the SPO to produce reports for people with specific program access. It produces rosters that lists people that are allowed access to a specific program.

In July 1990, the software for OP-SEC Information Security System was released to the SPO. This application answers questions regarding the handling of classified material. For example, where to mark "classified" on a document, or how to mail classified material. The maintenance manual for this application is in process at this time.

For the YFMC, Directorate of Acquisition Support at the ATF SPO, an analysis of the information Personnel Management work breakdown structure paragraph 6200 was produced. A preliminary analysis of Control Data Requirements Lists (CDRL's) for the SPO, locating redundant CDRL's and providing standardization of terminology was completed.

3.7 NAVY

3.7.1 Posture Planning Decision Support System (PPDSS)

One of the functions of the Naval Aviation Depot Operations Center (NADOC) is to support the Naval Aviation Depot community in the posture planning of the Naval Aviation Depot Maintenance Program. The Program Manager Depot Maintenance (PMDM) located at Naval Air System Command Code 43, in conjunction with the Naval Aviation Corporate Board and the Posture Planning Committee, is responsible for ensuring that the posture of the Naval Aviation Depot (NADEP) community is consistent with military and economic requirements and priorities. NADOC provides support in several ways, one of which is determining cost alternatives for various posturing issues related to the repair of aircraft and associated systems.

The posture planning process was intended to identify requirements and establish constraints within the Naval Aviation community in order to meet certain fundamental goals. These goals were to; (1) maintain a sufficient Naval Aviation Industrial base, (2) attempt to maintain a consistent workload among the NADEP community, (3) ensure the ability to meet wartime mission requirements, and (4) accomplish these charges in the most cost-effective manner possible.

Prior to the initialization of the development effort of the Posture Planning Decision Support System (PPDSS), the posture planning decision process was a somewhat loosely structured application of posturing guidelines and much manual compilation and analysis of the required data. The PPDSS is a Decision Support System (DSS) that was developed to assist NADOC in its support role for AIR-43 and to specifically aid in posturing analysis for new aircraft and missile systems, engines and systems with A/N designators in the Naval Aviation Inventory. The PPDSS was the result of a careful requirements analysis and various industrial posturing requirements. It has provided consistent, reproducible, documentable alternatives and supporting decision rationale. The following requirements formed the basis for meeting the overall objectives of the system:

a. Mission analysis for the determination of the need for an organic industrial base as a Ready and Controlled (R&C) source that can meet mobilization surge and wartime sustenance requirements.

b. Mission analysis for the determination of Private Sector industrial resources required to meet surge and wartime requirements.

c. Technology analysis for the determination of appropriate manpower skills and repair technology capabilities need to meet long term posturing objectives.

d. Cost analysis to evaluate posturing alternatives with respect to program-assigned costs.

e. Interface with econometric analysis tools to evaluate posturing alternatives with respect to workload impacts on NADEP business base costs.

f. Report generation for output of posturing alternatives, mission requirements, capabilities data, costs, workload impacts and reasons for decisions that lead to the development of these recommended alternatives.

g. Verification of posturing alternatives based on decision analysis consistent with DoD Depot Maintenance Interservice (DMI) Decision Tree Analysis (DTA) instructions.

The PPDSS identifies alternatives consistent with the above requirements and assigns appropriate costs. It should be emphasized that the PPDSS does not make decisions. It provides, in a structured process, information required to make such decisions while guiding the analyst through the analysis process.

The PPDSS is a knowledge based expert system that has been created using an Artificial Intelligence (AI) software package as the framework for system development. The system employs rule sets that have been created, where applicable, to determine the decision process flow and, when external data is required, how that data should be used. The application programs that have been developed serve as an intelligent front-end to the PPDSS and require large amounts of data to satisfy the functional requirements. To provide this data, the Integrated Information Support System (IISS) was integrated into the PPDSS.

The IISS is a form of distributed processing that facilitates the acquisition and manipulation of specific data requirements in a heterogeneous computing environment. It allows efficient access to required specific information elements that might be resident on non-compatible computer systems utilizing different data management systems and are geographically separated. The integration of the IISS into the PPDSS allows the PPDSS to operate more efficiently and provides access to all required information on an as needed basis while obtaining the most up-to-date information. This concept was successfully tested September 14, 1989 at a PPDSS installation located at NADOC, with data access from systems located at NADEP Cherry Point, North Carolina and the ManTech Test Bed located in Dayton, Ohio. While the demonstration used only a small portion of the total PPDSS capability, proof of concept and capability was achieved with remarkable success.

