ECSTASY — AN ENVIRONMENT FOR CONTROL SYSTEM THEORY, ANALYSIS, AND SYNTHESIS

N. Munro

Control Systems Centre, University of Manchester Institute of Science & Technology, Manchester, UK

Abstract. A new software infrastructure has been created to support both present and future CAD facilities for control engineering. The various facilities provided by the software infrastructure; namely, the man-machine interface, database, graphics, design and simulation tools; are described. The development philosophy is discussed, and the development environment used is described. Future extensions to the environment are indicated.

Keywords. Software Infrastructure, CAD, Man-Machine Interface.

INTRODUCTION

Academic control groups in the UK have been developing CAD packages for control system analysis, design, and simulation since the late 1960's, when interactive computing became a practical reality. An interactive suite of design programmes to implement Rosenbrock's Inverse Nyquist Array design method for multivariable control system design was implemented at UMIST in 1971. This was rapidly followed by the Cambridge University package, known as CLADP, to implement MacFarlane's Characteristic Locus design method for multivariable control systems in 1973. Both packages also provided the well established classical control design methodologies due to Bode, Nyquist, Nichols, and Evans (Root-Locus) for single-loop control system design. In the following decades, many other control groups also developed their own in-house packages, most notable being the identification, simulation and design facilities developed by Astrom's group (at Lund, Sweden) in 1976, integrated by their interactive dialogue module INTRAC, and the federated design suite developed by Spaug at GE in 1981, until by 1982 the DELIGHT system emerged to provide user-friendly on-line optimisation facilities developed jointly by Polak (at Berkeley, California) and Mayne (at Imperial College, London). This latter facility is now widely recognised in the control academic world as an initial attempt to produce a software development support environment, and led to the EAGLES system developed by Herget at the Lawrence Livermore Laboratory in the USA.

Commercial software concerns in the USA had by then become actively interested, and by 1983 various integrated CAD facilities, such as Matrix-X (developed by Integrated Systems Inc., California) and Control-C (developed by Systems Control Technology Inc., California) emerged. Both of these packages lean heavily on the facilities contained in MATLAB (MAThematics and LABoratory), which was available in an early form in the public domain about 1982 and then commercially in 1984 (developed by the Maths Workshop Inc., California). Two of the most recent academically developed control CAD packages offering a wide range of facilities are an updated version of CLADP (offered by Cambridge Control Ltd. UK) and CSS (Control System Software) developed by Edmunds at UMIST (1987).

In 1986, the UK SERC's Control & Instrumentation Sub-Committee commissioned a survey of major control system CAD facilities developed by academic groups and commercial concerns, having previously in 1982/83 defined a future control CAD development strategy. For a variety of technical and commercial reasons, none of the packages considered was seen to fully meet the perceived needs of both the academic and industrial control groups, and a decision was made to develop a new software infrastructure called ECSTASY (Environment for Control System Theory, Analysis and SYnthesis). This environment was to be developed to service three primary aims:

(1) To provide a common software base to assist the ready transfer of CAD tools between the various UK academic groups, and hence reduce the previously fragmented development of such facilities.
(2) To act as a means of transferring the control system design tools already developed in the academic world into industrial use in a consistent framework.
(3) To provide a common software base for the development and testing of new design algorithms and facilities.

THE ECSTASY SOFTWARE INFRASTRUCTURE

The ECSTASY infrastructure contains six major components, as shown in Figure 1, namely, a Man-Machine Interface, a Database and Data Management Facility, a set of Matrix Manipulation Tools, a facility to Manipulate System Descriptions in several mathematical forms, a Documentation/Report Preparation Facility, and a Simulation Tool.

The Man-Machine Interface facilities provided consist of both dialogue tools; allowing various dialogue styles, such as 'Question and Answer' mode, 'Command' mode, and 'Expression' mode, and various dialogue mechanisms, such as keyboard, menu, forms and button input. In addition, the MMI provides both graphical output of data, in the form of time and frequency-response plots, etc., and the input and output of system schematic representations. The user interface is provided in a windowing environment allowing several windows, such as a scrolling dialogue window, a graphics window, an editting window, and a read-only window, to be active simultaneously even on dumb terminals.

The graphics facilities implemented provide both

2-D and 3-D plotting capabilities with scaling, rotation, translation, panning and zooming of images. This enables the currently known graphics requirements for time-domain and frequency-domain plots to be readily achieved and also the generation and inspeciton of various 3-D maps. Graphics metafiles allow the archiving of displayed images, and the subsequent recall of these images for the production of hard copy on a number of different devices, or technical reports.

