

# Selection Criteria for CASE Tools

Geoff Beckworth  
Department of Computing and Mathematics  
Deakin University  
Geelong 3217  
Australia

Fax (052) 472028  
Phone (052) 471393  
email [gbeck@deakin.OZ.AU](mailto:gbeck@deakin.OZ.AU)

## Abstract

This paper details the selection criteria which were used to choose CASE tools for use in Undergraduate and Postgraduate studies at Deakin University. While the objectives for the use of CASE tools in a tertiary education environment are different from the business world, the same process of evaluation should be applicable and the results of general interest. The study, which took place in 1990 and 1991, considered 20 different CASE tools. A taxonomy of CASE tools was identified according to the level of support the tool provided during the life cycle of software development. The tools investigated covered the broad range of functionality identified by the study. The goals for a suitable CASE environment for our courses and the evaluation criteria were established. An overview of the products investigated is provided as well as a matrix showing the products against the characteristics of CASE tools.

## **Introduction**

This research was requested by the Computing and Mathematics department as part of a curriculum review of the current Information Systems (IS) and Computer Science (CS) streams at Deakin University. The committee, led by Professor Brian Garner, felt that CASE tools should play an integral part in our courses for the future and so an investigation of the current 'state of the art' software was carried out.

Necco et al (1989) recommended after a study of 100 organisations that systems analysis and design courses should include material about the appropriate use of CASE tools and should attempt to provide students with the opportunity to use CASE tools in a project environment. They also recognised the potential for research into the effective use of CASE tools and how they can further improve systems analysis and design tasks. Norman and Nunamaker (1989) performed a study of the various functional and behavioural aspects of CASE and attempted to determine the impact it has over manual methods for software engineering productivity. Their conclusion was that in the 1990's, information systems will be more sophisticated, integrated, interactive and distributed and therefore their development must be assisted by CASE technology. The importance of CASE tools for project work in a Software Engineering course has been identified by Grant (1991).

A preliminary study involved an analysis of the CASE tool domain and we established a description and model of the main components. Other features were identified which related to the environment in which they would be used. A list of questions was developed which when matched against our goals would form the basis of enquiry when reviewing the products. The information was obtained from comprehensive demonstrations of the software by the vendors as well as their marketing materials and discussion with other users. The results were presented to the department.

### **What do CASE Concept Encompass?**

A CASE environment allows the developer to model the system with diagrams, define functional and data objects, identifying relationships between components of the system, and provide documentation (for example Parkinson 1989, Gane 1990). The results of the work is stored in a central repository in a minimally redundant form. Analysis can be performed on the information so that problems can be identified during development. The repository provides a source of information for documentation. In a complete CASE product, the application can then be generated from the analysis and design specifications previously captured. The software can vary in sophistication from simple diagramming tools to a complete implementation using a comprehensive method. This range of capability will now be defined in more detail.

### **The CASE Tool Domain**

The CASE tool domain contains a variety of products and features which are dependent on the application area, the hardware platform, the operating system and the user interface required. The tools cover the planning, analysis, design and construction stages. The way in which these stages are implemented depends on whether there is a method associated with their use or whether the

techniques are generic. The ability to import and export data to other tools and to retrofit existing systems are also features which can exist. The power of the tool depends on the level of sophistication of the central repository of data and this can vary from a simple data dictionary to a knowledge based encyclopaedia. These features are considered below in more detail.

**Application Area :** This may vary from commercial systems which might be batch or transaction oriented system to real-time systems which may involve process control or embedded systems.

**Hardware :** Many of the tools allow development on an intelligent workstation which allows developers to work on individual parts of the project. The results of the analysis and design can then be exported to the target hardware for construction and implementation.

**Operating System :** This largely divides the products into those that can be used with Unix and those can run on DOS or OS/2 with exporting to IBM or DEC environments. There is movement toward products being available on a variety of platforms. The networking environment will determine whether teams have multi-access to the data and the tool will determine how effectively the data can be shared while maintaining integrity.

**User Interface :** This will determine whether how effectively a developer can work with different levels of the tool at the same time.

**Integration :** The tool can exist as a fully integrated tool covering all phases of the life cycle or it may only perform analysis and design. If this is the case then there may be the need to integrate it with other tools. The ability to import and export the data can be important issue.

**Repository :** This is the heart of the tool and acts as a meta-database to store information about the functions, procedures, processes, data models, process models, entities and their relationships and so on. The data should allow the user to view it in a variety of ways such as data flow diagrams, structure charts, in a consistent manner with appropriate rules being applied. The repository can exist at different levels of sophistication. In a powerful repository all information is stored, without redundancy, in only one place. If it is distributed then it should also be able to be consolidated into a master repository. A primitive repository would enforces that the user has to provide the link between the diagram components and the data dictionary manually. This can result in inconsistencies which take time to resolve.

**Workbenches :** Workbenches are the tools which most people associate with the market terminology of CASE. There are a range of tools which are an important part of system development and these will be discussed later. The workbench tools can be subdivided further according to the level of lifecycle support.

- (i) Planning - association matrices, organisational hierarchy, subject area, presentation graphs, function hierarchy, function dependency, document graph;
- (ii) Analysis - entity relationship models, entity life history, data model diagrams, data flow, process hierarchy, process dependency, process action, structure chart, control specification, state transition, matrix processor, prototyping;
- (iii) Design - dialog flow, screen design, report design, prototyping, pseudocode developer, procedure action diagrams, data structure, database design, database construction; and
- (iv) Construction - code generation for Cobol, C, Ada, DB2 etc, test data generation, performance optimisation, configuration management.

Upper CASE defines tools that cover the planning and analysis phases. Lower CASE defines tools that cover the design, construction, implementation and operation phases. Integrated CASE (I-CASE) cover the whole life cycle.

#### Application

##### Generation :

The 4GL and relational database environment provides a sophisticated development environment where applications can be generated and prototyping is supported. These tools in principle include data modelling, specification facilities and efficient performance.

##### Retrofitting :

This is a term which encompasses reverse engineering and re-engineering. Reverse engineering takes an existing system and recovers the original design specification from the current physical implementation. Re-engineering performs reverse engineering, modifies the existing system and then creates a new improved system.

IPSE : This stands for Integrated Project Support Environments and is a group of tools which provides support for project management, configuration management, version control, requirements tracing, quality assurance.

### **Background to Deakin University's Environment**

The Department of Computing and Mathematics currently offers a second year course in Systems Analysis and Design which covers the life cycle of software development using a structured approach of process and data modelling techniques.

There is a third year course in Software Engineering (SE) where alternative methodologies are studied as well as a more detailed study of requirements analysis, detailed design, quality assurance, system testing and configuration management. The integration of Artificial Intelligence with SE has also been introduced. The students complete a major project in teams of no more than five with one of the students acting as project manager to control the development of the software. The team must find an external project and communicate with the users to establish their requirements.

