The computer-aided design environment project (COMRADE)

by THOMAS R. RHODES*
Naval Ship Research and Development Center
Bethesda, Maryland

BACKGROUND

Since 1965, the Naval Ship Engineering Center (NAVSEC) and the Naval Ship Research and Development Center (NSRDC), sponsored by the Naval Ship Systems Command, have been actively involved in developing and using computer facilities for the design and construction of naval ships. The overall goals of this effort, known as the Computer-Aided Ship Design and Construction (CASDAC) project, have been twofold—first, to achieve significant near term improvements in the performance of ship design and construction tasks, and second, to develop a long term integrated CASDAC system for all phases of the ship design and construction process.

While pursuit of the first goal has achieved notable cost savings, it has also produced a situation tending to delay the attainment of the second goal, that of an integrated CASDAC system. There soon were many individual batch-oriented computer programs, designed and operated independently of each other, involving relative simplicity, low cost, and short-term benefits, all of which contrasted against the complications, high cost, and projected benefits of an interactive integrated-application system. But yet, it was considered that a quantum improvement in the time and cost of performing ship design could only be realized through a coordinated and integrated approach. The real question facing the Navy was whether such an approach was technically and economically feasible.

In an attempt to demonstrate the feasibility of an integrated approach, members of the Computer-Aided Design Division (CADD) and the Computer Sciences Division (CSD) of NSRDC joined with members of the CASDAC office of NAVSEC to investigate and develop a prototype system.

The phase of ship design known as concept design was chosen to be modeled as an integrated system and to provide the macrocosm for studying system requirements. Some of the characteristics favoring choice of this phase, were:

- that as an initial phase of ship design, wherein basic features such as ship type and size, weapons and electronics systems, propulsion machinery and major shipboard arrangements were determined, it represented a critical phase where optimization over many alternatives could result in improved ship performance and lower development costs;
- that as an activity with a high level of creativity and analysis, in which operational requirements were transformed into a feasible engineering reality, it could be enhanced through application of computer aids;
- that as an activity with extensive interaction and information exchange among a multiplicity of engineers representing different disciplines (e.g., naval architecture, marine, mechanical, electrical, etc., engineering), it produced a dynamic atmosphere that was considered a “natural” for an integrated solution; and,
- that with relatively few engineering tasks and data requirements compared to later ship development phases, it offered a tractable situation for analysis and application of existing computer technology.

SYSTEM REQUIREMENTS

The initial study effort led the engineers and application programmers of CADD and the systems programmers and analysts of CSD along different, but complementary paths in viewing the system requirements—one view reflecting the engineering requirements of concept design and the other, the imposed computer requirements.

The engineering analysis sought to identify the various tasks, their contribution to the overall design process, their relationships, dependencies, data input and output requirements, and the role of the engineer throughout the design process. Each task was further divided to reveal the major steps and to explore the computer implementa-
tion of them. While a "building-block" approach to problem solving, involving a strong interaction between the engineer and the system, was desired, questions were raised as to how much flexibility the system should provide. Should the designer work at a low level with "atomic" engineering functions to incrementally describe and solve his problem, or should the designer work within a pre-established set of alternatives, where major decision points have been defined, and he should have only to choose and sequence among a variety of algorithmic procedures in which much of the problem structure has been imbedded within the program logic? While the "atomic" approach appeared more flexible and was conceivably more adaptable to new design situations, the latter approach was favored for the initial effort since, under this approach it was deemed that satisfactory design results could still be obtained for a broad set of problems, many existing application programs were amenable for use, and less sophistication was required to develop and use the system.

From the analysis of the overall design process, a good indication of program and designer data requirements was obtained. The required data was organized to reflect various logical associations, producing a large and complex data structure. To minimize data redundancy a distinction was made between data describing the characteristics of a particular ship and data that was common to many ships. This resulted in separate ship and catalog files and in having data associations both within and between files. This separation of data was favored also because the catalog files would be less subject to change during the design process than the relatively volatile ship file. and less queuing would be required during processing.

The data base was considered to be the crucial link through which information would be shared among designers and programs, and hence it represented the key element in an integrated system. The demand for information required that data management capabilities be made available to both the application program during execution, and to the designer working directly with the data base at the terminal. The large and complex data structure indicated that efficient and flexible techniques would be necessary to structure, store, access, and manipulate this data, and finally, some means of controlling access to the files would be required to preserve data integrity.

