ACRU
MODEL DEVELOPMENT AND USER SUPPORT

SD Lynch • GA Kiker

WRC Report No 636/1/01

Water Research Commission

ACRU
MODEL DEVELOPMENT AND USER SUPPORT

SD Lynch and GA Kiker
With
A Pike and RE Schulze

School of Bioresources Engineering and Environmental Hydrology
University of Natal, Pietermaritzburg
South Africa

August 2001

WRC Report No : 636/1/01
ISBN No : 1 86845 654 4

Contributions towards a final report to the

WATER RESEARCH COMMISSION

on Project 636 "Hydrological Systems Modelling Research Programme"

Obtainable from

Water Research Commission
P O Box 824
Pretoria, 0001
South Africa
www.wrc.org.za

School of Bioresources Engineering and Environmental Hydrology
University of Natal, Pietermaritzburg
Private Bag X01, Scottsville, 3209
South Africa
www.nu.ac.za

WATER RESEARCH COMMISSION REPORT, 636/1/00
ACRU REPORT 47, 1999

ISBN 1 - 86845 - 654 - 4

When using any part of this document as a reference, please cite as in the following example:

Lynch, S.D., 1999: *ACRU*2000 Utilities. *In*: Lynch, S.D. and Kiker, G.A. *ACRU Model Development and User Support*. Water Research Commission, Pretoria, South Africa, Report 636/1/00.

TABLE OF CONTENTS

EXECUTIVE SUMMARY

ACRU MODEL DEVELOPMENT AND USER SUPPORT

SD LYNCH and GA KIKER

The overall aim of this project was to further improve the modelling of hydrological processes by restructuring the *ACRU* modelling with regard to model performance, user-oriented utilities and user educational support. This project was divided into three sections, *viz*. the internal development of the *ACRU*2000 model which sought to create a modular and well designed model framework for current and future hydrological modelling innovations, the *ACRU*2000 utilities which provided improvements to the user interface system as well as data processing capabilities and the operations of the *ACRU* User Consultant who provided both products and expert services to model users.

Over its history, the *ACRU* model has been refined and changed so that it can better serve the needs of the hydrological community. As subsequent changes and alterations began to limit the capability of the current *ACRU* framework, the *ACRU*2000 modelling system was developed to be a modular and expansion-friendly system that allows both current advances in spatial modelling as well as upcoming advances in catchment management. This innovative approach uses object oriented programming design and code development to manage the various components that occur in hydrological systems.

In the past five years, innovations in graphically-based, user-friendly programs have created many new options for integrating user knowledge and existing databases into model-ready inputs. The *ACRU*2000 utilities were developed to integrate geographic information system databases and user-derived information into a flexible menu system for the *ACRU* model. A program was developed to create an initial input dataset to the *ACRU* model and this is a unique feature, as far as computer simulation models are concerned, in that the user is able to get results, within minutes, from the *ACRU* model by only supplying the catchment layout in a digital form. In addition, the revised utilities host a large array of powerful input/output tools to allow users to easily manipulate hydrological data into a variety of formats.

The *ACRU* User Consultant has continued to provide support to a variety of researchers, industry consultants and stakeholder groups. *ACRU* user services have included various general and specialised *ACRU* courses, one-on-one tuition, numerous telephonic and Email discussions, participation in collaborative projects and assisting in real-world applications of the model.

CHAPTER 1

INTRODUCTION TO THE ACRU RESTRUCTURING PROJECT

SD LYNCH and GA KIKER

1.1 PROBLEM STATEMENT

Since its origins as a catchment model applied in the Natal Drakensberg of KwaZulu-Natal (Schulze 1975), the *ACRU* modelling system has been enhanced to address many water-related challenges in South Africa and beyond. As a result of the many researchers, graduate students and consultants interacting with the model code and making various improvements, each consecutive addition has created a more difficult design and coding challenge for subsequent researchers. With the many contributions to the model over the past fifteen years, the result has been a code framework within which it is relatively difficult to make new additions to the *ACRU* model, and in some instances the model structure has been unable to accommodate new modules at all. The *ACRU* program code was developed in the FORTRAN 77 programming language which has some advantages in computational efficiency, but also has disadvantages in developing a modular, easily expandable program design.

As social and governmental interest in water-related issues increases, new capabilities and tools are being requested by various stakeholder groups to allow them to manage hydrological information more effectively. These requests can be grouped into two general categories, *viz.* requests for improved input and output tools to aid in the interpretation of hydrological information and requests for improved model performance to simulate hydrological features of interest. This Water Research Commission project was focussed on these two categories.

1.2 PROJECT OBJECTIVES

The *ACRU* restructuring project is a component of the Hydrological Systems Modelling Research Programme in the School of Bioresources Engineering and Environmental Hydrology (BEEH), the overall aim of which was to

- further improve the understanding of hydrological processes
- and the modelling thereof
- using the existing *ACRU* modelling system
- in a restructured approach

in order to

- render the modelling system user-oriented

and furthermore to

- develop, revise and refine basic hydrological model input
- at a national scale and going beyond South Africa's borders
- using state-of-the-art techniques

for the

- enhancement of integrated water resources related assessment, decision-making and management.

The specific aims of this project component included the following:

- The *ACRU* modelling system was to be *restructured* to render it more modular, to include processing the hydrology of subcatchments in parallel so that it could also operate on a near *real-time basis. This restructuring would facilitate easier future* model enhancement, as it makes additions of newly developed routines (e.g. irrigation scheduling etc. and their associated Decision Support Systems (DSS)), changes to existing routines and inter-linkages with spatial and temporal data/information bases (including daily ones for hydrological forecasting) easier and less error prone. This objective was seen as fundamental for this and for other *ACRU* projects because it would make the model more user-friendly and model development, which takes place by a *team of collaborators* (from within BEEH, other Universities and overseas institutes), more manageable.
- The second fundamental aim of this project was to provide an excellent *user support* for the many users and potential users of the *ACRU* modelling system and the other hydrological products developed over the years by this School through Water Research Commission (WRC) funding and to *promote further user acceptance* of these products.

These objectives are further detailed in terms of the *ACRU*2000 utilities and the internal *ACRU*2000 model design within the following sections.

1.2.1 OBJECTIVE OF THE *ACRU*2000 UTILITIES SECTION

The research proposal drawn up late in 1993 states that the overall aim, as far as the utilities are concerned, of the Hydrological Systems Modelling Research Programme is to *inter alia*;

- *render the modelling system user-orientated*
- *develop, revise and refine basic hydrological model input at a national scale and going beyond South Africa's borders*
- *use state of the art techniques.*

1.2.1.1 MODEL INPUT REQUIREMENTS

The objective, *to develop, revise and refine basic hydrological model input*, includes producing a transparent interface to data that are housed in a variety of data formats in a variety of databases at a variety of different locations. Achievement of the objective will also allow the *ACRU* model to be able to access data generated, or that is required, by other simulation models without user intervention.

1.2.1.2 MODEL OUTPUT REQUIREMENTS

Historically the output from the *ACRU* model was printed to a file that was difficult to import into another computer program, but was laid out in such a manner that the hydrological-oriented user was at home with it. The objectives of the project thus include enabling the *ACRU*2000 model to present its results in such a manner that utilities from other organisations can be incorporated to assist the user to better interpret the simulation results and *to render the modelling system user-orientated.*

1.2.1.3 THE GRAPHICAL USER INTERFACE FEATURES

To render the modelling system user-orientated, the primary feature would be to include a Graphical User Interface (GUI) and GUI-enabled operating systems like Microsoft Windows 95+ which allow users to click on an icon to launch a program or to select and input data by simply using the computer mouse. It is a well-known fact that the process of making a computer application "user-friendly" does not add any scientific value to it. It does, however, allow the modeller the opportunity to select, edit and accept model input and output in a user pleasing manner. Modellers often execute a utility or a suite of utilities a vast number of times when performing a computer simulation and GUI-enabled utilities definitely make this process less time consuming and also less frustrating.

1.2.2 OBJECTIVES OF THE INTERNAL *ACRU*2000 DESIGN

The fundamental objectives of the "internal" design of *ACRU*2000 are to provide *an improved hydrological model* that strives for

- modular internal structure, and
- process representation.

1.2.2.1 MODEL EXECUTION REQUIREMENTS

The objectives of internally restructuring the model are:

- to enable parallel processing of subcatchments within the model
- to enable (near) real time processing
- to allow variable time steps and/or data driven time steps to operate
- to develop flexible code and data structures to accommodate future model additions
- to enable flexible operating rules for catchments, reaches and dams to be incorporated
- to enable flexible spatial configurations to be made and
- to accommodate flexible linkages to other models.

1.2.2.2 OBJECT ORIENTATION

In order to achieve the above objectives, the *ACRU* restructuring team utilised the object oriented design framework known as the Unified Modelling Language (UML). The UML design process was facilitated by the use of the *Rational Rose* visual modelling tool (Rational Software Corp., 1995) which automates software design by allowing diagrams to be accessed and updated in an orderly graphical format. In addition, the *Rational Rose* software has code generation properties in the Java computer language which aid in the transition from design phases to working source code.

CHAPTER 2

REVIEW OF RELEVANT SOFTWARE DEVELOPMENT

SD LYNCH and GA KIKER

2.1 *ACRU*2000 UTILITIES

When this project started the first "windows" operating system that was investigated was IBM OS/2 Warp in August 1995. A few months later the research team became one of the approximately 400,000 worldwide Beta testers of Microsoft Windows95. This operating system was found to be more suitable than OS/2 Warp. With this preamble the complexity of deciding on an operating system, a computer programming language (Fortran90 only became available in 1994) and a choice of commercial software should be recognised. At the time of writing in 1999 Microsoft Windows98 was already widely used, ArcView3.2 was about to be released in the USA and a 350MHz Pentium II processor accompanied by 128 MB of RAM was a common tool. This represents a quantum leap from what was available and was thought would be available when formulating this project in 1993.

2.1.1 DEVELOPMENTS IN GEOGRAPHICAL INFORMATION SYSTEMS

When this project started ArcView2 had just replaced ArcView1 and Microsoft Windows95 was a few months from being launched and was still code-named *Chicago*. This was a major disadvantage, as far as BEEH was concerned, as ArcView2 is a 16-bit application and could only access point, line and polygon coverages. The greater part of the BEEH data are stored in raster format and the research team had to depend on the GRID module in the Unix version of ARC/INFO (ESRI, 1997a) on the Computing Centre for Water Research (CCWR) mainframe computer to process the raster data.

The ArcView system

ArcView is a powerful, easy-to-use tool that brings geographic information to the desktop environment and gives the user the power to visualise, explore, query and analyse data spatially. ArcView is developed by Environmental Systems Research Institute (ESRI), the makers of ARC/INFO, which have been facilitating the solution of spatial problems with computers for more than 20 years. The version of ArcView at the time of producing this report in 1999 (ESRI, 1998a) can be used to access vector coverages, map libraries, GRIDs, images and event data. ArcView scripts are written in Avenue, ArcView's programming language and development environment. With Avenue one can customise almost every aspect of ArcView, from adding a new button to run a newly written script, to creating an entire custom application for distribution. An Avenue script can be ported to a host of different hardware platforms without making any changes to the script.

The ArcView Spatial Analyst (ESRI, 1998b) is a tool for helping to discover and understand spatial relationships in data. The main component of the Spatial Analyst is the GRID theme, which is the raster equivalent of the feature theme.

The ArcView Dialog Designer (ESRI, 1997b) is an extension to ArcView GIS that provides Avenue developers with a new tool, a dialog, to customise ArcView's interface. A dialog allows the organisation of a single task, or set of related tasks, onto a separate window, akin to organizing related tasks under a particular menu item or on the button bar.