A plan is submitted and if it is accepted they then proceed to design and construct the system to meet the requirements. Reports are submitted by the teams as the various 'milestones' of the are achieved.

An honours subject in Software Engineering was introduced in 1991 where the emphasis is on real-time systems and it is anticipated that a Masters course will be developed in the future.

The hardware environment at Deakin includes Sun 4's as file servers with Unix as the operating system and workstations or PC's. The PC's are supported by PC-NFS and are capable of being used as terminals or running DOS software. The latest version of PC-NFS supports Windows 3.0.

Software available in the Unix environment includes a relational database, Unify 2000, and a 4GL, Accell/SQL, with a wide range of 3GL languages including C, Pascal, Cobol and C++.

## **Goals and Requirements**

Given the current environment we established the following goals:

- (a) to provide a CASE toolkit as defined by McClure (1989), which can provide access for large volumes of students at an affordable cost;
- (b) to provide a sophisticated CASE tool environment with an 'Open Architecture' as defined by Wasserman (1990) for advanced and postgraduate students;
- (c) to provide a CASE tool which is capable of supporting analysis and design phases for the development of commercial databases;
- (d) to provide a CASE tool which is capable of supporting the development of real-time applications;
- (e) to provide a CASE tool which is capable to become part of a toolset which will include a 4GL, a DBMS which has forms and report generation. The research was directed toward upper CASE tools which cater for front-end analysis and design. During the investigation several integrated CASE products were also reviewed;
- (f) to consider the tools in terms of the educational level of the students; and
- (g) to establish if it is possible that one product can provide a suitable environment for second year, third year and postgraduate students whether simple products are needed for second year, commercial products for third year and experimental research products for postgraduates.

## **The Selection Process**

A systematic selection process was adopted similar to the approach suggested by Chantico (1989) which involved:

- (a) establishing the functional requirements (desirable and necessary) of the CASE software required;
- (b) obtaining literature and information from vendors;
- (c) using checklists to review the packages against the requirements. A number of these are suggested in Chantico (1989) covering general, technical, installation and operation,

modification and maintenance considerations, vendor capabilities and personnel training. General software package requirements must also be considered. The use of metrics can also be a useful procedure;

- (d) evaluating the products against the desirable requirements that have been established;
- (e) selecting products for demonstrations, vendor visits and hands-on trial; and
- (f) making a recommendation based on factors including cost, learning curve, vendor.

### **Desirable Features of CASE Software Products.**

To assist us in assessing the products we established a list of desirable features and questions relevant to these features. This checklist along with our requirements and goals provided the basis for selection. The following features, obtained from a variety of sources including, Mimno (1990), Yeh (1990), Burkhard (1989) and Chantico (1989), are considered to be generally the most important in CASE products;

- (a) support for the entire application development life cycle;
- (b) use of an information repository;
- (c) support for diagramming techniques;
- (d) availability of the product for the existing hardware platform;
- (e) open software architecture;
- (f) compatibility with current software - DBMS, application generators;
- (g) support for structured methodologies;
- (h) support for prototyping tools; and
- (i) use of a well-defined life cycle process.

### **Important Questions**

Important questions were drawn from the literature and our own requirements analysis are outlined below.

*Life Cycle Coverage* - What parts of the life cycle is the tool applicable for?

*Method* - Is the tool based on a recognised method (for example, information engineering, structured analysis and design)?

*Techniques* - Do the diagramming editors support various techniques? For example in data flow diagrams, are the Yourdon/De Marco and Gane/Sarson techniques supported?  
Is the Jackson technique supported in structure charts?  
What is the technique used for entity relationship modelling?  
Is there consistency checking and balancing?

*Integration* - Does the tool move from one function to another consistently with a seamless appearance. A complete integration would involve moving

smoothly through front-end CASE design to a forms/4GL applications builder, database, network software and query and report generators.

- Repository -* What is the level of sophistication of the data dictionary? (The level will determine if specifications can be shared across functions).  
Is the repository in a format which can be accessed by a DBMS?  
Does the structure of the repository allow data to be accessed by the various tools?  
How is the repository populated with meta-data regarding the data and processes?  
Is the repository held on individual workstations, centrally or in distributed fashion?
- Functionality -* Does the software should provide error detection, debugging facilities, data integrity, backup and recovery, documentation and auditability.
- Planning -* Does the tool provide the means to model the organisation?
- Analysis -* What type of analytical tools are available?  
Is consistency-checking within the data flow diagrams performed immediately or on request?  
Are there rules associated with various diagrams?  
How are violations to the methods reported?
- Design -* What type of design phase tools are available?  
Can the tool generate physical design specifications from logical design specifications?  
Are design specification reports generated automatically?  
Do the design tools allow diagrams to be exploded to many lower and more specific levels?
- Code Generation -* Does the tool-set generate code from the graphical specifications?  
Can it generate code suitable for a variety of hardware?  
What is the extent of this code:  
Is it processing logic, data schema, an/or input/output code?  
If schema is produced, is it normalised?
- Documentation -* Does the tool produce documentation for the developer?  
Is there integration with a wordprocessor?  
Can "live" links be established between the tool and the wordprocessor?
- Forms and Graphics -* Does the tool provide a screen designer for menu dialogues, screen panels?

	<p>Do the forms have full repository support?</p> <p>Are multiple windows supported?</p> <p>Are pop-up menus supported?</p> <p>Is there forms navigation logic through menus?</p>
<i>Query and Reporting -</i>	<p>Is the query language easy to use so that the developer can design a simple query as well as a complex query across several tables?</p> <p>Is the report write integrated to the repository?</p> <p>Can the report data be downloaded to decision support software?</p> <p>Do report formats exist as templates?</p> <p>Can templates be easily constructed?</p>
<i>Application Maintenance -</i>	<p>What is maintenance performed on, the design specifications or the code?</p> <p>What is the impact of altering the source code on the repository?</p> <p>Can existing applications be reverse-engineered?</p> <p>Can existing applications be re-engineered?</p>
<i>4GL -</i>	<p>Does the tool incorporate a fourth-generation language of its own to enable links with the 'outside' world to be created?</p>
<i>DBMS -</i>	<p>Can a physical design be generated for various target DBMS (relational, hierarchical and/or network)?</p>
<i>Architecture -</i>	<p>Is the architecture of the system open so that the working environment can be customized and extended?</p> <p>Does the format of the repository enable integration across various CASE products?</p>
<i>Hardware requirements -</i>	<p>What is the hardware configuration required to run the software?</p>
<i>Multi-user access -</i>	<p>Can the members of project teams share common data?</p> <p>Can data be shared between project teams?</p> <p>What level of record locking exists in the repository?</p> <p>What security controls exist governing single-user and multi-user access?</p> <p>How is individual work interfaced into the system?</p>
<i>Performance -</i>	<p>If there is code generation, does the code perform efficiently?</p>
<i>Suitability -</i>	<p>What academic level is the tool suitable for?</p> <p>How easily will it fit into the current work procedures?</p> <p>How much training will be required?</p> <p>For how long will staff be unproductive while learning to use the tool?</p> <p>How long will it take the students to learn how to use the tool?</p>

<i>Cost -</i>	How much is it? Does the vendor have an educational licence agreement? What are the maintenance arrangements?
<i>Vendor -</i>	Is the CASE product distributed in Australia? How long has the vendor been selling the CASE tool? Does the vendor have staff who are experienced with the software? Does the vendor provide training, maintenance and upgrades and at what cost?
<i>Future -</i>	What are the prospects for the future development of the tool? Some issues here are distributed database support, the use of expert systems, cooperative processing support, graphical user interface, portability, heterogenous systems gateways, and security.