The Database needs of the control engineering community differ radically from those of the commercial world which are already catered for by various existing facilities of this type. Commercial data tends to consist of very large amounts of relatively simple data types, whereas control engineering data consists of modest to large amounts of complex data objects with complex relationships between these objects. A variety of primitive data types have been defined and a range of operations such as creation, deletion, renaming and copying of data entities is provided. All data entities within the database have date-stamps, access-protection, and relational pointers associated with them. The database acts as a consistent central store of all data required by the other major facilities in the infrastructure and also of the data required by the design and synthesis tools connected to the infrastructure.

The necessary Matrix Manipulation Tools, which form the kernal of most of the current control system design and synthesis algorithms, and in particular are vital to the present research into 'robust' control system design techniques, are provided by the commercially available MATLAB package. In particular, PRO-MATLAB (which is the version of this facility optimised for use on the SUN, VAX and μ VAX range of machines) uses the robust numerical subroutine libraries in EISPACK and LINPACK. This facility provides a very wide range of easy-to-use matrix manipulations and allows new functions to be created on-line. It has a user friendly interface and can easily be combined with a Control System Analysis and Design Toolbox, providing a wide range of well established parametric and non-parametric system identification tools. The MATLAB package has also been interfaced to the NAG and SLICE subroutine libraries, making available various optimisation procedures and other algorithms.

The System Description Manipulation facility implemented provides a means of entering a system description graphically, essentially in block-diagram form. The various system components or subsystem blocks may be described in various mathematical forms, such as transfer-function descriptions, state-space equations, or non-linear differential equations and algebraic relationships, and may then be interconnected to create the desired system structure by graphical editting actions. For control system analysis and design purposes, this facility must be able to create the required overall mathematical model in a consistent format. Also, this facility will pass the necessary system information, in either linear or nonlinear equation form, to the simulation facility for time-response studies.

The Simulation Tools provided in the infrastructure allow the easy interactive simulation of both linear and non-linear system models. They also provide steady-state finders and linearisation procedures so that locally linearised models can readily be generated at various system operating points. The simulation facilities implemented are provided by the simulation language ACSL, which is already widely used by both the academic world and practicing engineers, and by TSIM, which is used by various MoD contractors in the aerospace industries.

The Documentation facilities of the infrastructure provide a mechanism for the production of both on-line and off-line documentation. In this respect, documentation will take several forms. On-line documentation will consist of both simple 'help' facilities as well as on-line access to the full user documentation, which describes the available facilities and how to use them. Off-line documentation will take two forms; namely, a printed version of the latter (i.e. for users of the facilities provided) and a printed manual giving internal details of the infrastructure facilities for researchers developing new algorithms and new facilities. The commercially available text formatting and production systems TeX (with Preview and Postscript) and the recent FRAMEMAKER can both be used for this facility.

DEVELOPMENT PHILOSOPHY

The initial implementation of the infrastructure is to be carried out on SUN-3 graphics workstations under UNIX, with further immediate implementations targetted for VAX machines running under UNIX and VMS and μVAX machines. Although some of the major components of the infrastructure are written in FORTRAN IV and some in Language C, all new code developed will be written in C. The resulting C-code is verified using the UNIX LINT verifier to ensure transportability onto other machines.

Considerable effort has been devoted to the creation of a user-friendly Man-Machine Interface. This must be consistent and easy-to-use with respect to each of the other major infrastructure components. The user interface must also detect any relevant event in the user input which requires the transfer of data from the ECSTASY database into the local datastructure of the facility being used, and vice versa. It is considered important that a high degree of flexibility is also built into the ECSTASY infrastructure. Major components such as the database, graphics facilities, and simulation tools must be capable of being replaced or augmented by other such facilities as the needs demand. The need for various simulation tools has already been mentioned above. However, it may be necessary to have separate database facilities; one to manage very large data sets and one to manage small data sets, in an efficient manner.

The useability aspects of the infrastructure must be high; i.e. it must provide for the implementation of existing control system analysis, design and synthesis tools and also for the prototyping and development of new tools. It is also essential that a clearly defined mechanism is provided for the connection of foreign CAD packages; i.e., other CAD facilities developed elsewhere by academic, industrial or commercial concerns. It is equally important that the initial infrastructure is capable of evolution to take account of new developments in software engineering, computer science, and artificial intelligence. A significant effort is now being dovoted by the engineering community to the exploitation of expert systems design aids and the use of object-oriented programming environments. It is more than likely that the next generation of control system CAD tools will be written in languages such as PROLOG and COMMON LISP, although at present this is a somewhat volatile area.