Following the demonstration, the PPDSS was fully developed as a PC based system using the GURU expert system shell. As a result of initial testing of the system, it was determined that a PC based system would not deliver the optimum performance required by NADOC personnel. The PPDSS was subsequently ported to the Micro-Vax 3600 located at NADOC using the VAX-VMS version of GURU. All functional areas of the PPDSS have since been exercised and appear to be running correctly. Additionally, an updated copy of the PC version of the system was delivered to NADOC along with a completed user manual.

The PPDSS is comprised of four major functional areas of analysis:

- Depot Source of Repair (DSOR)
- Reposture (REPO)
- Depot Maintenance Manager Logistics (DMML)
- Competitive Decision Package (CDP)

Each of these areas was specifically designed to support the policy and posturing objectives of the Naval Aviation Industrial community.

3.7.2 DSOR

The DSOR analysis is the assignment of the Depot Level repair capability and associated workload for a new aircraft or System to a specific depot repair facility selected to best satisfy the Navy's Industrial Posturing objectives. The first function of the DSOR, pre-mission analysis, was designed to determine if the new aircraft/system (system), has replacement requirements, due to attrition, that will increase from peacetime levels during a general national mobilization or conventional wartime conflict situation. The new system will require a mobilization analysis in the mission analysis component if it has a frontline deckload type of employment such as fighter/attack aircraft or if the system could be subject to hostile action resulting in increased attrition. Finally, the ability of the new system to offset existing or projected skills shortages in other systems received during mobilization is considered.

The mission analysis component of the DSOR function was designed to determine specific requirements that affect the placement of the workload for the new system. It was also designed to identify supporting data such as the total system inventory and distribution of that inventory, attrition projections based on mobilization scenarios, and the identification of labor skill requirements for the new system. The cost analysis component of the DSOR function was intended to determine the total cost for each alternative generated at completion of the mission analysis. If there is no mission analysis required for a system, the cost analysis will examine all possible sites from a competitive viewpoint. The posture factors that might affect the DSOR function introduces other factors that might affect the DSOR decision process, such as whether the new system represents a new industrial process technology.

3.7.2.1 REPO

The Reposture Section (REPO) performs the comparative analysis and feasibility of moving existing production requirements and associated resources from one location to another. The first functional area of the REPO section is designed to gather information for the reposture analysis and to define what exactly is to be repostured. The reposture analysis requires that a mission analysis be performed to identify any mobilization requirements. The mission analysis performed is a subset of the one performed during DSOR analysis, identifying any Ready and Controlled Source or skill deficiency requirements.

The Program Cost Analysis portion of REPO evaluates the costs to the program (customer/system manager) resulting from relocating the system workload from one site to another. These costs include both Non-Recurring and Recurring and if the gaining activity is known, are only done for the losing and gaining activity.

The Econometric Analysis portion of the REPO section determines the effect on the operating efficiency of both the losing and the gaining activity by relocating the applicable system workload.

3.7.2.2 DMML

The Depot Maintenance Manager Logistics (DMML) Section was created to identify the appropriate assignment of the responsibility for obtaining the resources and achieving the capability necessary for new production requirements. The first item to be examined by the DMML sections is an analysis of the old system workload to be replaced by the new system workload. This portion of the DMML analysis attempts to identify if the new system requires any new repair technology and if so, does any depot currently have plans to acquire that new technology.

The cost section of the DMML analysis is intended to calculate the total cost differential for the new workload for each depot for a projection of five years. The depot with the greatest cost differential will show the most benefit (increased efficiency) by obtaining the new workload.

3.7.2.3 CDP

The Competitive Decision Package (CDP) Section was created to provided the analysis for the "bid/no bid" decision and most advantageous bidder for new or current production requirements being offered for open competition. The first area of the CDP analysis is the reason for the competition.

The non-recurring cost section of the CDP analysis determines the non-recurring costs associated with proposal preparation and capability establishment, if applicable, for the repair workload under competition. The recurring cost portion of the CDP analysis first asks the user to provide the annual repair workload (five year) associated with the weapon system or support system under consideration. This workload in man-hours is then examined by the econometric model and a total cost differential is then calculated for each depot for the specific program indicated.

The final portion of the CDP analysis displays the information determined in the preceding sections. It also allows the user to assign probobilities of actually winning the competition to each depot.

Each major section contains an interactive report generator that allows the analyst to produce specifically tailored reports of the results of the analysis for review by the appropriate decision makers. Finally, the PPDSS contains a Data Management function for manipulation and extraction of the specific data required by the functional analysis area.