Well into this project the ArcView3 system, a 32-bit application that can access raster data was launched. The ArcView environment has since become the *deo facto* desktop GIS package in South Africa and is also a very sound cornerstone for the *ACRU*2000 utilities. BEEH has been instrumental in the creation of digital datasets for a host of agrohydrological data for South Africa (Dent *et al.*, 1989; Lynch, 1989; Lynch, 1998; Lynch *et al.*, 1989; Lynch *et al.*, 1994; Lynch *et al.*, 1995; Lynch *et al.*, 1996b; Schulze, 1997) and these data are maintained and used in the ARC/INFO and ArcView environment. Approximately 3000 copies of ArcView3 and roughly 200 copies of Spatial Analyst had been sold in southern Africa at the time of writing in August 1999 (McKivergan, 1999). The Dialog Designer extension is included, free of charge, in the ArcView package.

2.1.2 PLATFORM INDEPENDENT LANGUAGES

The name platform independent implies that the computer program can be run on any hardware platform. In some cases the source code needs to be compiled on that system and in other cases the code does not require re-compilation.

FORTRAN has been used as a programming language by scientists since it was introduced in the 1950s. It has undergone a few major revisions (FORTRAN IV, FORTRAN66, FORTRAN77 and Fortran90) and is still the most widely used scientific programming language in the world. FORTRAN is one of the languages where an executable file is created by compiling the source code on the platform where it is to be implemented.

Avenue, the ArcView scripting language, on the other hand does not require re-compilation when porting it to a different hardware platform for which the ArcView package is available (see 2.1.1).

Java is an Object Orientated Programming language that was introduced by Sun Microsystems in 1995 and is also designed to be implemented on any hardware platform.

Visual BASIC, Delphi and Microsoft Fortran PowerStation (MFPS) PC software can only run on machines that use the Microsoft windows operating system. If the GUIs are removed from a program developed in MFPS, the source code can be re-compiled on any hardware platform for which a Fortran90 compiler is available.

Computer programmers often fall into the trap of trying to make slight modifications in order for their software to become platform independent. This then means that whenever code is changed the necessary changes have to be made and tested on all the hardware platforms that do not comply with the one that they are working on.

2.1.3 GRAPHICAL USER INTERFACE DEVELOPMENT TOOLS

Fortran PowerStation

With the launch of the 32-bit Windows95 operating system in 1995, computer programmers are able to utilise the Graphical User Interface (GUI) to make their computer programs more user-friendly. BEEH has vast resources as far as FORTRAN programming is concerned and this release of "Visual" FORTRAN was seen as an answer to the programmers prayers. The major disadvantage is that the "Visual" part of the code cannot be ported to the Unix environment.

When Microsoft made the abrupt decision to halt the development and support of the MFPS PC software (Microsoft, 1995) and it was decided not to re-write the programs, but to use the executables and where modifications are required, to use this "Visual" FORTRAN to implement the changes.

Tcl/Tk

The Tool Command Language which is a toolkit for the X Windows system (Tcl/Tk) was initially developed by John Ousterhout (undated) as a hobby in his spare time in the late 1980s. Tcl/Tk was investigated at the initial stages of this project and was found to be unsatisfactory as this public-domain software at that time was prone to malfunctions which the Tcl/Tk developer team were only able to address when they had free time available.

Java

The Java platform is a fundamentally new way of computing, based on the power of networks and the idea that the same software should run on many different kinds of computers, consumer gadgets, and other devices. Java is a powerful programming language when it comes to GUI implementation. The speed at which the GUIs are developed and implemented was unacceptably slow for the purposes of developing the *ACRU* utilities. The fact that code for the manipulation of GIS information is not readily available to the developer was reason enough to decide against using Java as the programming language for GUIs.

Delphi

The Delphi programming environment was released in 1995 by Borland and when investigated initially for use as a GUI programming language for this project it was found to be unsuitable as it is designed to run under the Microsoft Windows operating system and initially there was no GIS-Delphi integration.

Visual BASIC

In 1991 Microsoft released Visual BASIC, a GUI programming language that can only run on computers that rely on the Microsoft Windows operating system. The initial investigation into this language did not find favour as it is platform dependent.

Avenue

The initial versions of ArcView did not have the capability of GRID-based modelling that Version 3 has, in the form of the Spatial Analyst module. ArcView scripts are written in Avenue, ArcView's programming language and development environment. With Avenue one can customise almost every aspect of ArcView, from adding a new button to run a newly written script, to creating an entire custom application for distribution.

Conclusion
The ArcView environment using Avenue was accepted to design and implement the *IMenubuilder*2000 program. The fact that Avenue is designed to work with GIS concepts and its ability to be ported across a host of different hardware platforms is sufficient evidence that this is a suitable language for the *IMenubuilder*2000 program.

2.2 *ACRU*2000 MODEL

The main objectives of the *ACRU* hydrological model development was to research the various model design methodologies that emphasise a rigorous and organised design methodology to minimise conflicts in code development from multiple additions to the modelling platform. Some of the primary claims of the Object Oriented Programming methodology was that the developed code was modular and did not conflict with existing software designs. In addition, code modules could be "re-used" so that further model development would occur at an increased rate as a benefit of the modular designs (Rumbaugh *et al.*, 1991). Most current modular systems that simulate hydrological or water flow systems use the FORTRAN77 language, with various methodologies to create a more modular environment. As new computer science developments have taken place, various modelling efforts have employed these new tools in various levels of the modelling system. While some have created a hybridised code between languages such as FORTRAN77 and C++, others have concentrated their effort in making modular utilities while leaving their main model code framework in the older formats.

While each modelling system has had advantages and disadvantages, there was no single "correct" method of creating a strong code foundation. In analysing the strengths and weaknesses of each approach, experience was gained in avoiding potential pitfalls in code design and implementation.

2.2.1 USE OF MODULAR OR OBJECT ORIENTED PROGRAMMING IN HYDROLOGY AND LAND USE MODELLING

***PRMS* Model**
The *PRMS/MMS (Precipitation-Runoff Modelling System / Modular Modelling System)* is a modular-designed, deterministic, distributed-parameter modelling system developed to evaluate the impacts of various combinations of precipitation, climate, and land use on streamflow, sediment yields, and general catchment hydrology (Leavesley *et al.*, 1983). The *PRMS/MMS* modular design provides a flexible framework for continued model-system enhancement and hydrological modelling research and development. In addition, the *PRMS/MMS* model has been used with GIS software to explore different land use scenarios and spatial unit definitions (Flügel, 1996). The model is written in FORTRAN77 with parts of the modelling system also constructed in C++.

Within the *PRMS/MMS* model, a catchment is divided into subunits based on such catchment characteristics as slope, aspect, elevation, vegetation type, soil type, land use, and precipitation distribution. Users can partition areas at two levels. The first divides the catchment into homogeneous response units (HRU) based on the catchment characteristics. Water and energy balances are computed daily for each HRU. The sum of the responses of all HRUs, weighted on a unit-area basis, produces the daily system response and streamflow for an entire catchment.

A second level of partitioning is available for storm hydrograph simulations. The catchment is treated as a series of interconnected flow planes and channel segments. Surface runoff is routed over the flow planes into the channel segments; channel flow is routed through the catchment channel system. An HRU can be considered the equivalent of a flow plane, or it can be delineated into a number of flow planes.

APSIM Model

The Agricultural Production Systems sIMulator (*APSIM*) model is a module-based crop model that can integrate a variety of models derived by different research efforts to facilitate comparison of models or sub-models on a common platform (McCown *et al.*, 1996). This functionality has been achieved via the implementation of a "plug-in-pull-out" approach to *APSIM* design as seen in Fig. 2.1. *APSIM* has been developed in a way that allows the user to configure a model by choosing a set of sub-models from a suite of crop, soil and utility modules. Any logical combination of modules can be simply specified by the user "plugging-in" required modules and "pulling out" any modules no longer required. *APSIM* is written in FORTRAN77 and employs a specific coding methodology so that various derived modules can be used in the model.

HSPF Model

The Hydrological Simulation Program-FORTRAN (*HSPF*) model simulates the hydrological, and associated water quality, processes on pervious and impervious land surfaces and in streams and well-mixed impoundments (Bicknell *et al.*, 1997). The model was developed in the early 1960s as the Stanford Watershed Model. In the 1970s, water quality processes were added. Development of a FORTRAN77 version incorporating several related models using software engineering design and development concepts was funded by the Athens Ga., Research Lab of the Environmental Protection Agency (EPA) in the late 1970s. In the 1980s, preprocessing and postprocessing software, algorithm enhancements, and use of the United States Geological Survey (USGS) Watershed Data Management (WDM) system were developed jointly by the USGS and EPA. The model uses a modular-style, functional programming approach to manage its various sub-modules.

HSPF can simulate one or many pervious or impervious unit areas discharging to one or many river reaches or reservoirs. Frequency-duration analysis can be done for any time series. Any time step from 1 minute to 1 day that divides equally into 1 day can be used. Any period from a few minutes to hundreds of years may be simulated. Generally, *HSPF* is used to assess the effects of land use change, reservoir operations and point or non-point source pollution treatment alternatives.

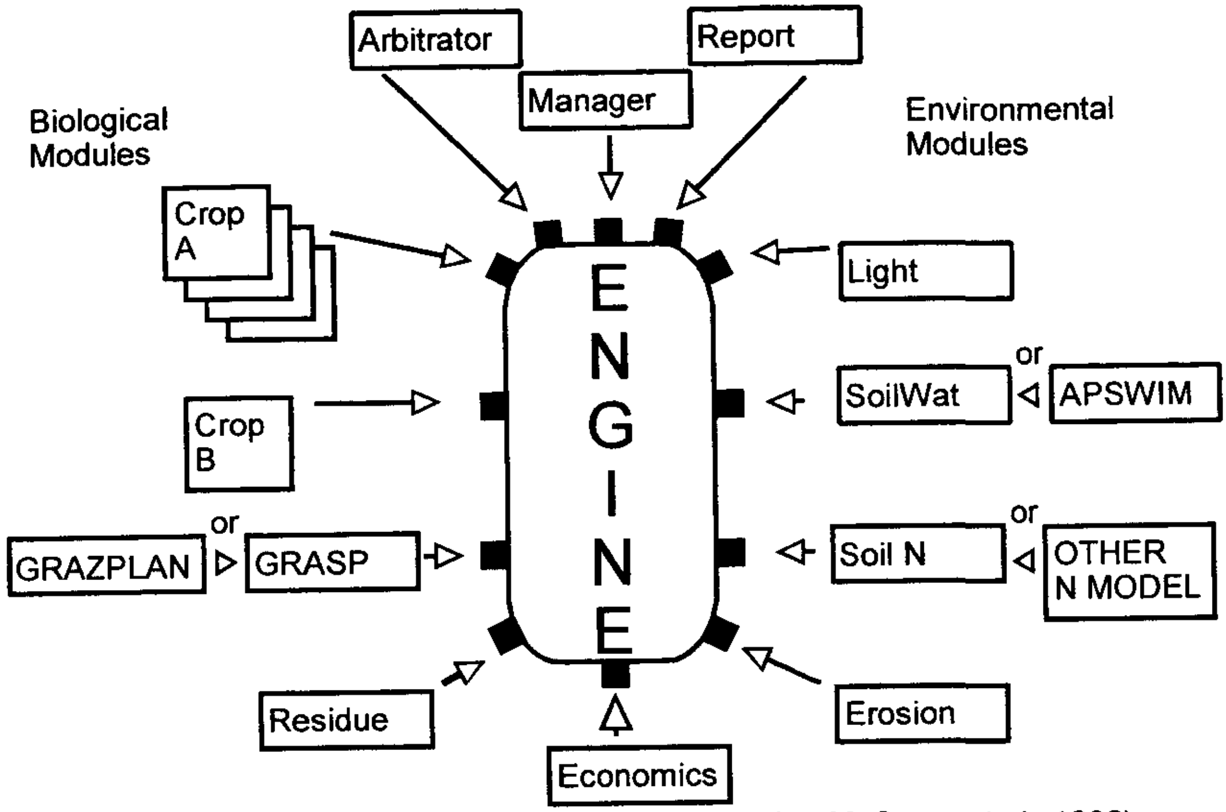


Figure 2.1 General layout out the APSIM model (after McCown *et al.,* 1996)

2.2.2 THE UNIFIED MODELLING LANGUAGE (UML) SYSTEMS FOR DESIGN OF OBJECT ORIENTED PROGRAMS

2.2.2.1 HISTORY OF THE UNIFIED MODELLING LANGUAGE

During the early 1990s a variety of object oriented (OO) methodologies were formulated, each with its particular strengths (Booch, 1991; Graham, 1991; Rumbaugh *et al.*,1991; Booch, 1995). As the various OO methodologies were improved and revised through the mid-1990s, it became evident that common traits were arising in each design system. However, with each system retaining its own symbols and nomenclature, users began to grow frustrated and confused at the lack of coordination between such similar OO design methodologies (Quatrani, 1998).