### **Summary of Products Reviewed**

<b>ADW</b>	Application Development Workbench from Knowledgeware provides an integrated environment implementing the Information Engineering method. An intelligent repository manages the data provided by a complete set of diagramming editors. It runs on PS/2's and provides code generation for IBM mainframe.
<b>Corvision</b>	Corvision covers the full life cycle and provides a partial implementation of the Information Engineering method. A fully integrated code generator produces optimized code for the DEC VAX VMS environment.
<b>Deft</b>	Deft is a tool for the Macintosh which provides structured analysis and design. It can generate schemas for RDBMS's and will support reverse engineering for existing database designs.
<b>Design/1</b>	Design/1 is a component of Foundation and provides the tools for analysis and design. Foundation also contains Plan/1 which allows strategic planning and Method/1 which is the life cycle methodology.
<b>Easy Case</b>	EasyCASE is a limited function tool with diagramming editors for structured analysis and design. It runs on a PC AT with VGA.
<b>Excelerator</b>	Excelerator has a comprehensive set of graphic editors for analysis and design, screen and report design and provides extensive documentation. There is a real-time version and XL/Recover converts COBOL application source code into design definitions. It runs on PC's and DEC VAXstation. Version 2.0 runs on OS/2.

<b>Foundry</b>	Sylva Foundry allows the creation of a customised systems engineering environment. It can be integrated with other tools and interfaces to RDBMS, 3GL and 4GL generator tools can be created. A frame library provides support for structured analysis and design.
<b>IEF</b>	Information Engineering Facility is based on the Information Engineering method and is an integrated tool. It covers the entire life cycle from strategic planning to monitoring of systems in production. The planning, analysis and design toolset operates on PC-compatible workstations while the construction toolset, including code generation, database generation and diagram testing, resides on an IBM mainframe.
<b>ISEE</b>	This is an integrated CASE tool for business systems which runs on a multitude of hardware platforms. It can be used in a heterogeneous environment with concurrent access being managed.
<b>Lansa</b>	Lansa is a product which runs on an AS400. It is a sophisticated 4GL producing interpreted code.
<b>LBMS Sys. Eng.</b>	LBMS Systems Engineer uses the LSDM method and also supports SSADM. It runs under DOS with Microsoft Windows and under OS/2. The tool set provides complete integration including Strategic Planner, Repository System, Code Generation and Reverse Engineering.
<b>Oracle CASE</b>	Oracle CASE* provides a top-down methodology for the planning, analysis and design of systems applications. It is integrated with the Oracle DBMS application development environment.
<b>PC Prism</b>	This is one of the third party products which can be used with Excelerator. It supports the strategic planning phase using Critical Success Factor analysis and various matrices.
<b>POSE</b>	POSE consists of nine design tools that can be used for both data-driven and process-driven system analysis and design using a common data dictionary. The system will run on an XT,AT or PS/2.
<b>PowerCASE</b>	PowerCASE is a CASE tool marketed by Cognos that runs on a wide range of operating systems. It uses a method called PowerDesign and integrates with PowerHouse and SQL relational databases.
<b>Reform</b>	Reform is a 4GL environment that provides for the design, development, implementation and maintenance of DBMS independent applications. There is an interface which allows design data to be exchanged with other CASE tools.

- STP** Software Through Pictures supports structured analysis and design methods for commercial and real-time systems. It runs under Unix with third-party products that can provide support for the full life-cycle.
- Synon/2** This is a lower case tool/application generator that runs on an IBM AS/400. Entity relationship modeling is used to design the data model.
- System Architect** System Architect is a design tool that operates under MS-DOS and Windows. It allows the design of transaction processing and real time systems.
- System Developer II** This is a PC based product providing support for structured methods. The environment can be customized.

**Teamwork** Teamwork is a UNIX based product which provides support for structured methods for information systems and real-time applications. It can interface to the RDBMS, Ingres, and Telon, a code generator).

These products were evaluated on the basis of the Desirable Features and Important Questions presented earlier. Exhibit 1 provides a summary of some of the important information arising from the project. For further detail, see Beckworth (1991).

## Results

Selecting one CASE tool from all of those evaluated is a difficult task. Given that we were trying to cater for a range of student audiences who would be using different hardware platforms we decided to opt for several tools.

- (a) Easy CASE will be used by first year students and students from programming courses at other levels to further develop their structured programming skills;
- (b) Excelerator will be used in the second year course. We successfully trialed the product in 1990 and have an investment in time and course development.
- (c) We decided on Software Through Pictures (STP) for the Unix environment, to cater for advanced undergraduate and postgraduate courses.

The main reasons for these decisions were :

- (a) the educational licensing arrangements available for these products;
- (b) their support for recognised structured analysis and design techniques;
- (c) the appropriateness of the levels of analysis and documentation support;
- (d) STP's open architecture, which will suit more advanced students. (It also allows the user to modify and/or extend the project dictionary structure and to use a program design language); and
- (e) the tools extensibility and customisability.

An further outcome of the project has been the establishment of a 'Centre for Software Engineering' at Deakin. The centre will be actively engaged with industry in research, joint projects and training.

Tools that cover other areas of the life cycle such as code generation and IPSE will be investigated further will continue and further consideration will be given to integrated CASE. We will also continue our evaluation of products throughout 1991.

## **Conclusion**

In selecting CASE tools for teaching purposes the analysis of the CASE tool domain was an important first step. The list of desirable features and important questions were critical in establishing a match with our goals and requirements. There were many excellent products which are available on other platforms. The vendors literature was not informative enough and often their claims were unsubstantiated. Our decision was largely determined by our existing hardware and operating system environment. Purchasing single copies for a more comprehensive evaluation would eliminate some of these problems.

Exhibit 1 : A rough assessment of the functionality of the CASE products reviewed.

(x means 'yes')

The number across the top of the table correspond to the following products.