THE DEVELOPMENT ENVIRONMENT

The current project to develop an embryo version of the ECSTASY infrastructure has been carried out at UMIST on a very short time-scale, with delivery to the SERC of a facility suitable for β-test evaluation in mid-1988. This could only be done by making use of a commercially available software engineering environment, and for this purpose the Software Engineering Tools (SET) developed by the Computer Aided Design Division of PA Management Consultants were used. This environment is based on the concept of software building blocks, as shown in Figure 2, where the user is considered to interact with a desired set of applications programmes primarily through a graphics display terminal.

The display interface is designed to handle a variety of both dumb and intelligent graphics display terminals of the DEC, Pericom, Tektronix, SUN and Apollo varieties. It is also responsible for providing the window management facilities (windowSET) on these devices, allowing a limited but useful windowing facility on the dumb terminals. Up to 20 windows can be open simultaneously. The interaction handler (inSET) looks after the decoding of the user input in a variety of forms. A wide range of interaction mechanisms such as on-screen menus, pop-up menus, forms, graphics, tablet, mouse, function keys, and keyboard, as well as input from macros and command files, can be handled. The inSET module provides a fairly robust input facility with the various dialogue requirements being defined and implemented using state-transition networks. The graphics facilities provided by the graphics module (graphicSET) are fairly comprehensive, as mentioned earlier, and include 2D and 3D drawing capability together with lines, surfaces, text, colour, and an independent set of text fonts. The database management facilities provided by the SET environment consist of a datastructuring system (dataSET), which allows the definition of various schemas to be used with different data sets, and a relational database management system. The latter is a comprehensive implementation of IBM's Structured Query Language, which provides high performance whilst supporting large, complex data structures. Some modifications to the dataSET module were implemented to be able to access to data entities and types by name, and to allow the creation of new data types at run-time.

An important feature of the SET tools is the operating system interface module (onSET), which buffers the implementation from the operating systems. This feature provides the portability of the ECSTASY infrastructure onto a variety of other machines with different operating systems such as UNIX and DEC's VMS. The software engineering tools being used provide a development environment which allows both the rapid prototyping and testing of the user interface and the database aspects of the infrastructure. Skeleton code generators are provided which create both C-code or FORTRAN-code based on the interaction model defined by the user. This code can be readily compiled and executed to ensure that the desired behaviour will be achieved before specific application code is added, thus accelerating the development, debugging and documentation stages.

RELATED ACTIVITIES

The embryo infrastructure being created using the software engineering tools described briefly above will be interfaced to various control system design and identification tools to provide a more complete facility, as indicated in Figure 3, before being released for evaluation at selected academic and industrial sites in mid-1988. Several features indicated in this diagram and not described in detail previously are SIMNON (a simple yet powerful and easy to use simulation facility) developed by Astrom's group at Lund Institute of Technology in Sweden.

The Swedish group are also developing a powerful graphical system description facility which offers a higher level of abstraction than the block-diagram approach. Work is also being carried out under the SERC's Computing and Design Techniques for Control Engineering (CDTCE) initiative on graphical front ends by the research group at Swansea University. Various database studies and, in particular, their design for control engineering applications are being carried out by Warwick University and by Cambridge University, where the latter are also exploring the use of object oriented computing environments for future control CAD developments. Several universities, such as Cambridge, Lund, Sheffield and UMIST, are investigating the use of Expert Systems as design aids. Research studies into the development of new design tools for nonlinear systems are being carried out by the groups at Imperial College and Sussex. The continuing development of design tools for multivariable control systems is being pursued by UMIST and Cambridge. The development of self-tuning control systems and their industrial application is being carried out at UMIST and Oxford University. New design tools for the development of robust control systems in the sense of H_∞ robustness is being carried out at Oxford, Cambridge, Imperial College and Strathclyde Universities, and the practical application of optimisation procedures is being actively pursued by Imperial College and Bangor University.

Recent research in the USA has indicated that known difficulties with certain design and synthesis algorithms could be significantly improved if the operations concerned could be reliably performed in symbolic form. Well developed symbolic algebra packages, such as MACSYMA, provide such tools. The use of MACSYMA, as indicated earlier, would provide symbolic model construction and simplification facilities. It would also allow the use of the more compact polynomial matrix algebra tools in the development of analysis, design, and simulation facilities.