The Unified Modelling Language (UML) was developed as a solution to this problem of conflicting methods. "UML is a language used to specify, visualise, and document the artifacts of an object oriented system under development. It represents the unification of the Booch, OMT and Objectory notations, as well as the best ideas from a number of other methodologies. By unifying the notation used by these object oriented methods, the Unified Modelling Language provides the basis for a *de facto* standard in the domain of object oriented analysis and design founded on a wide base of experience" (Rational Software Corp., 1995). Since this initial release, a variety of resources have been developed for the UML (Booch *et al.*, 1998; Jacobsen *et al.*, 1998; Rumbaugh *et al.*, 1998).

2.2.2.2 VISUAL MODELLING WITH THE RATIONAL ROSE SOFTWARE SYSTEM

"Visual modelling is a way of thinking about problems using models organised around real-world ideas" (Quatrani, 1998). Models are very helpful in managing the complexity of the real world into functional sections so that one can address specific needs and requirements. But by the same token that reality has complex systems and problems, the demands on models have grown more complex. In constructing complex commercial software systems, software designers regularly use a design methodology to manage the inputs from many programmers working together. In software designed by academic institutions, this is the exception rather than the rule. Usually the software is constructed with the tools (and languages) at hand by scientists with a secondary knowledge of computer science. For small, isolated, *ad hoc* programs, this methodology is effective and efficient. For larger, expansive and installed systems, the methodology becomes problematic as new developments often disrupt the effect functioning and expansion of the system. Visual modelling can provide a "big picture" and design-oriented, view of a software system without having to become ensnared in the explicit details of each specialised component.

Quatrani (1998) describes a triangle of success in Fig. 2.2 which illustrates all of the important parts of a successful software project. Notation, process and tools are equally fundamental in their function in a project. If one learns only notation, one often does not know to effectively use the technique (process). If one only learns the process, one cannot effectively communicate the ideas and designs to others (notation). Also, if one cannot document the specifics of the work (tool), then one is limited in generating follow-on ideas from the initial designs.

As notation was briefly introduced in the UML language section, this section reviews the role of process and tools in object oriented program development. Quatrani (1998) describes an iterative and incremental, development life cycle that may include: requirements capture, analysis, design, implementation and testing. These steps serve to minimise the risks associated with software design and construction by being iterative so that changes in requirements, design or implementation can be treated as a cycle as opposed to a "once-off" activity. The *Rational Rose* software (Rational Software Corp., 1998) has been designed to manage these various activities so that the process progresses smoothly from ideas to designs to code. Changes at the code level are also able to be "reverse engineered" to incorporate code changes quickly back into designs. This efficient process named "Round Trip Engineering" allows fast iteration between ideas, designs and actual code testing.

2.2.3 REVIEW OF THE CURRENT *ACRU* MODEL

The *ACRU* model was developed by Schulze (1984; 1989; 1995) to simulate hydrological and vegetation processes. The *ACRU* model simulates field- as well as catchment-scale processes by either a cell-based network or a lumped parameter model. A primary advantage of *ACRU* is its integration with South African climate data, soil and land use information and GIS (Schulze *et al.*, 1993a). In addition, the *ACRU* model has been widely tested within southern Africa and in the USA and Germany. The primary input to *ACRU* is daily rainfall and monthly temperature, with options of inputting, via a DSS, solar radiation, depending on the internal routines chosen. The time-step is daily, and monthly values can be aggregated. Another important input is soils information, which can be entered from field-test data or estimated by general soil classification.

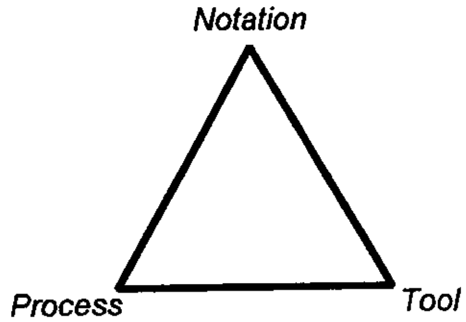


Figure 2.2 The triangle of success in software design and implementation (After Quatrani, 1998)

The *ACRU* water balance is illustrated in Fig. 2.3. Soil water storage is calculated with a two layer soil profile (surface and subsurface layers) with the model redistributing soil water under both saturated (when above drained upper limit) and unsaturated condition (when between its drained upper and the lower limits) according to moisture gradients between the top- and subsoil layers, as well as accounting for capillary action. The fundamental water balance equation on a catchment scale is given as

$$P = ET_{veg} + e_c + \Delta STOR_{surf} + \Delta STOR_{subsurf} + RO + PERC$$

where P is the gross daily precipitation (mm), ET_{veg} is the consumptive use by the soil-vegetation complex (mm), e_c is the evaporation of intercepted water from wet surfaces (mm), $\Delta STOR_{surf}$ is the change in surface, soil layer storage (mm), $\Delta STOR_{subsurf}$ is the change in subsurface soil layer storage (mm), RO is the surface and near-surface runoff (mm) and $PERC$ is deep percolation beyond the subsoil (mm). Most of the water balance components have several options which can be selected for calculation, according to data availability. The precipitation component can use a historical record or a stochastic weather generator for rainfall input. Evaporation from soil surfaces and plant transpiration have a variety of options including ones derived from Ritchie (1972), Linacre (1977) and Penman (1948).

The *ACRU* model was developed by Schulze (1984; 1989; 1995) primarily to simulate hydrological processes within southern Africa. The *ACRU* model can simulate field-scale as well as catchment-scale processes by either a cell-based network or as a lumped parameter model. The procedure for dividing a catchment into subcatchments depends on the spatial variability of precipitation, topography, land use, or soil characteristics (Schulze 1995). Each subcatchment area can range from 0.01 to 50 km^2 and can be linked to other subcatchments as a cascading streamflow system.

The *ACRU* model has been combined with both simple and more complex sub-models to conduct specialised research into water and vegetation issues. Schulze *et al.* (1993b) utilised the *ACRU* model, along with the Rosenzweig (1968) primary production model and a national climate database (Dent *et al.*, 1989) to simulate veld biomass over South Africa with respect to impacts of potential climate change. The *ACRU* model was also linked with the CERES crop model (Jones *et al.*, 1986) to explore potential crop yields over South Africa (Schulze *et al.*, 1993a).

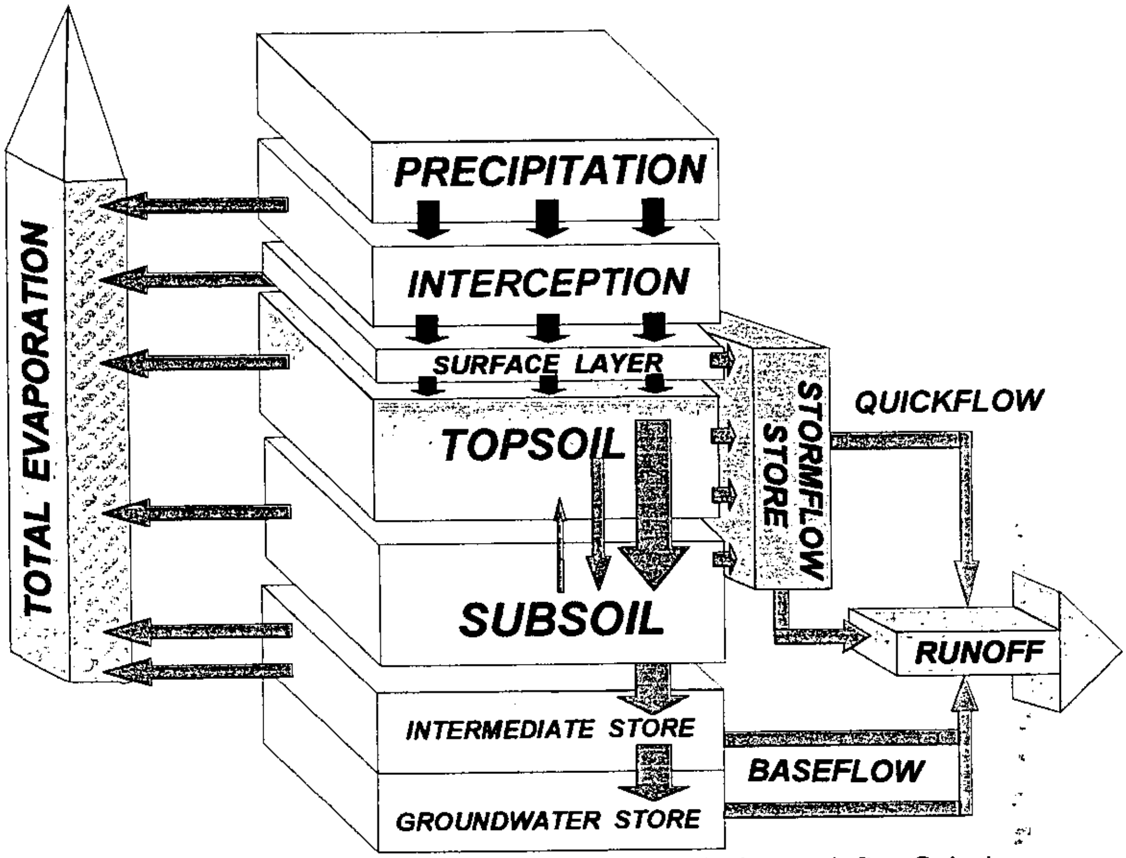


Figure 2.3 The *ACRU* agrohydrological model water balance (after Schulze, 1995)

CHAPTER 3

ACRU2000 UTILITIES

SD LYNCH

The past ten years have seen the developers of hydrological support utilities, the pre- and post-processing routines literally re-invent the wheel numerous times, and BEEH is no exception. Frequently an organisation uses and has developed their own data format, their own plotting routines and their own suite of statistical analysis tools. All have access to commercial software packages like Corel Presentations, Microsoft PowerPoint, Corel Quattro Pro and Microsoft Access to graph and perform statistical analysis. The problem that developers are faced with is that these software packages can only be customised using their propriety macro scripting languages and a "Corel person" does not use the "Microsoft equivalent" and visa versa. These software packages also do not include all the graphing options and specialised statistics that are required to successfully interpret the data.

A decision was therefore taken by the research team to accept that there is no universal data format acceptable to all. Translators have been developed to convert from one data format into the data format used by the relevant software. BEEH has developed utilities to convert their data formats to, *inter alia*, the TSOFT, WDM, flat ASCII and dBASE IV data formats.

The most widely accepted data format is ASCII which can be read by any computer program and can be transferred between hardware platforms without any alterations. Unfortunately anyone can edit and make changes to this file using a standard text editor and insert the letter 'o' instead of the number zero. The binary file types, on the other hand, are usually edited using a problem-specific editor such as a commercial spreadsheet program or a user-developed program, as in the case of the WDM files, and these programs usually have the necessary error-checking code embedded.

3.1 DATA FORMATS

When data are stored in flat ASCII files, many options are available as far as formatting is concerned. The data may be in a "columnar" format or each value per record might be comma or space delimited and of variable length and in some cases there might be no delimitation at all. When developing software to read these ASCII files one often has to hard-code the data type and also the position and field length of each value.

When data are stored in a binary file, using an application or a utility, there is only one way to access the data. In this binary format values are stored without delimiters between the values. For example, one programmer can output a REAL, an INTEGER and a REAL to a binary file and another programmer need only know what the data types are that are being read, irrespective of the number of decimal places of the REAL values.

The binary type files do, however, have some disadvantages which include that:

- they cannot be viewed using a text editor, and
- some of the files require manipulation before being ported between hardware platforms.