1. **ADW**
2. **Corvision**
3. **Deft**
4. **Design/I**
5. **Easy Case**
6. **Exclerator**
7. **Foundry**
8. **IEF**
9. **ISEE**
10. **Lansa**
11. **LBMS Systems Eng.**
12. **Oracle CASE**
13. **PC Prism**
14. **POSE**
15. **PowerCASE**
16. **Reform**
17. **Software Through Pictures**
18. **Synon/2**
19. **System Architect**
20. **System Developer II**
21. **Teamwork**

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	
<b>Life Cycle Coverage</b>																						
Planning	x							x			x	x	x									
Analysis	x	x	x	x	x	x		x	x	x	x	x		x	x	x	x	x	x	x	x	x
Design	x	x	x	x	x	x		x	x	x	x	x		x	x	x	x	x	x	x	x	x
Construction	x	x						x	x	x	x	x		x	x	x	x	x				x
Installation	x	x						x	x	x	x	x			x	x		x				x
Maintenance	x	x						x	x			x										x
<b>Components</b>																						
Diagramming Tools	x	x	x	x	x	x	x	x	x		x	x		x	x		x		x	x	x	x
Pc Repository	x		x	x		x		x			x			x	x				x	x		
Mainframe Repository	x	x		x		x		x		x		x					x	x				x
Design Analyser	x	x	x	x		x		x	x	x	x	x		x	x	x	x	x	x	x	x	x
Code Generator	x	x						x	x	x	x	x			x	x		x				
Expert System Rules	x						x	x														
<b>Diagrams Supported</b>																						
Data Flow	x		x	x	x	x	x	x	x		x	x		x	x		x		x	x	x	x
Decomposition	x		x		x	x		x			x	x		x			x		x	x	x	x
Data Model	x	x	x	x	x	x	x	x	x		x	x		x	x		x		x	x	x	x
Data Navigation		x																x				
Action	x	x						x						x		x						
Matrix processors	x					x		x				x										
Pseudocode											x											
Structure Charts	x			x	x	x	x	x	x					x			x		x			
Dialog flow	x							x														
State transition					x		x		x								x		x			x
Control flow																	x		x			x
<b>Integrated functions</b>																						
Screen/report painting	x	x	x	x		x		x	x	x	x	x		x	x	x	x	x	x	x	x	x
Dictionary definition	x	x	x	x		x		x	x	x	x	x		x	x	x	x	x	x	x	x	x
DBMS	x	x	x					x	x	x	x	x		x	x	x	x	x	x	x	x	x
Prototyping tools	x	x		x		x		x	x		x	x		x	x	x	x	x				
Procedural language	x	x						x		x	x			x	x	x						
Network Capability	x	x				x		x	x	x	x	x		x	x	x	x	x				x
Windows	x	x	x			x		x	x		x	x		x	x	x	x			x	x	
<b>Methodologies</b>																						
Information Engineering	x	x						x														
Yourdon/DeMarco	x	x		x	x	x	x		x								x		x	x	x	x



Chantico	CASE : The Potentials and Pitfalls, QED Information Sciences, Inc, 1989.
Chen,M, Nunamaker,J & Weber,S	Computer-Aided Software Engineering: Present Status and Future Directions, Database Spring 1989.
Chikofsky,E & Rubenstein,B	CASE: Reliability Engineering for Information Systems, IEEE Software, March 1988.
Computerworld	Software Development, Product reviews, March 15, April 19, May 17, June 21, 1991.
Dewey,R	Computer-Aided Software Engineering: Its Configuration, Impact and Directions, 2nd Pan Pacific Computer Conference, Singapore, 1987.
Finkelstein,C	An Introduction to Information Engineering: From Strategic Planning to Information Systems Addison Wesley, 1989.
Gane,C	Computer-Aided Software Engineering: The methodologies, the products and the future Prentice-Hall 1990
Grant,D	Undergraduate Software Engineering - An Innovative Degree at Swinburne, The 6th Australian Software Engineering Conference proceedings, 1991.
McClure,C	CASE is Software Automation, Prentice-Hall, 1989
Mimno,P	Survey of CASE tools, a chapter in Modern Software Engineering : Foundations and Current Perspectives, Ng and Yeh, Van Nostrand Reinhold, N.Y. 1990
Necco,C, Tsai,N & Holgeson,K	Current Usage of Case Software, ACM Journal of Systems Management, May 1989.
Norman,R & Nunamaker,J	CASE Productivity Perceptions of Software Engineering Professionals, Communications of the ACM, Sept 1989.
Parkinson,J	Computer Aided Systems Engineering: an introduction, Software Tools '89, Conference Proceedings, Blenheim Online Ltd, 1989

Rombach,H	Software Specifications: A framework, SEI Curriculum Module SEI-CM-11-2.1, published in Standards, Guidelines, and Examples on System and Software Requirements Engineering, Dorfman, Thayer, IEEE Computer Society Press 1990.
Sun Microsystems,Inc.	Integrated CASE and Technical Documentation, Technical White Paper, Sun Microsystems,Inc, February 1991.
Wasserman,A	The Architecture of CASE Environments, Interactive Development Environments Inc. report, 1989.
Wasserman,A	A Graphical Extensible Integrated Environment for Software Development, published in System & Software Requirements Engineering, Thayer&Dorfman, IEEE Computer Society Press, 1990.
Wilson,D N	CASE: Guidelines for Success, Information and Software Technology, Butterworth & Co, September 1989.
Yeh,R	An Alternative Paradigm for Software Evolution, a chapter in Modern Software Engineering : Foundations and Current Perspectives, Van Nostrand Reinhold

## **Detailed Review of**

### **CASE Tools**

Geoff Beckworth  
Department of Computing and Mathematics  
Deakin University  
Geelong 3217  
Australia

Fax (052) 472028  
Phone (052) 471393  
email gbeck@deakin.OZ.AU

The following CASE tools were reviewed (alphabetical order).

**Application Development Workbench**  
**Corvision**  
**Deft**  
**Design/1**  
**EasyCASE**  
**Excelerator**  
**Information Engineering Facility**  
**ISEE**  
**Lansa**  
**LBMS Systems Engineer**  
**Oracle CASE**  
**Power CASE**  
**POSE**  
**Reform**  
**Software Through Pictures**  
**Sylva Foundry**  
**Sylva System Developer II**  
**Synon/2**  
**System Architect**  
**Teamwork**

## APPLICATION DEVELOPMENT WORKBENCH (ADW)

ADW is the OS/2 version of IEW from Knowledgeware. IEW had a workstation repository which could be consolidated into a mainframe repository. ADW requires a PS/2 model 70 or 80 and a minimum of 8 Mb of RAM and 115 Mb of hard disk space. ADW is divided into 4 main components, planning, analysis, design and construction.

**Graphics** Planning diagramming tools include an association matrix, property matrix, decomposition and entity relationship.

The analysis workstation uses tightly integrated diagrams including decomposition, data flow, entity relationship, minispec action. A relational translator converts the data model into a relational schema from which SQL DDL can be generated.