The PPDSS system will be primarily used by NADOC personnel to analyze DSOR alternatives and provide recommendations to the decisions makers. Personnel outside of NADOC will have limited access to the system, but will be able to view all supporting data and review the logic process of the system. The PPDSS will contribute greatly to maintaining the continuity of the posture planning process during personnel changes that might occur.

3.7.3 Product Support Directorate (PSD)

3.7.3.1 PSD-V22

During the period from January 1990 to June 1990, multiple IDEF0 information models were generated for the V-22 Product Support Directorate and subsequently validated. They consisted of two primary areas: the Logistics Support Analysis (LSA) Task Requirements Model, and the Depot Maintenance Program activity and Task Analysis Model.

The LSA Task Requirements Model consists of an IDEF0 model of the MIL-STD 1388-1/2 developed specifically in support of the V-22 Airframes Logistics Lead (code 35240). This model will support information engineering efforts for the production phase of the V-22 weapon system life cycle. The purpose of the model will be to ensure that the requirements of MIL-STD-1388-1A are met. Additionally, the model will increase the Logistics Lead's productivity through the use of a systematic application of computer technology.

The Depot Maintenance Program Activity and Task Analysis Model is a revision to the previously submitted (January 1990) draft with NADEP changes incorporated. Of the initial functions analyzed, seven were selected for modeling. These are: Program Management, Weapon System Management, Resource Management, CFA Assignment, Workload Determination, Product Support Execution, and Weapon System Support Requirements Determination. This model will be the basis to develop a Precision Requirements Model for V-22 Airframe Support.

This modeling activity will be the basis for an integrated system that will monitor Integrated Logistic Support (ILS) element level support analysis enabling the Airframes Logistic Lead to have readily available upon demand, an analysis of supportability derived data, and an assessment and verification of supportability. This tool will become, for the Logistics Lead, an effective means of managing changes as the V-22 Weapon System migrates from full scale development into the production phase of the acquisition process.

As different tasks are applied this automated decision support system will be used to edit and "signal" when changes begin to impact on supportability; for example, how will Engineering Change Proposals (ECP's) effect the various ILS elements and what type of retrofit kits are required? A Major concern during the process of weapon system transition from development to production has always been maintenance and operational support during Initial Operation Capability (IOC). The information engineering model will provide audit trails. The audit trail for the LSA process is to be documented and will ensure that all Logistic Support Analysis guidance has been properly applied in accordance with MIL-STD-1388 requirements.

3.8 Technology Transfer Activities and Accomplishments

Fasteners, Actuators, Connectors, Tools and Subsystems (FACTS) Total Quality Management (TQM) models were completed in September 1989. Consulting is beginning for future FACTS projects.

In March 1989, CDC completed the white paper, "Use of NDDL for Managing PDES Models," and sent the paper to the PDES Dictionary committee chairman for review.

From April to July 1989, four phases of the Spring session of IISS Workshop '89 were presented to over 30 personnel representing several government and private organizations. This first session presented the concepts of IDEFO and IDEF1X methodology. Another workshop on IDEF methodology was presented later in April for personnel who missed the first workshop. The second phase presented the concepts and setup of the Common Data Model. The third phase covered all aspects of the User Interface (UI). Students began writing ADL and FDL for their application programs.

In May, the planning began for the Fall IISS Workshop '89 Session, to be held in the August-November, 1989, time frame. Two phases of this session were presented in August and September.

3.8.1 Technical Accomplishments

Technical accomplishments for this reporting period include the following:

- Completing the definition for Geometry Express conceptual schema.

- Demonstrating the MT/ML Travel System.

- Defining a scenario for a CDM PDES demonstration for CALS EXPO '89.

- Completing a definition for the PLMM database to the CDM.

- Completing Macintosh Device Driver coding for better cursor support.

- Completing IBM-PC Device Driver coding, with the exception of support for the Logitech mouse.

- Resolving problems with SDRC communications for the Test Bed.

- Completing coding changes for CDM_IDEF1X Impact Analysis and for the ES-CS transformer of the NDML precompiler.

- Completing coding the printer for both Laser Writer and Fujitsu printers.

- Defining the domain, conceptual schema, internal schema and conceptual to internal schema mappings for the PLMM project in NDDL.

- Selecting and delivering classified removeable hard disk drives for ATF.

- Loading the PDES data into an ORACLE database and describing all three schemas to the data dictionary. Control Data also developed a CDM application to extract data from the data dictionary and to provide input to an engineering workstation for plotting circles.