The *ACRU* modelling system uses a variety of data formats to allow modellers a host of utilities to analyse and manipulate their data (Fig. 3.1).

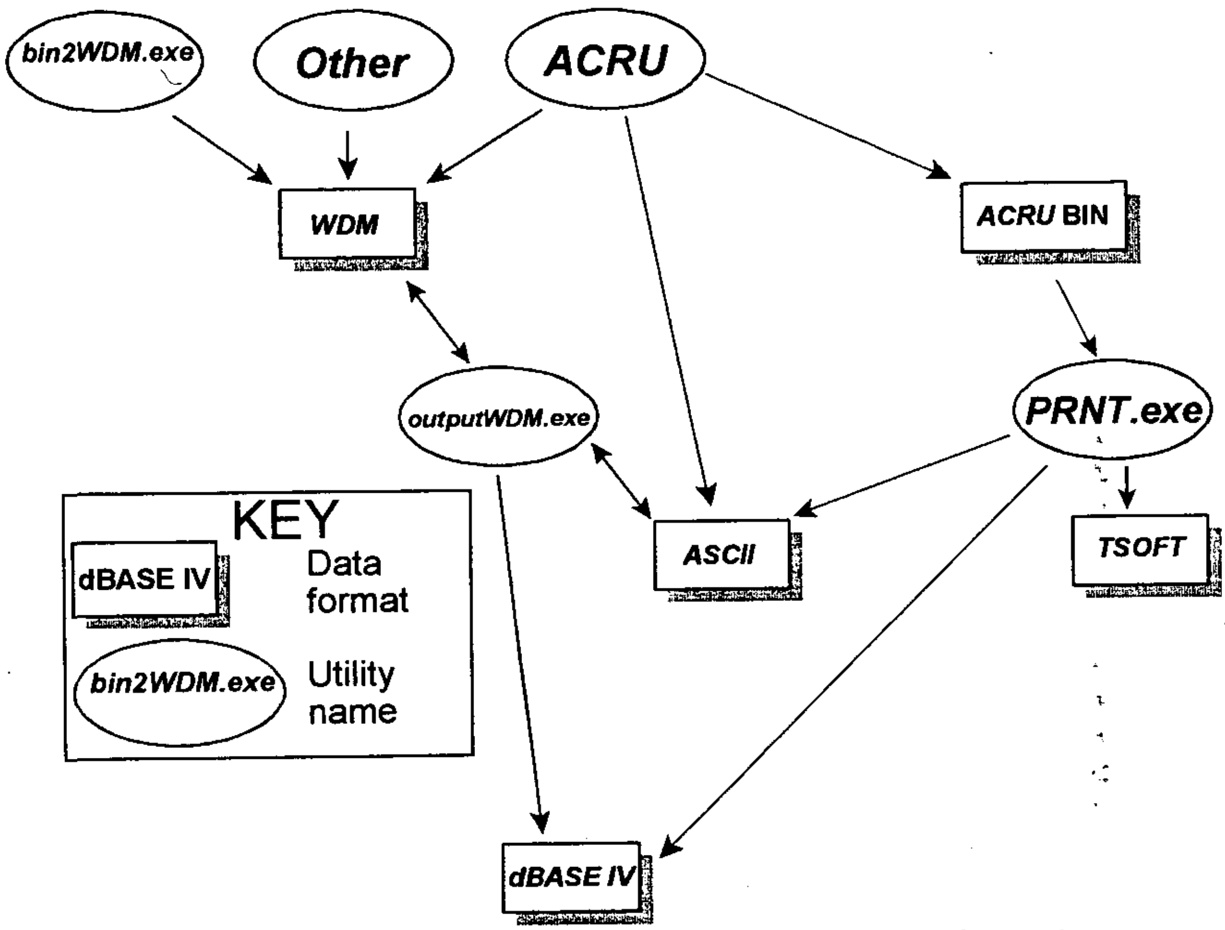


Figure 3.1 Layout of the utilities to convert data between the different data formats that are used in the *ACRU*2000 utilities

ACRU BIN

The current public version of the *ACRU* model (Schulze, 1995) has 67 output variables that can be stored at a daily time step and the values are written to a binary file, *ACRU* BIN. Data that are stored in a binary format greatly decrease the processing time of a computer program and all the data are stored at the full numeric precision. These factors influenced the research team to develop and maintain the *ACRU* BIN file format and the structure is described in detail in the Technical report.

ASCII

An ASCII editor in the wrong hands can be a very dangerous tool. For example, a user can enter the letter 'o' instead of a zero and that can cause major problems. The ASCII file format has been chosen for some of the input and output files that the *ACRU* modelling system requires to accommodate users without access to the ArcView package.

PRNT
This routine reads the *ACRU* BIN file and inserts the output values to an ASCII file depending on the option that the user selected. There is an option to request that the data are written to a file wherein a complete and full description of the simulation is presented. Another option requests a "spreadsheet" type file to be created in ASCII wherein the first record contains the *ACRU* variable names and the following records contain first the date and then the corresponding simulation results. When these daily simulation results are required to be stored at a monthly or annual time step, the conversion process is not a straightforward summation process. Three operations are used to determine monthly values from the daily values, *viz*. sum, average, and maximum as not all hydrological data can be summed to produce a meaningful monthly or annual value.

Menubuilder2
At present there are 125 sections and 243 variables of input data that are required by the *ACRU* model. Some sections contain a single input value and the maximum number of input values in a section is twelve. The *MENU* for a distributed simulation is all contained in a single file.

***IMenubuilder*2000**
The structure of the "new" *MENU* file is very similar to the *Menubuilder2* file format with the main difference being the introduction of a separate control file and that each subcatchment has its own *MENU* file. The control file contains information pertaining to the layout or subcatchment configuration and also contains the file names for the accompanying *MENU* files. The current version of the *ACRU* model reads a *Menubuilder2* type file and the *ACRU*2000 model reads an *IMenubuilder*2000 type file. Utilities do, however, exist to convert between these two formats.

TSOFT
This binary file format was designed and developed by the Institute for Water Research at Rhodes University in Grahamstown (Hughes, 1999) to manipulate and display their time-series data. Software has also been developed to convert the *ACRU* type output data to this TSOFT format.

WATERSHED DATA MANAGEMENT SYSTEM
The WDM system is a mechanism for managing the data which are needed in water resources investigations which include time-series data, tables, text and vectors to be stored in a single storage mechanism. The original design and implementation of the WDM file was a cooperative effort of the USGS and the Soil Conservation Service of the USDA in 1983. Additional types of datasets have been implemented by the USGS and EPA. *ANNIE*, *IOWDM* (Flynn *et al.*, 1995) and *HSPF* (Bicknell *et al.*, 1997) were the original programs that used the WDM file. Since then, over a dozen programs developed by the USGS and EPA have been developed or modified to use the WDM

file. The USGS maintains and distributes the official version of the WDM library. The WDM file is a binary, direct access file for the storage of these data. The WDM toolkit of routines, developed by the USGS, enables expedient input, update and output of stored data. The WDM files allow a comprehensive specification of data attributes, multiple time-steps in single time-series data sets, user-defined formats for data stored as tables, compression of data, improved speed and flexibility of interaction between a model and its database, as well as automatic file maintenance as the user adds, modifies or deletes files. The WDM system allows multiple models, requiring different data types and data with different time-steps, to use a common database.

The WDM system provides a systematic approach to the storage and retrieval of data required to operate hydrological, hydraulic, and water quality models. The WDM system uses a well-defined binary, direct-access file structure accessed through a library of routines to create a file, add data to the file, replace data in the file, get data from the file and delete data from the file. Five categories of data may be stored in WDM file data sets: time-series, tables, text, vectors and space-time. A comprehensive set of attributes is available for documenting and describing individual data sets.

The major limitations of the WDM system include;

- no stand-alone programs in the suite; one needs to be a FORTRAN programmer to access and use the library,
- when porting the WDM file between different hardware platforms the data have to be exported to a WDM-ASCII file and this file needs to be imported into a WDM file on the other system, a process that may require a vast amount of computer disk storage, and
- the USGS is not open to suggestions regarding changes to the structure of the WDM file or to the code in the WDM library (Flynn, 1996).

Datasets stored in the WDM format can be shared between different computer modelling systems without the programmers having to develop new computer code to access the data. The different software developed by the different modelling groups can therefore be shared and this means that no valuable time is lost in "re-inventing the wheel".

Programs using the WDM library have as standard output, *inter alia*, the number of time-steps between two given dates or which dates have concurrent data if two or more sets of time-series data are used. The WDM library consists of more than 650 routines that allow programmers the benefit of using existing software rather than developing and testing their own.

dBASE IV

Commercial spreadsheet applications each have their own unique way of storing tabular or columnar data. The dBASE IV file format is one common format that is shared by all these commercial spreadsheet packages and ArcView also uses this database format to store its attribute information. It is therefore necessary to include support for this format in the *ACRU*2000 utilities.

3.2 MODEL INPUT-RELATED UTILITIES

Computer modellers have in the past spent large amounts of valuable time preparing the input data to the *ACRU* modelling system. These data are often stored in data formats that are not compatible with the data formats supported by the *ACRU* model. The utilities that are described in this section integrate and interface these data and the ever-increasing amount of data that are stored in a GIS to a format that is acceptable by the *ACRU* model. Options have also been included in the interface utilities to re-format these data for integration into other modelling systems.

3.2.1 *IMenubuilder2000*

In the early 1990s it was decided to design an interactive and user-friendly program to assist the user in preparing an input dataset to the *ACRU* model and was named the *Menubuilder*. It was written for MS-DOS using FORTRAN77 and the ANSI.SYS driver. The *Menubuilder* program consists of a main program and 235 subroutines with a total of 21000 lines of computer code and the current version of the *ACRU* model contains a main program and 165 subroutines with 16000 lines of computer code.

The *Menubuilder* is designed to only prompt the user for input information that is required depending on what input has been supplied previously. For example, if PPTCOR = 0, then the user would not be prompted to enter the CORPPT values. A series of error and warning conditions are also incorporated into the *Menubuilder* to prevent the user from entering physically incorrect or impossible information. If, for example, the mean annual precipitation (MAP) value were entered as negative, the user would not be allowed to proceed. If a MAP value of greater than 3500 mm is entered, a warning message would appear to inform the user that the MAP value entered exceeds the maximum for any area within South Africa, but they will be allowed to proceed.

BEEH has been instrumental in the creation of digital datasets for a host of agrohydrological information for South Africa (Dent *et al.*, 1989; Lynch, 1989; Lynch *et al.*, 1989; Lynch *et al.*, 1996b; Schulze, 1997) and this information is maintained and used in the ARC/INFO and ArcView environment. Research into a new component, the *IMenubuilder*2000 (Lynch, 1999), started in 1996 with the main objective being a platform independent GIS-based graphical user interface program that would be able to extract the required information from these digital datasets and insert the values directly into the *ACRU MENU*.

The *ACRU* modelling team of BEEH work mainly on PCs, running Win95/98/NT, that are linked via Ethernet to an IBM RS6000 Unix mainframe computer (CCWR, 1999) used locally and internationally by researchers involved in water research. ARC/INFO and ArcView are the preferred GIS packages that are currently being used in South Africa and therefore form the cornerstone of the *IMenubuilder*2000 program.

Avenue, the programming language incorporated into ArcView, is used in the *IMenubuilder*2000 program as it has full GIS capabilities and the code can be ported between PC ArcView and IBM RS6000 ArcView without any modifications. The associated database files that are in dBASE IV format can also be ported across hardware platforms without any modifications.

Included in the *IMenubuilder*2000 program are DSS that have been developed for South African conditions to assist the user when preparing, *inter alia,* the soil and land cover information that is required by the *ACRU* model. Ten variables are required by the *ACRU* model to represent the soil characteristics of a catchment. If these values are not known then the user is only requested to enter a value for the dominant soil texture class (i.e. clay, sand, loam, loamy sand, sandy loam, silty loam, sandy clay loam, clay loam, silty clay loam, sandy clay or silty clay) and a value for the total depth of the soil layer (i.e. very deep, deep, moderately shallow, shallow, very shallow or impervious), as well as values of the initial soil water content for the two horizons. The DSS will then assemble the ten values that are required by the *ACRU* model.