The design workstation include diagramming tools such as a structure charts, module action which supports many languages through templates, screen and report layout, data structure, hierarchical, relational and VSAM files.

**Repository** ADW uses an encyclopedia which is a rule based system that ensures completeness, consistency, integrity of information and real time error checking. As you update the encyclopedia the knowledge co-ordinator automatically updates all diagrams effected by the changes. The encyclopedia objects can be transfered bi-directionally between the encyclopedia and Repository Manager/MVS. With the multiple windowing and expanded memory capabilities of OS/2 you can open workstation tools for any or all phases of application development at once. Multiple versions of the encyclopedia can be opened simultaneously.

**Code Generation** The construction workstation generates both online and batch applications in ANSI standard COBOL or COBOL II for CICS, IMS/DC, DB2, DLI, IDMS and VSAM from the action diagrams. Re-engineering is possible using the Cross System Product/Application Development Facility.

**Cost** \$500 per module (educational price)

**Contact** Ernst & Young CASE Technology Pty Limited (03) 670 3674  
422 Collins St, (03) 642 1100

## **CORVISION**

CorVision is an application development environment for DEC VAX users. Developers can use PC's or terminals as work stations for design and maintenance.

Graphics	The picture programming tools includes an entity diagrammer, menu diagrammer, screen painter, report painter and an action diagrammer.
Repository	A central repository on the VAX stores all design information and allows shared access by development team members. An Intelligent Guidance System automates project management, detects errors in real-time and makes logical next-step recommendations throughout the development process.
Code Generator	An application generator produces applications executable under VMS and it is fully compatible with RMS and Rdb. Applications can also call subroutines written in any native VAX 3GL. Prototyping is available before code generation and a query facility allows end users to query data from generated applications. Full system documentation is automatically produced from the specifications and updated when then specifications are altered.
Contact	Corvision (03) 823 6271

## **DEFT**

Deft is a tool for the Macintosh. It has two main components, Deft Analyst and Deft Designer.

Graphics	Deft Analyst includes a dataflow diagram editor(Gane/Sarson, Yourdon), an entity relationship diagram editor(Chen/Bachman,Martin,IRM, a forms editor and a design analyser. The Designer includes a program structure diagram editor (Jackson) as well as support for INGRES, ORACLE, Rdb, SYBASE, INFORMIX and DB2.
Repository	A multi-user server-based repository is also available. Deft gateways will generate schemas for the RDBMS's mentioned above and will also support reverse engineering from existing database designs.
Contact	Sequel, 276 Heidelberg Rd, Fairfield, (03) 489 3222

## **DESIGN/1**

Design/1 is a component of Foundation which is the CASE tool marketed by Arthur Anderson. Foundation is actually made up of Plan/1 which is an integrated toolset that

supports the creation of strategic information plans, Method/1 which is the life cycle methodology and Design/1 which has the tools for design and analysis.

There are versions of Design/1 which run on DOS and OS/2. The DOS requirements are AT, 640Kb memory and 10mB disk space. The OS/2 requirements are 80836 processor with 8mB memory and 10mB disk space. The following features refer to the DOS version.

Graphics                      Design/1 has a data model editor, data flow diagram editor, general document editor, screen/report design facility, prototyping facility, structure chart editor, reporting facility.

Plan/1 has a data model editor, decomposition diagram editor, matrix editor and an application model.

Contact                      Anderson Consulting,  
20 Flinders Lane, (03) 659 5533

## **EasyCASE**

EasyCASE provides the user with an introduction to the concepts of structured analysis and design methods. From this point of view it is suitable for first year students and given its price, it is affordable for a laboratory environment. It runs on a PC AT with VGA and mouse.

Graphics                      EasyCASE has limited functions but will allow the generation of data flow diagrams, transformation graphs, state transition graphs, structure charts, data structure diagrams, and entity-relationship diagrams. They are produced in accordance with Yourdon/De Marco, Gane/Sarson, Ward/Mellor, Hatley/Pirbhai, Yourdon/Constantine, Jackson and Chen notations. Objects can be drawn on the charts and information about the objects and the relationships with other objects is maintained. If objects are modified then associated objects are also modified.

Repository                    It provides chart linking, level balancing or methodology rule checking.

Other                          There is good printing support to laser printers and plotters.

Cost                            EasyCASE Plus 3.0                      \$595

Contact                      Business Planning Associates. PO Box 4693, Kingston, ACT 2604  
Phone 06 281 4987

## EXCELERATOR 1.9

This is a PC based product which has a substantial user base in tertiary institutions throughout the world. Version 2.0 will be available in 1991 which runs on OS/2.

Graphics	Excelerator allows you to model the systems processes using data flow diagrams (Gane/Sarson and Yourdon/De Marco), structure chart and diagrams and state transition diagrams (Ward/Mellor) for real time systems. The systems data is modelled using an entity relationship diagrams (Chen/Merise) and data model diagrams. A presentation graph allows you to develop an overview of the system for end users or managers and a work breakdown structure diagram allows you to model the systems development graphically.
Repository	The diagrams you create do not interact with the repository (called XLDictionary). You need to make entries manually. In version 2.0 of Excelerator the repository is an encyclopedia where the data dictionary is interactive with the diagrams.
Analysis	Analysis reports are comprehensive. They include record content analysis which looks at potentially redundant data, key validation analysis which looks at completeness and consistency of key data, data model validation analysis which compare the data model diagrams with the relationships in the repository. Data normalisation analyses records to be normalised, a screen/report data analysis is available and record conflicts can be determined.
Screens/Reports	Screen and report layouts can be designed and limited prototyping can be performed.
Code Generation	Data structure definitions can be exported for use in COBOL, BASIC, C and PL/1 programs and screen designs can be exported as data maps. There is an interface to Telon or APS.
Multi-user	Projects can be established with different levels of access for team members and an import/export facility then allows work to be completed independently and later combined.
Third Party	There are a range of other products available. These include PC Prism which allows you to perform strategic information systems planning and enterprise modelling. XL/Interface for Micro Focus Workbench allows COBOL code skeletons to be developed from structure charts and screen designs. Design recovery performs reverse engineering and a code generation module will be available in the future.
Cost	Excelerator/IS Educational Grant Scheme \$50

Contact Intersolv (was Index Technology), (03) 6541744

### **INFORMATION ENGINEERING FACILITY (IEF)**

IEF is based on the Information Engineering method and is a truly integrated tool (I-CASE). It covers the entire life cycle from strategic planning to monitoring of systems in production. The planning, analysis and design toolset's operate on IBM compatible workstations while the construction toolset including code generation, data base generation and diagram testing resides on IBM mainframe.