In addition to the analyses of the design process engineering requirements, consideration was also given to coordinating or controlling the overall design process. Although the designer would be responsible for performing design tasks, he would do so under the general direction of the project leader or design administrator. Task assignments and final acceptance of design results would normally be under the purview of this member of the design team, which implied that the system would need to be responsive to the administrator's role by providing controls over program and file access, and reports on the design status and system usage.

While the engineering analysis was directed toward identifying the elements and requirements of an integrated ship-concept design system, the computer science effort was directed toward providing general mechanisms that were adaptable to ship design and to similar situations where it was necessary to coordinate and integrate many users, programs, and data files.

The effort to produce a prototype ship-concept design system was termed the Integrated Ship Design System (ISDS) project, while the effort to develop a framework of capabilities for constructing integrated systems was termed the Computer-Aided Design Environment (COMRADE) project.

SYSTEM DESCRIPTION

From the analysis done during 1970, design and development of a prototype system was scheduled to begin the following year using NSRDC's CDC-6600 computer with the SCOPE 3.3 Operating System.

The NSRDC computer configuration, shown in Figure 1, provides remote access from interactive graphic, conversational keyboard, and batch stations to high-speed dual processors with extensive secondary storage. Conversational teletype (TTY) and medium speed batch (200 UT) capabilities are provided through the CDC INTERCOM Time-Sharing software, while high-speed remote batch and interactive graphic communications are serviced by the CDC EXPORT-IMPORT software. This configuration appeared satisfactory for a prototype effort; however, the relatively small main memory resource and the separate job schedulers for interactive graphics and conversational keyboards were considered major problems for program development and integrated operations. To minimize difficulties in a first level effort, exclusive attention was given to use of the more available conversational keyboard (TTY) as the principal designer interface to the system rather than to the more desirable interactive graphic terminal. However, some graphic applications were planned and these would interface with the data base for test purposes.

From a consideration of the ISDS requirements and an examination of related efforts, such as the Integrated Civil Engineering System (ICES), COMRADE proceeded to design and develop three major software components: an Executive system; a Data Management system; and a Design Administration system. References 4, 5, 6, and 7 describe these components in greater detail, however, the following summary gives an indication of their functions and capabilities:

• Executive System—An interactive supervisor program, functioning under the INTERCOM Time-Sharing system, that interprets and processes command procedures. Through supporting software,
known as the Procedure Definition Language (PDL), command procedures are defined as the complex sequence of computer operations necessary to perform a corresponding design task. Operations that can be automatically performed include: printing tutorials or data at the terminal; reading data from the terminal and passing it to programs through a System Common Communication area, and vice versa; setting default data values in system common; attaching, unloading, and purging files; initiating programs for time-shared or batch execution; executing most SCOPE control statements; and, altering the sequence of operations through conditional or unconditional transfers. Capabilities are also provided to control command usage through command-locks and user-keys, and to define unique “subsystems” of related commands.

Through the Executive capabilities, command procedures representing design tasks can be defined in such a way as to present a problem-oriented interface to the user and to hide distracting computer operations. Since computer actions can be dynamically altered during command processing, considerable flexibility for user decision-making can be provided. Finally, during execution of an application program step, the Executive is removed

![Figure 1—the NSRDC computer system](image)

![Figure 2—COMRADE command definition and execution phases](image)
from main memory, permitting larger residency by the application module. Upon termination of the module execution, control is returned to the Executive. (In Figure 2, the command definition and execution phases are figuratively shown as steps 2 and 3.)

- Data Management System—A library of FORTRAN-callable subroutines and user-oriented command procedures that provide data management capabilities to both the programmer and terminal user. Users may store, update, and retrieve data by name, retrieve data via queries on data attributes, cross-link data in different files through pointers, and in general define and process large, complex file and data structures.

The COMRADE Data Management System (CDMS) is hierarchically structured into three levels:

1. The foundation or interfaces with the SCOPE I/O operations consists of the direct access technique and directory processing programs. Variable length logical records can be accessed by name, where each name is “hashed” to form an index into a directory that can reference up to 516,000 data records. Previously used disk space is reallocated and a paged, circular-buffer is used to store and process data records.

   This set of programs, called the COMRADE Data Storage Facility (CDSF), can be used by a programmer to store and retrieve data records or blocks by name; however, at this level, he would be required to do his own internal record structuring and processing.