3.2.2 THE INITIAL *MENU*

The *IMenubuilder*2000 program has been designed to create an initial input *MENU* to the *ACRU* modelling system that will allow the user to automatically create the *MENU* for a given catchment configuration by only supplying the catchment layout in the form of an ARC/INFO polygon coverage or as a shapefile. The required information for South Africa is contained in 49 ARC/INFO GRID coverages that include;

- altitude,
- mean annual precipitation (Dent *et al.*, 1989),
- monthly means of daily maximum temperature,
- monthly means of daily minimum temperature,
- means of monthly totals of pan evaporation (Schulze, 1997),
- fraction of "saturated" soil water to be redistributed daily from the topsoil into the subsoil when the topsoil is above its drained upper limit,
- fraction of "saturated" soil water to be redistributed daily from the subsoil into the intermediate/groundwater store when the subsoil is above its drained upper limit
- soil water content (m/m) at permanent wilting point for the topsoil,
- soil water content (m/m) at permanent wilting point for the subsoil,
- soil water content (m/m) at drained upper limit for the topsoil,
- soil water content (m/m) at drained upper limit for the subsoil,
- soil water content (m/m) at saturation (i.e. porosity) for the topsoil,
- soil water content (m/m) at saturation (i.e. porosity) for the subsoil,
- thickness (m) of the topsoil of the soil profile,
- thickness (m) of the subsoil of the soil profile (Schulze *et al.*, 1990; Schulze *et al.*, 1992), and
- land cover index for South Africa (Acocks, 1988; Schulze *et al.*,1993b).

The relevant information, for each subcatchment, is extracted from the 49 GRID coverages and is stored in a dBASE IV file that the *IMenubuilder*2000 program then uses to create an initial *MENU* for the *ACRU* model.

The flags that require the user to enter information pertaining to additional options which include, *inter alia,* the estimation of peak discharge, the analysis of shallow groundwater and the estimation of crop yields are deselected automatically in the initial *MENU*.

This automatic initial *MENU* creation, shown in Fig. 3.2, is a unique feature, as far as computer simulation models are concerned, in that the user is able to get results, within minutes, from the *ACRU* model by only supplying the catchment layout in a digital form. This *MENU* can then be fine-tuned, at a later stage. This initial *MENU* feature can also be used elsewhere in the world if the 49 GRID coverages are available for the particular area of interest.

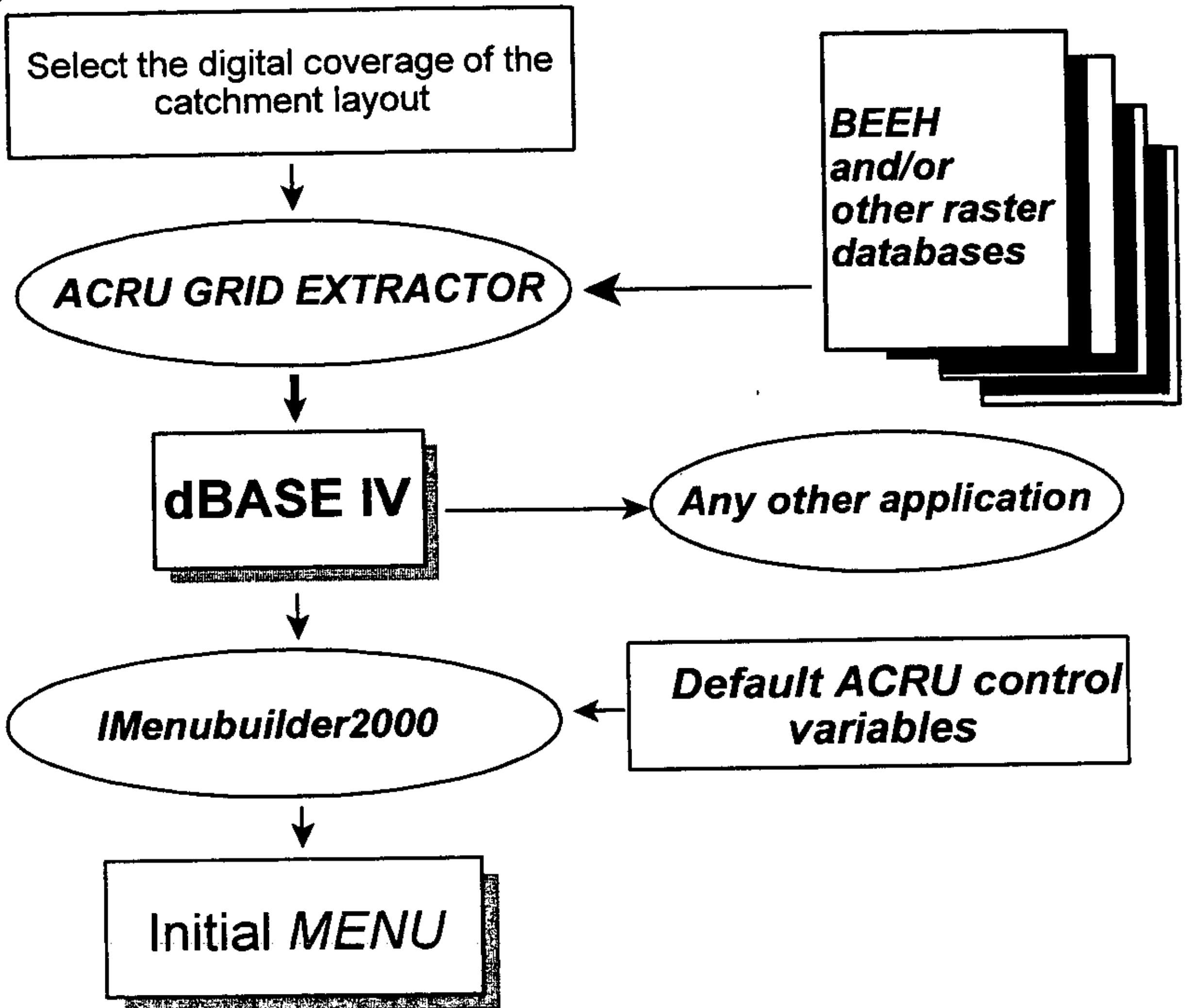


Figure 3.2 The procedure involved in the creation of an Initial *MENU*

3.2.3 **INPUT DATABASES**

Once an initial *MENU*, or any other *MENU* for that matter, has been created the user can access, via a host of utilities, a wide range of hydrological information that is contained within the BEEH data warehouse. The *South African Atlas of Agrohydrology and -Climatology* (Schulze, 1997) is one example of recent research by BEEH to enhance the quality and quantity of hydrological information. It is envisaged that when more detailed soils information becomes publically available in the not too distant future that the quality of modelling results will greatly improve.

The Quaternary Catchments Database of BEEH (Lynch *et al.*, 1996a, Meier,1997) contains all the information that is required by the *ACRU* model to produce simulation results at a Quaternary Catchments scale for South Africa.

3.2.4 NON-ARCVIEW OPTIONS

This section is devoted to the users of the *ACRU* modelling system who do not have access to ArcView or the Spatial Analyst module. The *IMenubuilder*2000 program requires ArcView and the *ACRU GRID EXTRACTOR* program requires ArcView and the Spatial Analyst extension. The new *ACRU MENU* is once again an ASCII file very similar to the *MENU* used with the *Menubuilder2* program. Care has been taken to ensure that utilities exist to convert *MENU*s between the "old" and the "new" *ACRU* modelling system. The main disadvantages that the non-ArcView users will encounter include the following:

- time taken to create an initial *MENU*
- if data from the BEEH raster databases is to be used then rectangles will have to be used in place of the subcatchment polygons to extract and average these values
- the user will have to use alternative graphical software to produce maps showing the output from the *ACRU* model.

The time-series display software, on the other hand, is designed to operate independently of the ArcView system and in most cases the user will be able to use the previous suite of *ACRU* utilities to display and manipulate the data that are to be used in the modelling exercise. Care has been taken to develop a suite of software (Table 1) to enable these users to use the *ACRU*2000 software to prepare their input data.

Table 1 Solutions for users who do not have access to ArcView or Spatial Analyst

Program	ArcView	Spatial Analyst	Work around for users who do not have access to ArcView
ACRU GRID EXTRACTOR	✔	✔	Extract the data via the CCWR using a rectangle rather than a polygon
*IMenubuilder*2000	✔		Use *Menubuilder2* and then convert the *MENU* to *IMenubuilder*2000 format

3.3 MODEL OUTPUT-RELATED UTILITIES

The research team decided not to develop any new model output utilities. In the case where no utilities exist, they have been developed and utilities have also been developed to create *ACRU* model output to the data formats required by existing utilities that have been developed by researchers elsewhere.

3.3.1 WDM UTILITIES

The simulation results from the *ACRU* model are written to either a WDM file or to an *ACRU* BIN file. A suite of utilities have been developed (Lynch, 1997a) to allow the users to manipulate the data stored in the WDM file to suit their requirements.

The user can launch the programs individually when required to perform certain tasks which may include the following:

- convert and add *ACRU* BIN files to a WDM file (*bin2WDM.exe*),
- convert *ACRU* single format files to WDM files (*single2BIN.exe*),
- determine the basic statistics of the data within the WDM file (*bstatsWDM.exe*), and
- output the datasets within a WDM file to a columnar ASCII file or add these datasets to an existing or new WDM file (*outputWDM.exe*).

An "expert system" (*expertWDM.exe*) has been developed to assist the user in determining which utility should be used to perform a specific task on a WDM file. This interactive program, illustrated in Fig. 3.3, initially prompts the user to decide whether an existing WDM file is to be used or whether a new WDM file is to be created. Depending on the response, another series of questions is posed. This process continues until the desired utility has been found and the user can the launch the utility at that point or exit from the “expert system”.

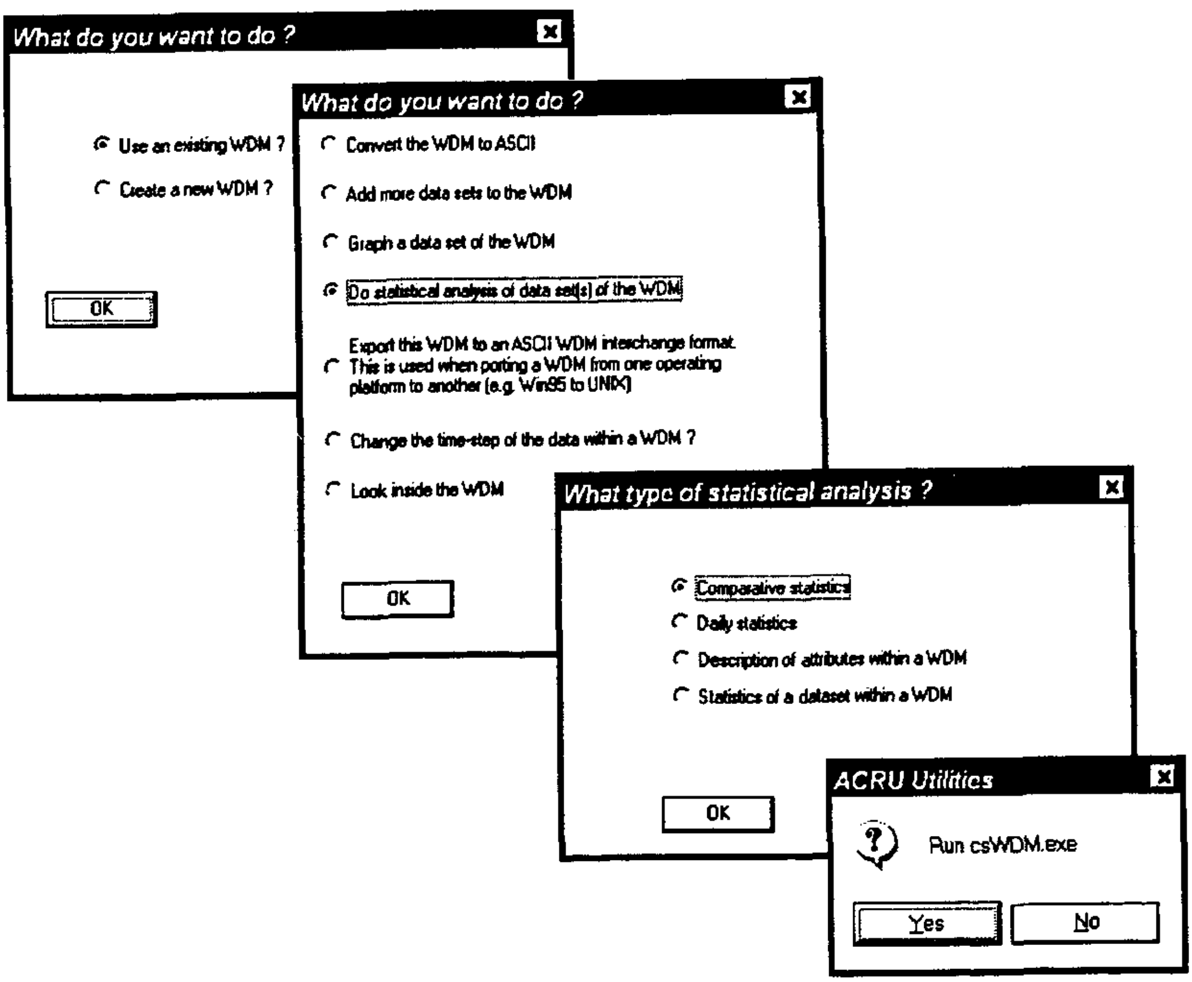


Figure 3.3 GUIs of the “expert system” guiding the user through a selection for a particular WDM utility