Graphics	The toolset for specification and analysis includes a matrix processor which allows cluster and affinity analysis. The comprehensive set of diagramming tools include data modelling, activity hierarchy, activity dependency (data flow), organisation hierarchy, action diagramming, structure charting, action block usage, screen design, dialogue flow, data structure and prototyping.
Repository	The toolset's are linked to a central encyclopedia which stores all business and systems information created by the workstation toolset's as models. Models are verified for consistency and completeness at each stage in the methodology.
Analysis	Comprehensive reporting functions are available including model history tracking which provides a record of activities that change a models contents.
Strengths	Model subsetting allows multiple developers to work simultaneously with models being able to be merged or versioned. A public interface allows importing and exporting to third party products.
Code Generation	Tools for development automation include conceptual to physical database design, business process to computer procedure, code (Cobol II) and database generation (DB2), consistency checking, interactive testing of action diagrams.

Contact Texas Instruments (03) 696 1211

### **ISEE**

This is an integrated CASE tool (I-CASE) for business systems which runs on a multitude of hardware platforms including DEC VAX/VMS, VAX/Ultix, HP, IBM RS 6000, Prime, Pyramid, Sequent, Seimens, Sun and Unisys. It can be used in a heterogeneous environment with concurrent access being managed. Running under X-windows or OSF/Motif provides

the user with a sophisticated development environment. The first layer of the software provides access to configuration control, workbenches and other utilities.

Graphics	Analyst supports data flow diagram (Yourdon/De Marco), entity relationship diagrams(Chen), data structure diagrams (Jackson) and other utilities. Architect also supports state transition diagrams (Yourdon). Designer supports structure charts (Constantine)
Repository	Data is stored in a central repository which is an RDBMS and can be accessed with SQL.
Code Generation	The 4GL code is generated from the structure charts which are integrated with RDBMS development tools. A programmer workbench allows the application to be implemented and tested.
Documentation	Diagrams, text, tables and relationships can be imported into Framemaker.
Management	The manager workbench offers work breakdown, estimation, scheduling, monitoring and reporting based on the analysis and design.
Contact	Integrated Systems Engineering PTY LTD, Level 3, 432 Kent St, Sydney, (02) 2615300

## **LANSAS**

Lansa is a product which runs on a AS400. It is a sophisticated 4GL providing an interpretive code. There are no diagramming tools.

Repository	It has a data modelling utility, a data dictionary and the 4GL called the Rapid Development and Maintenance Language (RDML). The data dictionary from Excelerator can be imported.
Code Generation	An application template facility provides a fast development environment by using pre-written modules which can easily be modified or generated. To execute Lansas applications, Lansas or an additional run-time module is required.
Strengths	The 4GL is very powerful and applications could be developed very quickly.
Contact	Aspect Computing, 03 8180604

## **LBMS Systems Engineer**

LBMS Systems Engineer uses the LSDM methodology (LBMS Structured Design Methodology) as opposed to Information Engineering. It also supports SSADM and a new method from LBMS. It runs under DOS with Microsoft Windows and under OS/2 with Presentation Manager. Versions for Unix, VAX and MVS are supposed to be available in the near future.

LBMS is an AD/Cycle product which will use IBM's repository and CSP. The interface is CUA compliant. The tool set provides complete integration including Strategic Planner, Repository System, Code Generation and Reverse Engineering.

Graphics	Data modelling and data inventory, data flow diagrams, dialogue design, general forms and reports.
Code Generation	Automatic generation of pseudo code,
Screens/Reports	Screen painting and report design, prototyping combining dialogue designs with painted screens, define processing from data flow diagrams to operations in program modules, DIF files facility.
Future	Additions will include relational data analysis, physical data design and report generator.
Contact	Consulit, 422 Collins St, Melb (03) 642 1100

## **ORACLE - CASE**

This product provides a top down methodology, designer, central repository, and applications generator producing a complete system which can be put into production. The Oracle CASE family consists of CASE\*Method, CASE\*Dictionary and CASE\*Designer and CASE\*Generator.

Method	CASE*Method provides a set of staged techniques to follow the life cycle of the system using a top-down approach combined with bottom-up cross-checking techniques.
Repository	CASE*Dictionary provides a central repository for the development team. Information can be added in the form-based interface or through CASE*Designer. It is a multi-user database that allows the team to concurrently access and edit overlapping information. Oracle RDBMS provides concurrency control. Security and versioning can also be implemented. Documentation including cross-references, impact

analyses and exception reports can be obtained. Controlled sharing of definitions across multiple applications simultaneously is also possible. Multiple versions are possible and versions may also be frozen.

**Code Generation** It automatically creates a normalised database design and this can be tuned in Oracle as well. It generates the SQL statements that create the tables, views, and indexes of the database. The dictionary is accessible on a variety of platforms. With CASE\*Generator a complete system can be produced. No programming is required, however the system can be retrieved in Oracle RDBMS and modifications can be made. A retrofit facility also allows for reverse engineering to occur from existing Oracle applications. CASE\*Generator will create working applications for SQL\*Forms which is the 4GL application development tool. The application can then run on any Oracle supported environment. It creates all of the code for the application.

**Graphics** CASE\*Designer is a multi-windowed, multi-tasking graphical development environment using X-windows and Presentation Manager graphics interfaces. It is directly interfaced to the dictionary and allows all information to be available to the development team. Output can be directed to postscript printers or plotters. The entity relationship diagrammer allows you to create the database design. It supports subtype, bidirectional relationship naming and assisted layout. The function hierarchy diagrammer allows you to model the functional needs of the business and the structure of the application. Vertical, horizontal or hybrid layouts can be obtained. The matrix diagrammer provides cross-referencing on the contents of the dictionary. This allows checking for consistency and completeness and determining the impact of proposed changes.

**Screens/Reports** SQL\*Forms allows you to build screens from the table definitions. It also allows exits to 3GL routines. SQL\*Menu allows you to build menu interfaces and SQL\*ReportWriter allows you to build reports using a screen-based interface.

**Cost** The following pricing is the retail price for 2-8 users on a Sun4.

	\$
Oracle RDBMS	9,180
SQL*Forms	2,754
SQL*Plus	2,295
CASE*Generator	2,754
CASE*Dictionary	4,590
CASE*Designer	18,360
Subtotal	39,933

Maintenance 8,149

TOTAL 48,082

The educational discount is 50% and 10% of list price for maintenance.

The cost is approximately \$22,000

Contact Oracle Systems Australia, 03 6859100

## **PICTURE ORIENTATED SOFTWARE ENGINEERING (POSE)**

The POSE system consists of nine full-functioned design tools that can be used for both data-driven and process-driven system analysis and design using a common data dictionary. A common graphics interface provides a consistent approach to their use. The system will run on an XT,AT or PS/2 with DOS 3.0 or higher and a minimum of 10mB of hard disk space and 640kB RAM.

Graphics POSE-DMD - data model diagrammer. The outputs include data model diagrams, redundancy reports, undefined association reports and attribute cross reference reports.