2. Built on this foundation are system procedures, called the Block-Type Manipulation Facility (BTMF), that enable the data record contents to be defined, stored, retrieved, and updated by name, thus enabling the programmer to logically define and process records without regard to the internal structure. At this level, the format of each unique block-type is defined before the data file is generated, and then, subsequently it is used to build and process corresponding data blocks. Each block-type can be logically defined as subblocks of named data elements. Each element can be of real, integer, character, or pointer data type, and can be single- or multi-valued (i.e., array). Single-valued elements can be “inverted” and used as keys for a query-language retrieval, and pointer-elements are used to form relationships among data records within one or several files. Sets of elements can be grouped together under a group name, with the group repeated as needed (i.e., repeating groups).

   Using the BTMF programs, the user can then process data logically by name while the system identifies and resolves the internal record structure.

3. While the second level capabilities were provided as subroutines for programmer use, the third level consists of user-oriented command procedures for terminal use. Utilizing the BTMF routines, these programs enable terminal users to define data records; to load and update files; to retrieve data by name or through a query language; and to obtain information on file characteristics, such as size, record names, block-types and inverted attribute names and ranges.

In Figure 3, the various CDMS components for file definition and processing are shown.

- Design Administration System—A set of command procedures and program capabilities to assist the design project leader or administrator to:
  - identify valid subsystem users and assign appropriate command and file access keys;
  - identify subsystem files and assign appropriate file locks and passwords;
  - selectively monitor and receive reports on subsystem activities, such as, names of subsystem users, dates and times, commands used, significant events, estimated cost of processing, etc.

Additional functions are provided to allow programs to dynamically attach and unload files during execution, and to prepare and cleanup necessary files during subsystem sign-on and sign-off.
STATUS

In the Spring of 1972, testing of the described COMRADE software began, using a selected set of ship design programs and data necessary to verify operational capabilities and to demonstrate the formation of an ISDS. Various interactive design commands were implemented, together with ship and catalog data files of limited size, for evaluation purposes. Figure 4 illustrates the functional components involved in the system development effort. During testing, engineers and ship designers who saw and used these capabilities and who were not involved with implementation, generally approved the system interface and the capabilities provided. Correspondingly, subsystem developers found the COMRADE capabilities to be convenient and necessary, but not always sufficient, thus providing feedback for corrections and further development.

While the current ISDS effort is directed toward constructing a set of design commands and data files sufficient to engage in actual ship concept design studies, COMRADE efforts have been concentrated on evaluating performance, “tuning” components for more efficient operation, documenting existing work, and planning major enhancements. For example, work planned for the coming year, includes:

- developing an interface between the Executive and the interactive graphics terminal for a balanced system environment;
- developing a report generator facility to conveniently retrieve and format data file information;
- developing a PERT-like facility to conveniently define, schedule, and monitor a subsystems activity; and,
- considering application of computer-networks for a computer-aided design environment.

While enhancements and improvements are planned, application of COMRADE capabilities to other areas of ship design, engineering, logistics, etc., will also be investigated. For example, planning is under way to develop a Computer-Aided Piping Design and Construction (CAPDAC) system which will integrate shipyard planning, design, and fabrication activities related to piping systems.

SUMMARY

In response to the Navy requirements for an integrated ship design system the Computer-Aided Design Environment project has developed an initial set of general software capabilities, not merely limited to ship design, that provide a framework for assembling and coordinating programs, data, and their users into an integrated subsystem. The three major COMRADE components are: the Executive System, an interactive program operating under the INTERCOM time-sharing system of the CDC-6700 computer at NSRDC, which can process a complex and varying sequence of computer operations in response to user-defined commands; the Data Management System, a direct access capability to process large complex file and data structures via subroutines or terminal commands; and, the Design Administration System, a set of subroutines and terminal commands used to control subsystem and file access, and to optionally monitor and report on selected information, such as user-names, date and time, commands used, cost estimates, etc., during subsystem operations.

These capabilities have been applied to several prototype application systems, most notably the Integrated Ship Design System, and several other application systems are being planned.

While the COMRADE mechanisms have been shown to work, they are merely “tools” in constructing integrated systems and therefore depend on careful system planning and judicious use by subsystem developers to achieve an effective man-machine system. Many other factors, such as the performance and capabilities of the computer system, the application of software engineering techniques to modular program construction, the organization of data files and data communication regions for programs and users, and the efficiency of program elements are all particularly significant in determining the appearance and performance of an integrated system.

The COMRADE capabilities, in conjunction with the ISDS project, have demonstrated the technical feasibility of constructing a convenient and effective computer tool that will provide guidelines and direction for continuing and similar efforts toward achieving a completely integrated CASDAC system.

REFERENCES