3.3.2 STATISTICAL UTILITIES

The suite of statistical utilities (Schulze *et al.*, 1995) that accompany the *Menubuilder2* suite of programs was developed to use ASCII files for input and is therefore compatible with the output utilities associated with the current version of the *ACRU* model as well as being compatible with the output from the *ACRU*2000 model. These utilities include the following:

- Daily time-step data interrogator (*ARF.exe*)
  This program uses as input an ASCII file that contains data at a daily time-step in a columnar format and calculates a variety of statistics that are required for analysing data at a daily-step.
- Monthly time-step data interrogator (*S.exe*)
  This program uses as input an ASCII file that contains data at a monthly time-step in a columnar format and calculates the relevant statistics pertaining to monthly data.
- Comparative statistics (*CS.exe*)
  This routine is normally used when simulated and observed data are to be compared to each other for a test of goodness of fit. The data can be independent of a time-step as the routine only reads two values per record. A large number of these statistics can be calculated using a standard commercial spreadsheet program, but this utility has been developed and perfected in FORTRAN over the past 10 years and includes all the major goodness of fit statistics that are required in interpreting simulation results. A graph showing the 1:1 line, the linear regression line and the point values is also displayed on the computer monitor and the output from the utility that uses WDM data is displayed in Fig. 3.4. This utility can also be used for comparing other data by simply noting which value in the input data file will be labelled observed and simulated in the output file.

3.3.3 GRAPHICAL UTILITIES

The research team decided not to develop additional graphical utilities but rather to develop utilities to interface the *ACRU* output to the already developed suite of graphical software that has been developed and collected by BEEH and other research organisations.

3.3.3.1 WDM GRAPHICAL UTILITIES

The only graphical utility that was available in 1996 to display WDM data was developed by the USGS and it was a very basic plot routine, *ANNIE,* which had no scope for customisation as far as line colours, text font size and the like are concerned. Software was therefore developed by BEEH to graphically display the contents of a dataset within a WDM.

The utilities that were developed in-house include the following:

- a GUI-enabled routine to display the time-series information contained within a WDM file (*plotWDM.exe*)
- an utility to display and perform comparative statistics on two datasets within a WDM file (*csWDM.exe*).

GENeration and analysis of model simulation SCeNarios utility (*GenScn*) was developed by the USGS (Kittle *et al.*, 1998) and acquired by the CCWR towards the end of this project. An initial investigation by BEEH revealed that the manual preparation of the additional data inputs were too time-consuming to be able to include this software in the stable of utilities that are being used by the *ACRU* modellers. The CCWR has subsequently minimised and automated most of the inputs. Upon re-examination, the utility was found to be useful.

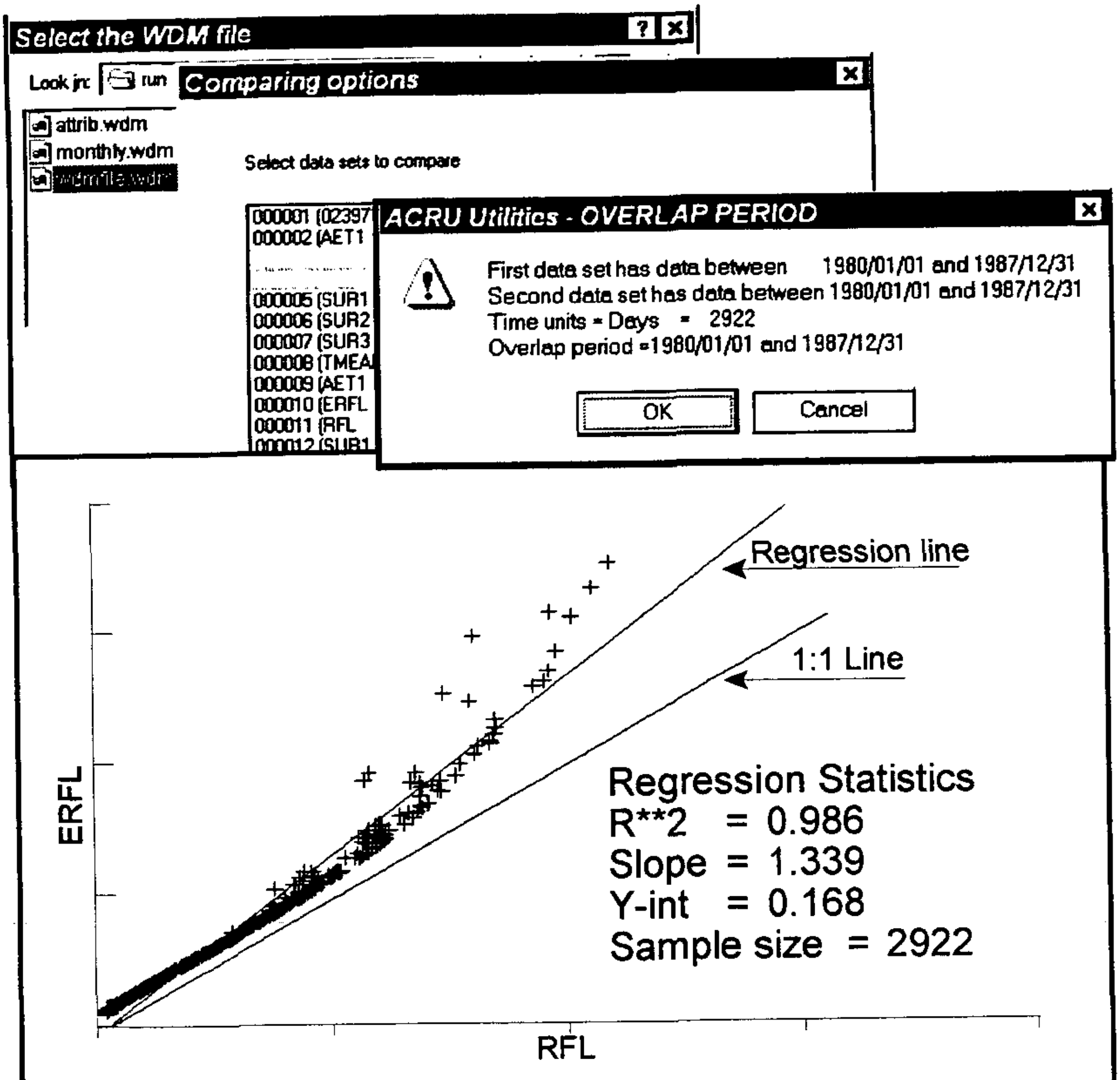


Figure 3.4 Computer monitor output when using the WDM comparative statistics utility (csWDM.*exe*)

3.3.3.2 **ARCVIEW GRAPHICAL UTILITIES**

The simulation results from the *ACRU* model are written to either a WDM file or to an *ACRU* BIN file. These results can then be extracted at a daily, monthly or an annual time-step using the *outputWDM.exe* or the *PRNT.exe* utilities and the output that is generated can then be joined, using the *ACRU JOINITEM* extension, to the digital coverage representing the catchment under investigation. Once this has been done the

ACRU ANIMATOR extension can be invoked to animate these results through time over the catchment. The simulation results can be joined to the shapefile and maps of the various scenarios can be displayed graphically or converted to a host of graphical file formats (Lynch, 1997b). Avenue can also be used to customise the display of these digital maps.

3.3.3.3 OTHER GRAPHICAL UTILITIES

The Times Series Analysis Software, TSOFT (Hughes, 1999), that uses the TSOFT data format, has also been accepted to assist in performing some of the graphical interpretation of the *ACRU* model simulation results and is described below.

Duration Curves

This series of program modules is designed for the display and analysis of duration curves using varying duration data (original data as variable interval at either daily, monthly or annual time-steps) and varying duration integration interval (1 day to 30 day using daily data or finer). It facilitates for several duration curves to be plotted, either using different data from the same flow site, or using data from different flow sites.

Time-Series Plots

These are designed to display time-series of observed or simulated variables, extracted from files, and allow for multiple axes on the same graph, and/or more than one graph.

Scattergrams

This facility is basically for the pair-wise analysis of the relationship between two variables, extracted from time-series. This could mean two different variables from the same multiple variable file, two variables from different files or one variable versus its time component. The program also accommodates the graphical display and analysis of the relationship, as well as some commonly used statistical relationships.

CHAPTER 4

*ACRU*2000 *MODEL*

GA KIKER

4.1 THE *ACRU*2000 MODEL PLATFORM

The *ACRU*2000 model has been restructured entirely to create a strong foundation for future code expansion and functionality. While the internal hydrological modelling features of *ACRU* described in Schulze (1995) have been retained, the underlying foundation has been changed to allow new features to be developed.

As an example, Fig. 4.1 shows the "traditional" *ACRU* (*ACRU* versions through the 300 series) conceptualisation of a catchment where a large catchment may be divided into subcatchments. The traditional *ACRU* execution technique simulates water flow in subcatchment #1 for the entire simulation period (for example, twenty years), then will move to subcatchment #2 and repeat the process. While this execution sequence is computationally efficient, it has limitations in representing the inter-connectedness of many developed catchments. Often water from one dam or river reach may be transferred to another catchment with specific rules governing how much water is moved.