The above brief outline is intended only to indicate that in addition to the development of new CAD facilities, further research is being actively sponsored under the SERC's CDTCE initiative into the development of new design tools, which it is intended will be implemented in the proposed new software infrastructure or its future versions. Figure 4 indicates a possible future form of this type of infrastructure. The introduction of the more modern computing components indicated will have to be carefully considered and properly phased so that a 'bumpless transfer' is achieved.

ACKNOWLEDGEMENTS

The author wishes to gratefully acknowledge the financial support provided by the SERC's Control and Instrumentation Sub-Committee for the development of the ECSTASY infrastructure and also the help and advice being provided by the SERC's Rutherford-Appleton Laboratory staff and Professor K.J. Astrom's group at Lund (Sweden). The diagram reproduced as Figure 2 was provided by the Computer Aided Engineering Division of PA Management Consultants.

BIBLIOGRAPHY

Rosenbrock, H.H. (1974). Computer Aided Control System Design. Academic Press.

Patel, R.V., and N. Munro (1982). Multivariable System Theory and Design. Pergamon Press.

Pang, G.H., and A.G.J. MacFarlane (1987). An Expert Systems Approach to Computer-Aided Design of Multivariable Systems. Springer-Verlag.

Munro, N., and B.J. Bowland (1981). The UMIST Computer Aided Control System Design Suite : User's Guide. Control Systems Centre, UMIST.

Edmunds, J. (1981). Cambridge Linear Analysis and Design Programs. Control Engineering Group, University Engineering Department, Cambridge.

Weislander, J. (1979). Interaction in Computer Aided Analysis and Design of Control Systems. Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

Spang, A. (1981). Workshop on Control Design. G.E. Research and Development Center, Schenectady, New York State, USA.

Polak, E. (1981). Interactive Software for Computer-Aided-Design of Control Systems via Optimization. 20th IEEE Conference on Decision & Control, San Diego, California, USA.

Lawver, B., and P. Poggio (1985). EAGLES Requirements. Engineering Research Division, Lawrence Livermore National Laboratory, Livermore, California, USA.