In June 1989, Control Data, SDRC, and UTC demonstrated a prototype IISS application designed to assist users in handling the paperwork associated with government travel, and to provide government managers with a superior management-tracking capability. The objective was to build a working prototype Travel System which could generate standard forms, be extended to multiple, heterogeneous databases, could enforce business rules and constraints, could provide a consistent user interface with built-in help, and could provide management with a reporting and tracking capability for TDY funding at Branch, Division and Directorate Levels for all account classifications. It also had to be

system-independent, providing a consistent user interface across several types of equipment currently in use in MT, including PC's and VT terminals.

The Travel System was demonstrated in June and was completed in September 1989. It is ready for implementation and has been installed on the testbed computer. Certain enhancements were made to the user interface to provide more flexibility in where the user can direct printed output, and increased ease of use for filling in the forms. The communication capability of the PC drivers was enhanced to help provide stable access to host computers, even over dial-in lines.

3.8.2 CALS Expo 1989

The Computer-Aided Acquisition and Logistics Support (CALS) program comprised objectives to accelerate integration of reliability and maintainability tools into contractor CAD/CAE systems, to encourage automation and integration of technical data throughout the weapon system life cycle, and to improve DoD capabilities to manage and use technical issues (communications, distributed database technology), the legal and policy issues (rights to data, security, competition) and the cultural changes also combine to influence the long term objectives of the Phase II CALS Program.

Control Data was responsible for coordinating the CALS EXPO '89 DoD-Industry Coalition Booth, including development of the EXPO '89 scenario used to portray application of PDES, CDM, and the CALS 1840A standard in a real world situation. This activity was a follow-on to the lead role taken by Control Data in the USAF/Industry Coalition Boothe at CALS EXPO '88.

The situation a demonstrated by showing of the process an engineering design change must go through as the change is applied to a lens cover part used to protect mission-critical components on weapon systems in the Air Force, Navy, and Army, as well as NASA space-born systems. In the demonstration scenario, failure data associated with the lens cover flagged it as a problem within a "lead service" branch. A "lead service" is the particular service branch in the rotation that is presenting the problem, for example, Air Force, Army, or Navy).

In the demonstration, the operations chief related the current 10-month scenario required to perform the change, and then began to wonder "what if?"...He was thinking of a new way to do business.

The operations chief's "vision" shifted the scene to a Total Quality Management (TQM) Meeting held at the Defense Logistics Agency. Throughout the demonstration, Quality Teams were assembled to deal with the change at both management and production levels. Concurrent Engineering principles were applied to the design change to ensure cost-effectiveness, quality, manufacturing ability, reliability, and maintainability. Progress was tracked by two mechanisms: A "Time Line" that compared the time required for each step as performed "traditionally" in contrast to the time required given the implementation of this CALS vision. A "TQM Matrix" that charts progress was applied and displayed throughout the process depicted in the demonstration.

A key feature of the demonstration was the use of CALS technologies as they exist today, and the presentation of CALS futures. The technical emphasis was on adding "intelligence" to the data in the form of information conforming to the Product Data Exchange Specification (PDES).

SECTION 4

REFERENCE MATERIAL

4.1 Acronyms

The following acronyms are used in the IISS and its documentation.

ADL	Application Development Language
AI	Application Interface
AP	Application Process
APC	Application Process Cluster
ASCII	American Standard Code for Information Interchange
COMM	Communication
CDM	Common Data Model
CDMA	Common Data Model Administrator
CDMP	Common Data Model Processor
CEX	Conceptual to External Transformer
CGM	Computer Graphics Metafile
CS	Conceptual Schema
DBA	Data Base Administrator
DBMS	Data Base Management System
DDBS	Distributed Database System
DDL	Data Definition Language
DDP	Distributed Database Process
DTD	Document Type Definition
EIF	Enterprise Integration Framework
EIP	Enterprise Integration Program
ES	External Schema
FDFE	Forms-Driven Form Editor
FDL	Forms Definition Language
FE	Forms Editor
FP	Forms Processor
GDL	Graph Definition Language
ICAM	Integrated Computer Aided Manufacturing
IDEF0	ICAM Definition (Functional Model)
IDEF1	ICAM Definition (Data Model)
IDEF1X	ICAM Definition (Data Model-Extended)
IISS	Integrated Information Support System
IPC	Interprocess Communication Primitive
IRDS	Information Resource Dictionary System
IS	Internal Schema
ISO	International Standards Organization
JQG	Join Query Graph
NDDL	Neutral Data Definition Language
NDML	Neutral Data Manipulation Language
NTM	Network Transaction Manager
OSI	Open Systems Interconnect
PS	Presentation Schema
RAP	Rapid Applications Generator
RFT	Result Field Table

RW	Report Writer
SGML	Standard Generalized Markup Language
UI	User Interface
UIDS	User Interface Development System
UIM	User Interface Monitor
UIMS	User Interface Management System
UIS	User Interface Services
VT	Virtual Terminal
VTI	Virtual Terminal Interface

4.2 Component Structure Outline

The following outline gives the component structure of the Integrated Information Support System. The various components are named or described.