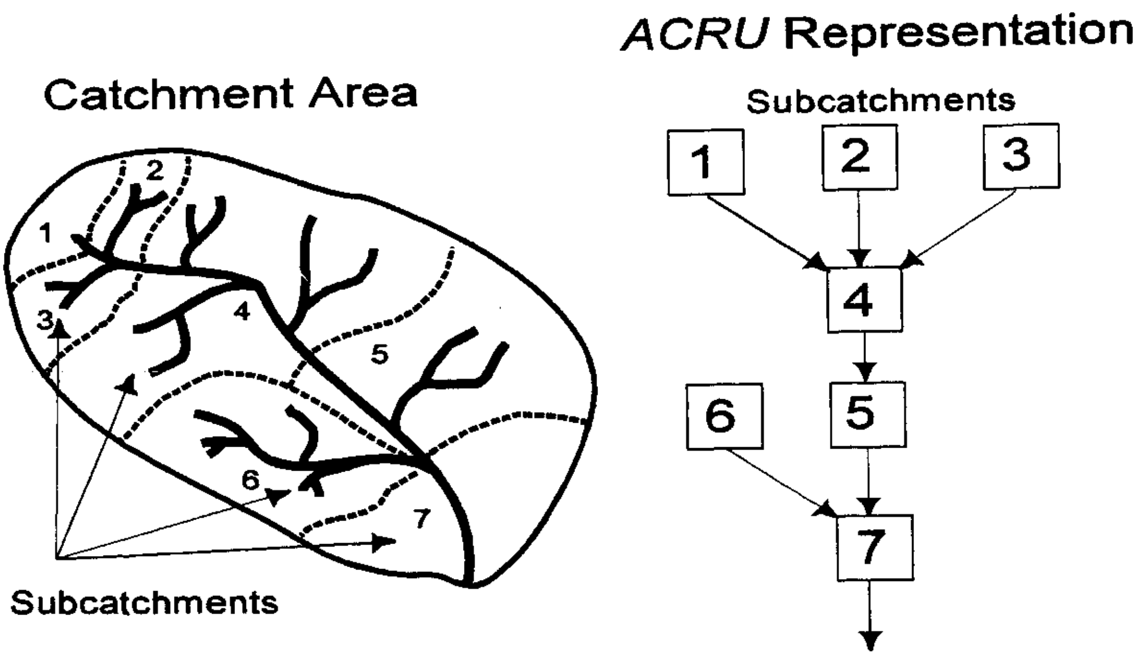


Figure 4.1 Current *ACRU* subcatchment configuration

As digital elevation and terrain models became more accessible to model users, user groups expressed opinions that *ACRU* should be more spatially explicit in its water flow simulations. Fig. 4.2 shows a different view of a catchment which is more spatial in its representation. The term subcatchment has been changed to land segment to show that an area need not be self-contained hydrologically and could be created from a digital elevation grid. Each land segment must have a flow direction which directs water outflows from one land segment to another. The model should execute each land segment once per time step so that all the land segments have been simulated before moving to the next time interval. Even the original structure in Fig. 4.1 can be utilised with this new execution structure where each subcatchment is simulated for one time step before moving to the next lower area. This new approach illustrated in Fig. 4.2 shows the basic spatial structure behind the *ACRU*2000 model.

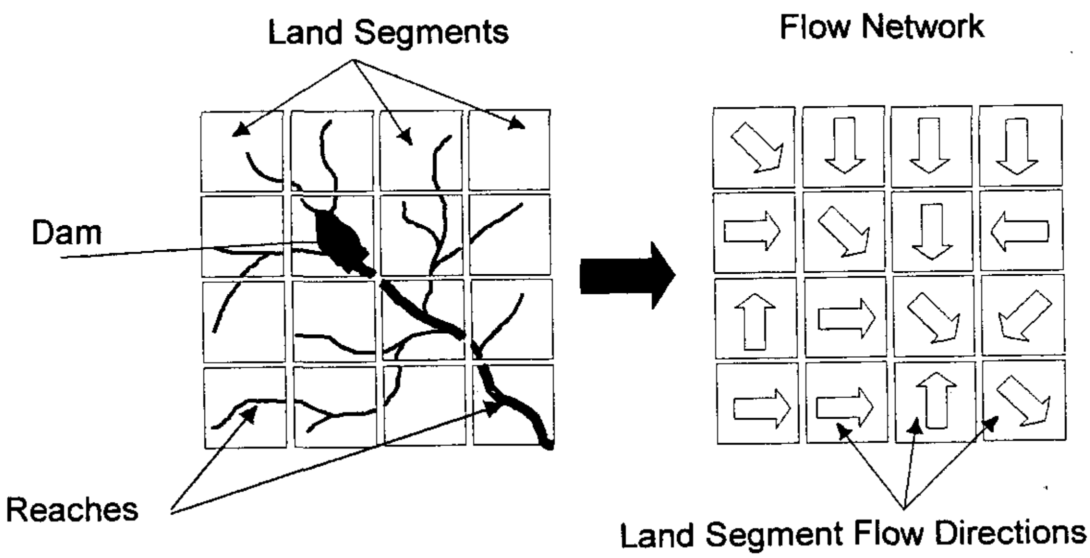


Figure 4.2 *ACRU*2000 model configuration into land segments

From a model development view, the *ACRU*2000 model is essentially composed of two sections, the object designs and the subsequent code development. In the object oriented programming methodology, the object design occupies a large part of the project effort as a proper design will ensure that future additions will not cause problems for the current or future developers. The following sections will detail the design and implementation methodology for the *ACRU*2000 model.

4.1.1 OBJECT ORIENTED MODEL DESIGN

Object Description

The *ACRU*2000 model is made of objects. Objects are defined as those things and processes that are important in modelling hydrological processes. Some of these objects are illustrated in Fig. 4.3, which illustrates the composition of typical land segment objects. In *ACRU*2000 there are three main types of objects, Components, Processes and Data. In the *ACRU*2000 object designs, one can easily distinguish what is what in the designs by the first letter of the object name is either C, P or D, corresponding to Components, Processes or Data. Fig. 4.3 shows some of the objects that can comprise a component object like land segment (in *ACRU*2000 terminology, CLandSegment.) Components are literally the building blocks of our hydrological world. They are the climate (CClimate), vegetation (CVegetation), and river reaches (CReach).

If the hydrological world consisted just of objects, not much would be happening as we are interested in the things that occur to these components. A process object describes an action or event that involves one or more objects. For example, the process of evaporation that transfers water from a river reach object (CReach) into the atmosphere (CClimate) could be called PEvaporation. Fig. 4.3 shows three process objects that describe water flows occurring at the surface, *viz.* (PSurfaceFlow), the sub-surface (PSubSurfaceFlow) and at the groundwater (PGroundWaterFlow) level.

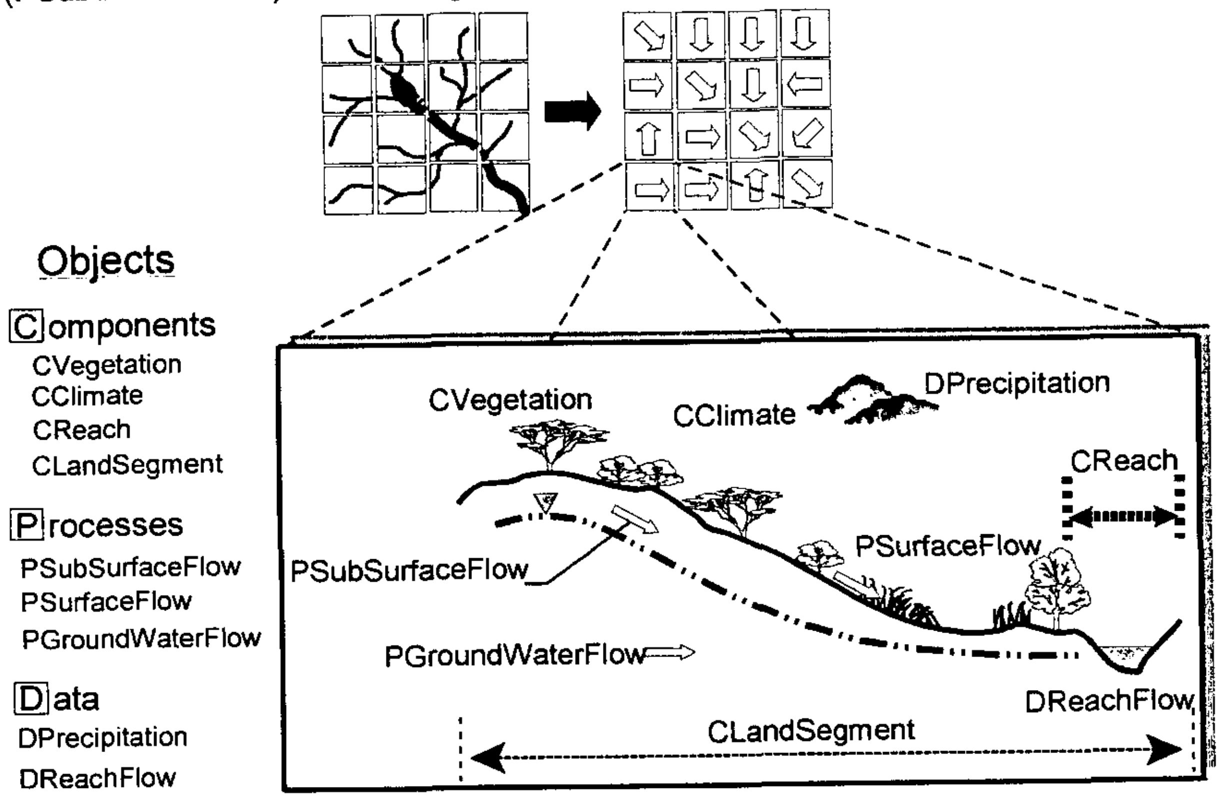


Figure 4.3 Objects in the *ACRU2000* model

Data objects are useful in helping to store descriptions about the various components and processes. For example, in Fig. 4.3, the CClimate object has a DPrecipitation object to describe the amount of rainfall that falls over a time step. Also the DReachFlow object describes the amount of water that flows out of a CReach.

The *ACRU*2000 object naming system is particularly useful when using the same word to describe two different things. If a user wanted to create a rainfall object that explains how precipitation could be used in exploring erosion, she/he may want to create a PPrecipitation object to describe the process of rainfall instead of simply a data value of rainfall as used in the DPrecipitation object.

Object Relationships

These component, process and data objects can be linked together in a variety of ways to simulate a hydrological system. The Unified Modelling Language (UML) provides a graphical method for building the various objects into useful designs. For example, in Fig. 4.4, one of the important components of *ACRU*2000 is the vegetation that may cover a land area. Because the vegetation is a component object, it is represented in *ACRU*2000 designs as a CVegetation object. Objects that are related to vegetation can be linked together in several ways, as shown in Fig. 4.4. An object can be a part of another object, as a CLeafCanopy object is one part of a CVegetation object. The CStem and CRoot objects also have this same "part of" relationship. In the UML object designs, this relationship is shown as a line ending in a diamond connecting the "part of " of object with the "overall" object.

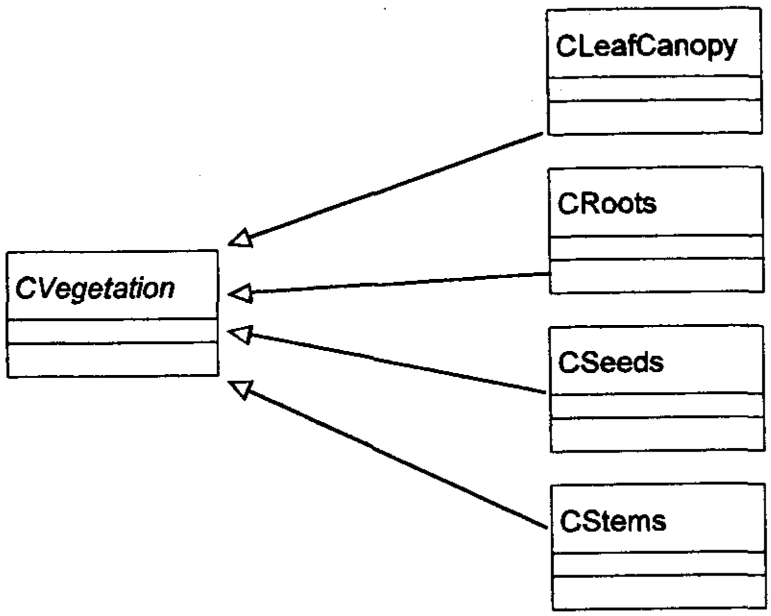


Figure 4.4 The UML aggregation, or "part of", relationship

Another important relationship between objects is inheritance. This relationship connects objects in a "parental" hierarchy, as exhibited in Fig. 4.5. In the *ACRU*2000 conceptualisation, land cover can be either pervious (some water flows through the land cover surface into the soil layer below) or impervious (where most water does not penetrate below but runs off the surface). This relationship is shown by the line with a triangle connecting the CPervious and CImpervious objects to the CLandCover object. Since there may be many types of pervious land covers in the our world, one possible type of pervious land covers would be vegetation. In Fig. 4.5, the CVegetation object "is a type of" CPervious object by the same inheritance relationship.