POSE-DFD - data flow diagrammer used to define process models of business activities. The outputs include the data flow diagrams, data stores and data flow definitions, hierarchy of data flow diagrams, documentation reports and consistency check reports.

POSE-DCD - decomposition diagrammer used to define the functional breakdown of a system. The output include decomposition diagrams and customised documentation reports.

POSE-SCD - structure chart diagrammer used to define the hierarchy of program modules and document their interfaces. The outputs include the structure chart diagrams, module descriptions, consistency reports, module libraries and documentation reports.

POSE-ACD - action chart diagrammer used to capture and document multi-level program logic. The output is the program specifications.

Repository POSE-DMN - data model normaliser. The outputs include un-normalised and normalised E-R attribute lists.

POSE-LDD - logical database designer for transaction analysis and optimisation of data models. The outputs include transaction usage maps, logical load matrices of transactions, analysis reports and access path load statistics.

	POSE-DBA - database aid to create first-cut database schemas from the optimised data models. The outputs include database schemas for target DBMS (DB2, SQL, ADABAS, FOCUS, AS/400 and ORACLE).
Screens/Reports	POSE-SRP - screen report prototyper. The outputs include customised screen displays an reports and the system prototype.
Costs	Not educational prices. Data Model Tool Kit (DMD, DMN, LDD, DBA) \$2,300 Process Model Tool Kit (DFD, SCD, DCD, ACD) \$2,300 SRP \$800 Planning Matrix Diagrammer \$800 Data Model Bridge \$800 Extra DBA's \$800 FlexGen4 (Source Code Generator) \$3,250
Contact	ACS International, Suite 10, The Chancellor 174-180 Pacific Highway, Nth Sydney, 2060.

## **PowerCASE**

PowerCASE is a CASE tool marketed by Cognos that runs on a wide range of operating systems. It uses a method called PowerDesign and integrates with PowerHouse and SQL relational databases.

Graphics	The data and processes are modelled with an extended entity relationship editor and a data flow editor. They are validated through attribute normalisation and entity life history analysis.
Analysis	The logical models are checked for completeness and consistency and various reports are available. The physical analysis includes completeness reports for access maps, file conditions and mini-specs.
Code Generation	The logical models are translated into physical designs using access maps, database design diagrams, menu hierarchies, physical data flows and mini-specs. The mini-specs and menus are converted into PowerHouse and the database design is generated into SQL.
Screens/Reports	These are designed using QUICK and QUIZ.
Repository	PowerHouse Architect provides a customizable dictionary and provides audit trails on dictionary elements. Functional prototypes can also be generated with data entry screens with error checking, look-ups, element clustering, menu systems and documentation.

Contact Cognos Pty Ltd, 62 Barry St, Carlton, Vic, (03) 3476300

## **REFORM**

Reform is an integrated software application development environment that provides all the facilities necessary to design, develop, implement and maintain database independent applications. There is an interface which allows design data from other case tools to be exchanged.

**Analysis** The analysis phase of the system requires a logical data model and a list of processes to be developed. REFORM allows the analyst to describe business rules and objectives, define system boundaries, build and normalise relational data models, log design considerations and decisions. Cross referencing is used to produce optimised database access and to provide impact studies of the effects of data model changes.

**Code Generation** The code for each task is system generated as a product of the design process. The application windows, logic, reports and database access are generated via C and SQL code generators and fully compiled. An interface layer is established between each program and the target relational database.

**Documentation** All details of the design and development process is captured and collated for documentation. The documentation describes the objectives, operations, windows, reports, help screens and system messages for each process in the system. It also extracts relevant comments from user written source code.

Changes to a system can be made after evaluation of an impact study. The application is regenerated and recompiled. All alterations and additions are subject to version control allowing a history of changes to be stored.

Applications developed using REFORM are independent of hardware, terminal and database platform. REFORM developed applications will operate on all major relational database systems as well as C-ISAM and Cobol-ISAM file systems.

**Cost** Educational prices : 1 user - \$12,500 5 users - \$42,000

**Contact** Software Associates Pty Ltd, 02 959 5887

## SOFTWARE THROUGH PICTURES

Software Through Pictures is a Unix based product which has a lot of potential for use in undergraduate and postgraduate areas.

Graphics                      It is a graphical front end with a WIMP interface using an open architecture environment called Visible Connections which allows the user to extend and customise their environment.

It supports the following methodologies :

- Structured systems analysis (de Marco/Yourdon and Gane/Sarson)
- Real time requirements specifications (Hatley/Pirbhai)
- Hierarchical Data structures (Jackson)
- Entity-relationship modelling (Chen)
- Structured design (Constantine/Yourdon)
- User Software Engineering (Wassermann)
- Object Oriented Modelling

Repository                      An object orientated data dictionary connects all tools and provides multi user access.

Analysis                        It has a document preparation system which integrates with Framemaker and is template driven. It supports 2167A in its templates.

Screen/Reports                It has a 4GL for prototyping. It uses templates for code generation into C, Pascal, COBOL, or Ada.

Third Party                    There are several third party products which link with STP providing a very sophisticated set of tools.

(a) There is an interface to Ingres which provides an Ingres database definition.

(b) KeyOne also supports the design, programming and documentation phases of products using Ada, C, C++, Fortran, Cobol, Pascal. Reverse engineering will provide hierarchical structures for STP.

(c) SES Workbench allows users to evaluate system design decisions throughout the development cycle.

(d) Uniface is a 4GL application development and run time system which has a bridge to STP.

Cost	5 user	\$15,000
	Site licence	\$25,000

Documentation x 1           \$350  
Installation               \$1000  
Maintenance 15% of licence price.

Contact                    Prophecy Technologies, Glen Waverley (03) 550 1863

## **SYLVA FOUNDRY**

Sylva Foundry allows you to create your own customised systems engineering environment. This would be very useful for advanced students.

You can develop new systems engineering techniques or customise existing structured techniques to meet specific design requirements and also design and implement interactive menus and control screen that guide users through the different processes in the environment.

This tool can be integrated with other tools through the use of triggers to enable you to integrate external programs with the standard Editor functions.

Users can create interfaces to RDBMS, 3GL and 4GL generator tools and expert systems and a reverse engineering facility is available for creating diagrams from external data.

A frame diagram library exists for De Marco Data Flow, Gane and Sarson Data Flow, Ward-Mellor Control flow, Chen entity relationship, state transition, and Constantine structure.

## **SYLVA SYSTEM DEVELOPER II**

Sylva System Developer II uses a rule based approach which ensures that teams work within methods, procedures and techniques.

Graphics                 The standard techniques are DeMarco Data Flow, Gane and Sarson Data Flow, Ward Mellor Control Flow, Entity Relationship, State Transition, Constantine Structure. The editor allows you to interact with the diagrammatic models and access external programs. It provides access to text, graphic text, specification and pseudo-code editors. Screens and dialogues can be defined, prototypes can be run and code generated.