Common Data Model (CDM)
    CDM Maintenance
        Utilities
            DDL to NDDL Translator
            CDM Impact Analysis
            Compare Utility
        Access Control
        CDM Data Access
            Entry
            Update
            Edit
            Selective Retrieval
            Reporting
            Delete
    CDM Application Development
        IDEF1X Integration Methodology
        NDDL Processor
                620X41
            Initiator
            Parse NDDL Statement
            General Statement Processor
            Terminator
            Service Routines
                Input-Output
                Error Handling
                Data Base Access
        NDML Precompiler
            Parse NDML Statement
            Schema Transform
            Code Generation
                Query Decomposition
                Query Scheduler
                Query Processor
                Data Aggregator
    Distributed Database Manager
        Distributed Request Supervisor
            Inter-database Operations Sequencer
            AP / RP / Aggregator Coordinator

```
            Data Aggregator
                Join
                Union
                Not In Set
            File Utilities
                File Transfer
                File Delete


Network Transaction Manager (NTM)
    Monitor
            IISS Operator Interface
                Accept Command
                Service Command
                Deliver Status
            Status Maintenance
                Send Status
                Status Message Processor
    Message Processing Unit
    NTM Services
            Connection
            Communication
            NTM Requests
            Privileged
    File I/O Primitives


User Interface (UI)
    User Interface Development System
            Text Editor
            Application Generator
                Report Writer
                Rapid Application Generator
            Forms Editor
                Form Definition Language Compiler
                    IISS Invoked FLAN
                    Host Invoked FLAN
                    REVFLAN
                    MAKINC
                Forms Driven Form Editor
                    FDFE Driver
                    LISTIT
                    VIEW
                    EDTMOD
                            LISTFM
                            SAVFLS
                            INSFRM
                            DRPFRM
                            VIEW
                            STNMFD
                            EDTWHL
                                    FLFMST
                                    VALINP
                                    DRPWHL
                                    MODWHL
                                    INSWHL
```

                                GWHINP
                                MODFRM
                                COPFRM
                                FLWHST
                        EDTFLD
                                FLFMST
                                VALINP
                                GNXTFLD
                                DELFLD
                                MODFLD
                                INSFLD
                                GFDINP
                                GTCPFD
                                MODFRM
                                COPFRM
                                FLSTRC
                        LAYOUT
                                SCRMAN
                                        CHGPOS
                                        TRNSTR
                                        TRNSCR
                                EDTFLD
                        PRSCMD
                PRSCMD
                UNLINK
                COPY
                GETFLS
                RENAME

User Interface Management System
        User Interface Services
                Message Management
                Application Definition
                        Get Application Definition
                        Define New Application
                        Update/Delete Application
                Change Password
                        Process Password Form
                        Update UIUSER Table
                Forms Processor
                        User Interface Monitor
                        Scripting
                        Window Management
                Message Processor
                        Calculated Fields
                        Business Graphics
                Virtual Terminal Interface
                        Virtual Terminal Routines
                                Initialize VT
                                Get VT Data
                                Put VT Data
                                Terminate VT
                Device Driver Routines
                        Master Device Drivers

Slave Device Drivers
Reverse VTI
IBM Device Support
Application Interface
Electronic Documentation System
Document Type Definition Builder
Layout Editor
SGML Parser
SGML Tagger
EDS SGML Tagger
Formatter
EDS MacPaint to Postscript

Communication (COMM)
Generic COMM Protocol
Message Processor
Interhost Communication Primatives (IHC)
Initialize Communication Port
Send Message
Receive Message
Get Message
Cancel Receive Request
Terminate Communication Port
Interprocess Communication Primatives (IPC)
VAX
Create Mailbox
Send Message to Another Program
Receive a Message from Another Program
Get a Message from Another Program
Delete Mailbox
Release an Event Block
Start a Timer
Stop a Timer
Wait for Event
Terminate Program
Save Event Indicator
Request an Error Be Logged
Log an Error
IBM
Create Mailbox
Send Message to Another Program
Receive a Message from Another Program
Get a Message from Another Program
Delete Mailbox
Release an Event Block
Start a Timer
Stop a Timer
Wait for Event
Terminate Program
Save Event Indicator
Request an Error Be Logged
Log an Error