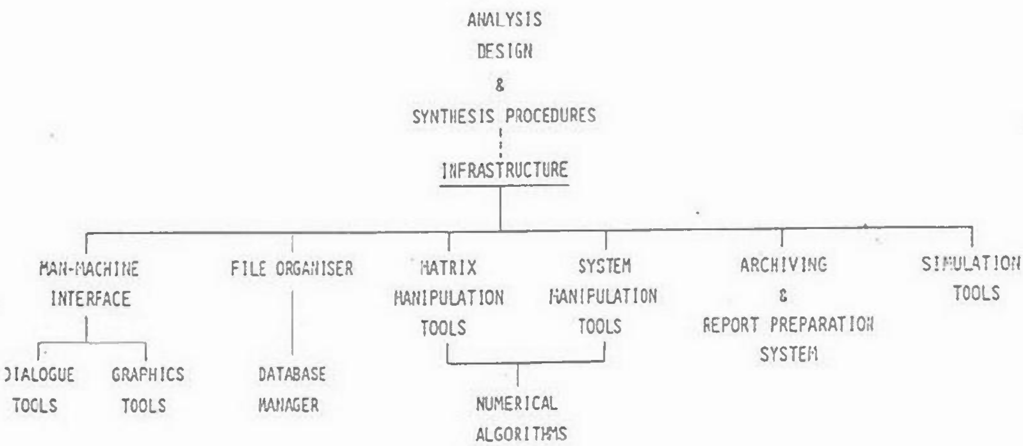

FIGURE 1

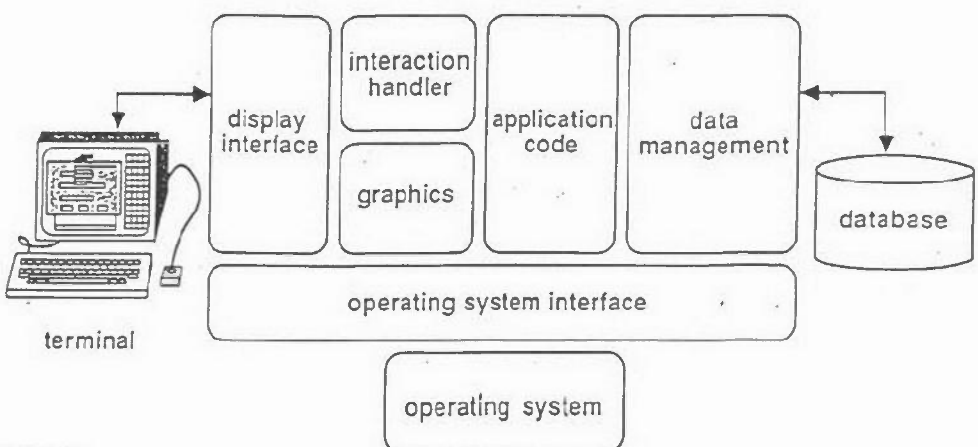

FIGURE 2

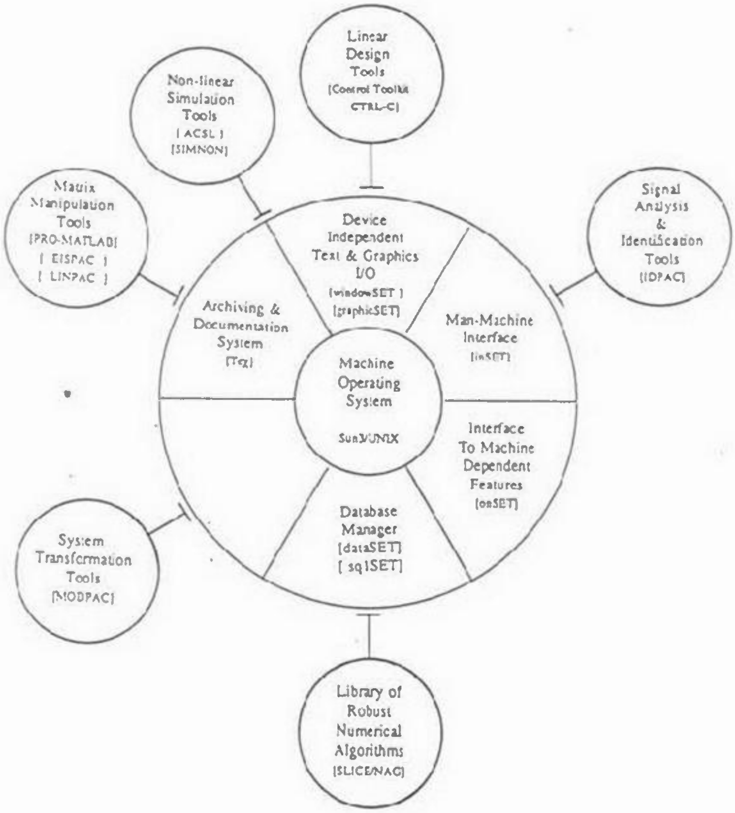

FIGURE 3

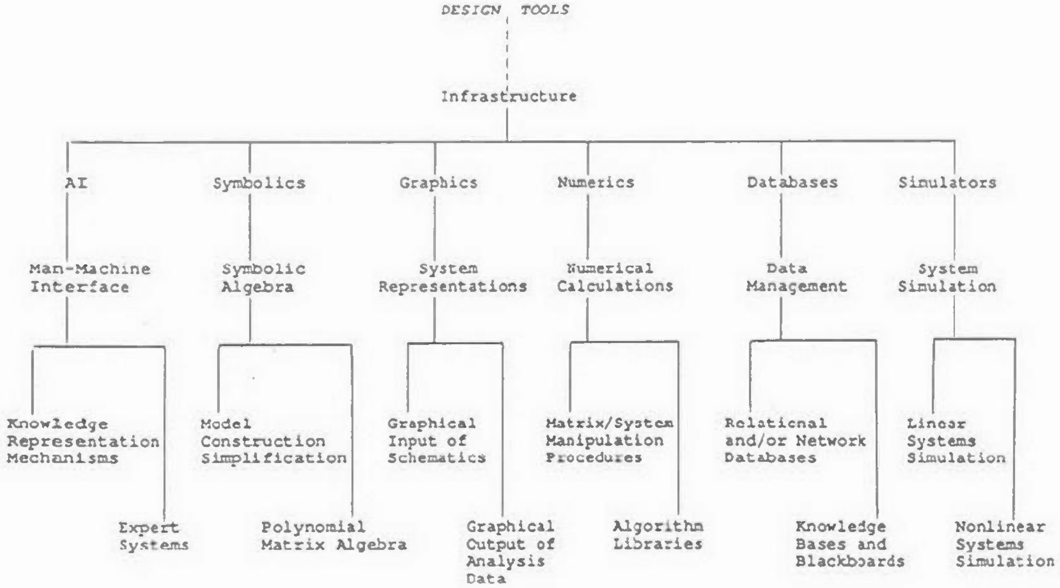

FIGURE 4