Repository               The team level data dictionary catalogues information supplied through the editor and the automated techniques. Information can be captured automatically from diagrams or from the text editors. Dictionary entries in use are locked at record level and users can select when the

dictionary is updated. Interactive Access is an applications interface allows you to integrate other applications with the editor. It provides the means of sharing diagram information with external systems. This can be used to create interfaces with rdbms, generator tools and expert systems.

**Analysis** A balance function lets you balance designs from any level of the project. Flows are analysed from level to level and reports are available on compositional flows, internal mismatches, missing sub-level flows, multiple representations, incorrect representations and extra inputs. N-ary relations and inheritance are supported for defining integrity relations in object orientated techniques.

**Screens/Reports** Dialogue objects can be created to allow you to perform prototyping. This can be performed independent of System Developer. A Dialogue Definition Facility is used to enter commands that control the display and behaviour of the panels you define including conditional statements and variables. Dialogues can call and execute other dialogues, so complex prototypes can be created. The panels and dialogues can be used to create screen handling logic in COBOL for MS-DOS or Unix. This generates Procedure division display and accept logic and working storage source code. A test program is also generated so that it can be run in these other environments.

**Documentation** A document formatter allows you to collate and combine the design and analysis materials into high quality documents.

**Other features** The Foundry Crossover kit based on the Sylva Foundry package which provides a customizable environment. Rule tool enables you to create custom engineering techniques or modify the provided techniques. Screen maker allows you to define the screens, panels, menus and actions that guide and control the customized method. Diagram generator functions as a reverse engineering tool that uses external data files to automatically generate diagrams.

**Cost** System Developer II \$1,648.00  
Foundry \$13,000.00

**Contact** Prophecy Technologies, (03) 550 1863

## **SYNON/2**

This is an IBM mid range lower case tool/application generator. It runs on an IBM AS/400. Entity/relationship modelling is used to design the data model.

Graphics	It will also import Data Dictionary entries from Excelerator so that Excelerator could be used in the analysis and design phase and then Synon used to produce the application.
Repository	The data model and data dictionary are fully integrated.
Analysis	The environment is text based and integrity checking is performed to ensure correctness of the model.
Screens/Reports	Screens and reports are established. Sophisticated prototyping can be achieved during the design phase.
Code Generation	Action diagrams (James Martin) provide a procedural language. An application generator generates native RPG or COBOL code.
Strengths	Synon/2 enforces a structured top down methodology ensuring consistency in application development. It produces compiled code so applications have high performance. A graphic design tool similar to Excelerator will be available next year. Synon/2 developed applications can execute without the presence of Synon/2 or a run-time module.
Contact	Synon Pty Limited, 03 8091933

## **SYSTEM ARCHITECT**

System Architect is a design tool that operates under MS-DOS within the Microsoft Windows environment. It allows the design of transaction processing and real time systems.

Graphics                      Data flow - DeMarco/Yourdon, Gane/Sarson, Ward/Mellor for (real time). Structure Charts, Entity Relation diagrams, Decomposition diagrams, State Transition diagrams, Flow Charts, Object Orientated design - Booch.

An interesting feature is that you can build a hierarchy of diagrams, each diagram linked to a specific symbol on the level above it, and each level consisting of a different type of diagram.

For example, an entity relationship diagram may be linked to a data flow diagram, and a process in a data flow diagram may be defined by a system flow chart.

ER diagrams support additional views, which include the display of keys, non-keys and volume statistics. Normalization to third normal form can be performed. Super type and sub type relationships can be defined.

The graphic user interface allows reduction, zoom and page previews as well as cut and paste options.

Analysis	Dataflow diagram rule checking and balancing is available as well as on-line error and reporting capabilities for all of the common rules associated with DFD's. Level to level balancing, data store balancing and split/join balancing is available. On-line checking and reporting is provided for Structure Chart rules. The specifications and tracking of system requirements, test plans and change requests are available as well as traceability reports identifying specifications not met in the design. Matrix reporting is also included. There is a customizable query/reporting system and documentation facility which allow the user to specify a series of diagrams and reports as part of a document.
Repository	The data dictionary is stored in DBase 3+ format and the user has direct access to all the information stored in the system. All data can be imported and exported to other CASE tools via CSV file format. Each dictionary item can have attributes and every diagram, graphic symbol and dictionary item has a separate entry in the database.
Networking	Complete record and diagram locking is available. Each user must have a copy of system Architect and a key disk. It is currently running on 3Com, Novell, Starlan, Token Ring and DECnet networks. The encyclopedia can be on a server or on individual hard disks and merged periodically.
Future	OS/2, AIX versions, screen/report design painting, rapid prototyping, code generation, SQL server interface, schema generation.
Contact	Prologic Pty Ltd, 75 Federal St, North Hobart, Tasmania 7000. Phone (002) 346499. Fax (002) 342719.

## **TEAMWORK**

Teamwork is a UNIX based product which is marketed by Cadre Technologies who are based in Canberra. The Australian distributor is EXECP who are in Toorak. Teamwork is a set of tools which perform various aspects of the analysis and design cycle.

Graphics	Teamwork/SA is the structured analysis module using the Yourdon/De Marco methodology. It builds, stores and modifies the DFD's with automatic generation of process specifications for textual detail and data dictionary entries.
----------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Teamwork/SD is the structured design module based on the Constantine method for the development of structure charts.

Teamwork/IM is the information modelling module which allows you to model the entities, relationships and attributes of all application data on a conceptual level. The graphic editor uses Chen's notation and understands e-r diagram modelling rules. The data definitions associated with the entities and relationships can be automatically generated with user-configurable physical and logical attributes. Consistency and completeness checking is performed as well as normalisation and cardinality. Schema generation is possible through Teamwork/ACCESS and are compatible with all major rdbms's.

Teamwork/RT provides an environment for real-time modelling by providing extensions to SA with control flow and control specification connectors. A state transition diagram and matrix editor allows you to create decision tables, state transition tables, process activation tables and state event matrices.

Repository It is integrated with the data dictionary and allows for consistency and completeness checking.

Code Generation Teamwork/CSB or C Source Builder provides code generation from the structure charts and embedded source code.

Other Teamwork/IPSE\_Toolkit integrates all of these products and includes user menus, document production interface, ACCESS to allow tools integration and CDIF which allows for the transfer of data between different case tools. Configuration management features such as baselining and the ability to merge models are available. There is also another module for the development and analysis of Ada systems as well as an Ada source builder.

Cost The system can be implemented in a variety of configurations (educational prices).

	\$
1 or 2 user	7,308
5 user	9,135
10 user	10,962
TPE/SA	914
TPE/SD	914
TPE/IM	1,609
TPE/RT	914
2 user maintenance	3,806
5 user maintenance	7,540
Contact	EXECP, 03 828754