Following on, there are many different types of vegetation in the *ACRU* model world, so the CVegetation object is further broken down into other more detailed objects. For example, CEucalyptusPlantation and the CVeldGrass are two objects that inherit from the CVegetation object. The inheritance relationship allows the objects to gain common traits from their parents such as a leaf canopy, roots and stems while giving the space for differences. For example, one would expect the CEucalyptusPlantation object to have a very different growth and development than the CVeldGrass object. These differences would be listed only as needed in the "child" objects. This inheritance relationship creates a very efficient method of sharing similar traits between objects and including differences only where they are needed for modelling requirements.

The "part of" and "is a type of" relationships form the basis by which the objects are designed in the *ACRU*2000 model. Other object relationships exist to express how one object may use another object to go about its necessary activities. These associations are often used by process objects that may transfer water from one object to another. Fig. 4.6 shows a design diagram for the process of evaporating water from the soil layer (named PSimpleSoilEvap). The PSimpleSoilEvap object has an operation named evaporateWater() that will calculate the amount of water to be evaporated and will transfer the water from the soil horizon (CHorizon) object into the atmosphere (CClimate) object. To adequately calculate the amount of water to evaporate, this process object requires several data objects "owned" by other components. The arrows from the PSimpleSoilEvap object to the CVegetation and the CHorizon objects express in the design (and later in the Java code) that the process will get information from these objects. Because of the object inheritance principle, the PSimpleSoilEvap object can access data from the CClimate object because of its relationship with the CHorizon object (which subsequently inherits the CClimate relationship from its "parent" CClimateRelatedComponent). The "uses data from" relationship along with the "part of" and "type of" relationships represent the most prevalent types of communication between objects. Almost all the subsequent design drawings in the *ACRU*2000 model utilise these three relationships.

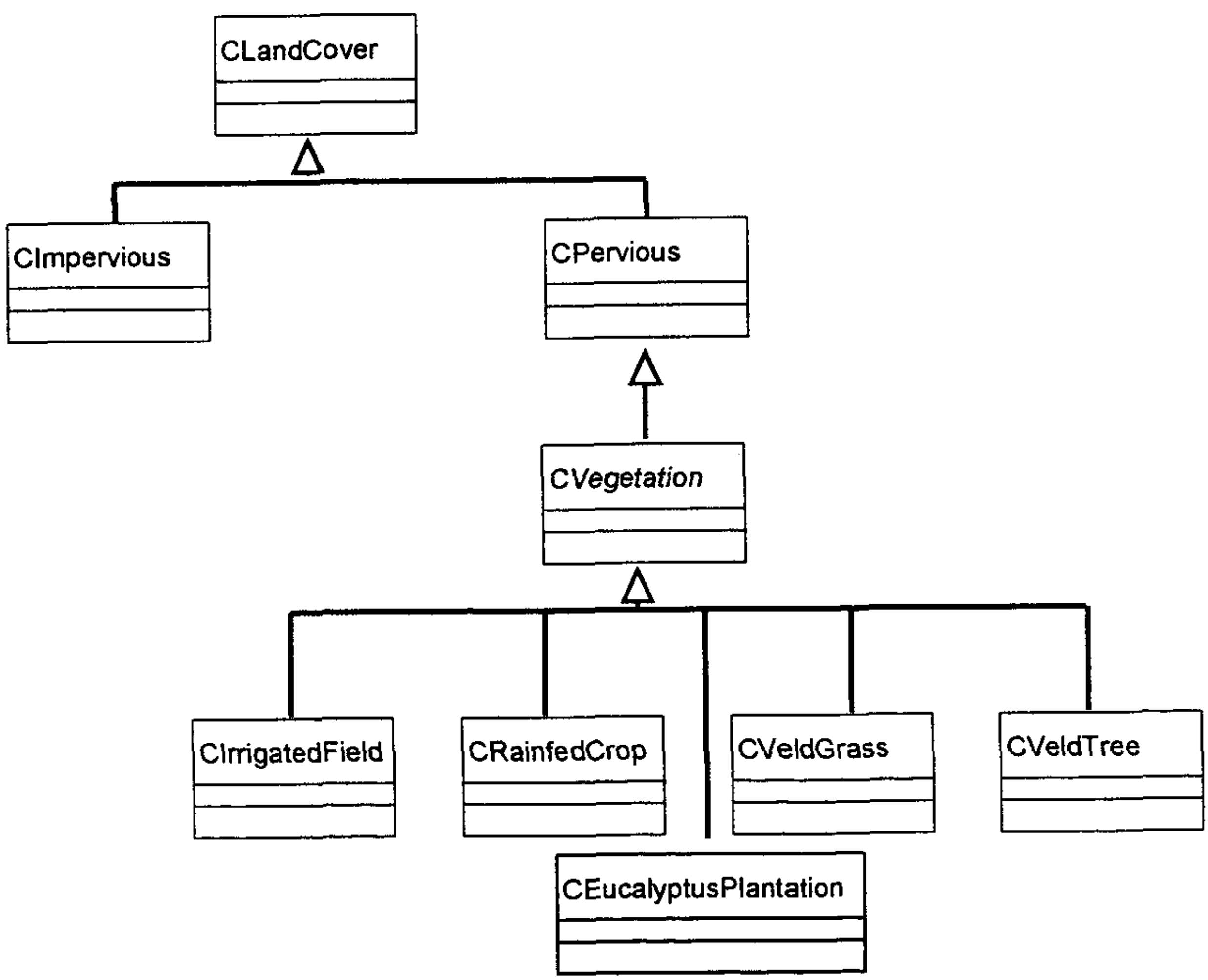


Figure 4.5 The UML inheritance, or "is a type of" , relationship

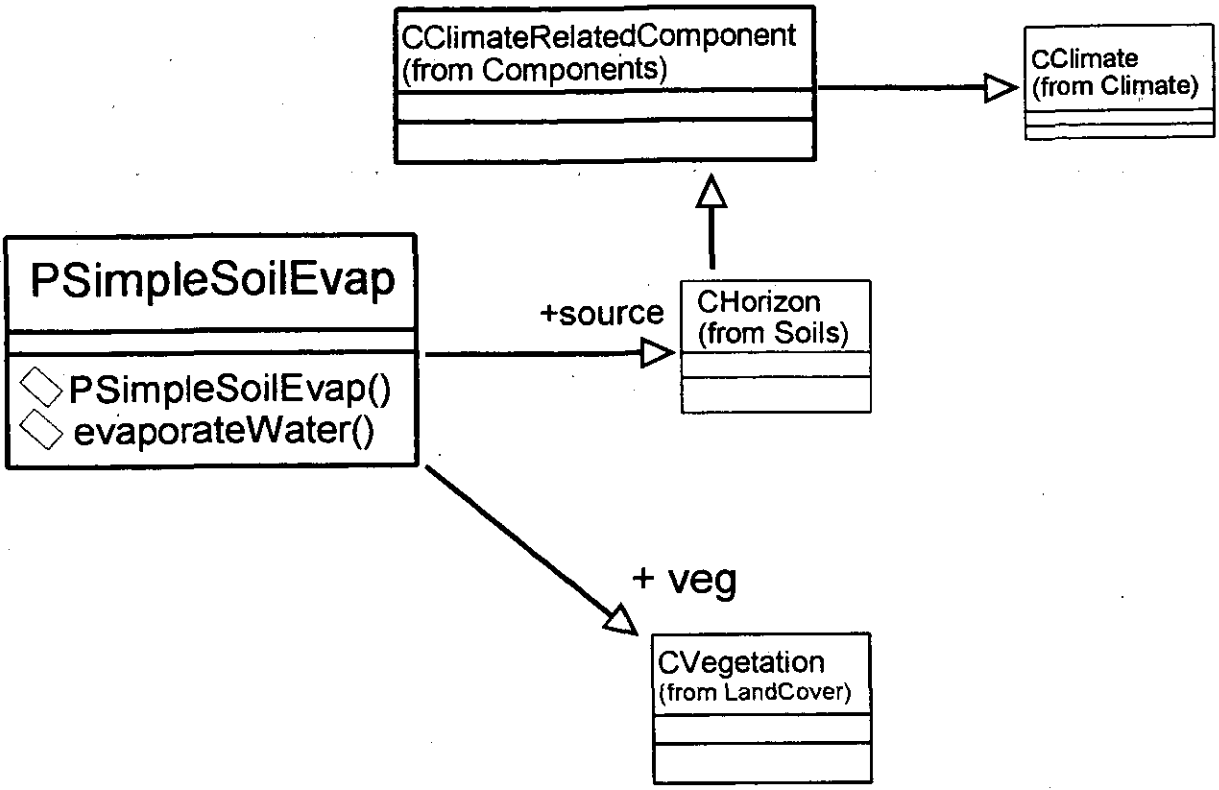


Figure 4.6 Design diagram for the PSimpleSoilEvap process

4.1.1.1 UNIFIED MODELLING LANGUAGE (UML) DESIGNS

The graphical design drawings shown in Figs. 4.4 through 4.6 are critical to the further use and code development of the overall *ACRU*2000 model. Literally hundreds of these diagrams are created by model developers to describe the various relationships between objects. The Rational Rose software is used to manage these many drawings and to translate the drawings into Java code which is then compiled into the overall *ACRU*2000 model.

Packages

The object designs in *ACRU*2000 start at a very general level and become more detailed as one "zooms in " on the particulars surrounding each object. When one is creating many objects, it becomes useful to put similar objects into "boxes" that allow for organised storage. These boxes are called "Packages" and collections of similar objects are grouped together for convenience and functionality. Packages can exist inside other packages to allow even more detailed sorting of objects. The most general design drawing of the *ACRU*2000 model (Fig. 4.7) shows all the packages (shown with file icons) used in the system. For example in Fig. 4.7, a package named Components holds all the component objects, a package named Processes holds all the Process objects and a package called Data holds all the data objects. These objects were introduced, previously. Also shown in Fig. 4.7 are other packages used for error handling (in a package named Exceptions) and for overall model guidance (in a package name Control). The various packages may be used or may be left untouched depending on what nature of additions are contributed to the *ACRU*2000 model.

If one were to look inside the Processes package in Fig. 4.7, one would find the design in Fig. 4.8. A package can contain objects or other sub-packages. If one were to continue to "zoom in" on the Evaporation Package in Fig. 4.8, one would eventually arrive at the PSimpleSoilEvap design diagram shown previously in Fig. 4.6.

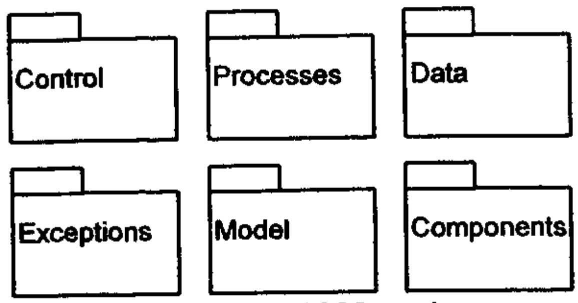
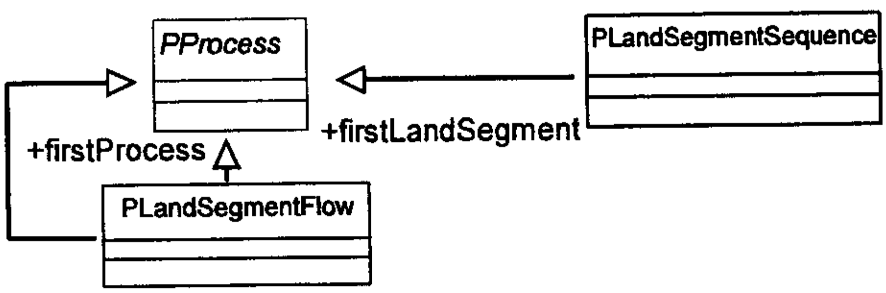


Figure 4.7 *ACRU*2000 packages that hold sets of objects

Through this hierarchy of packages, model developers can select certain areas of the code to develop and modify while other packages are left alone. The modular nature of the UML designs implies that changes to objects in one package have no, or at most a minimal, effect on other objects in other packages. A more comprehensive listing of design diagrams is included in the Technical Report.



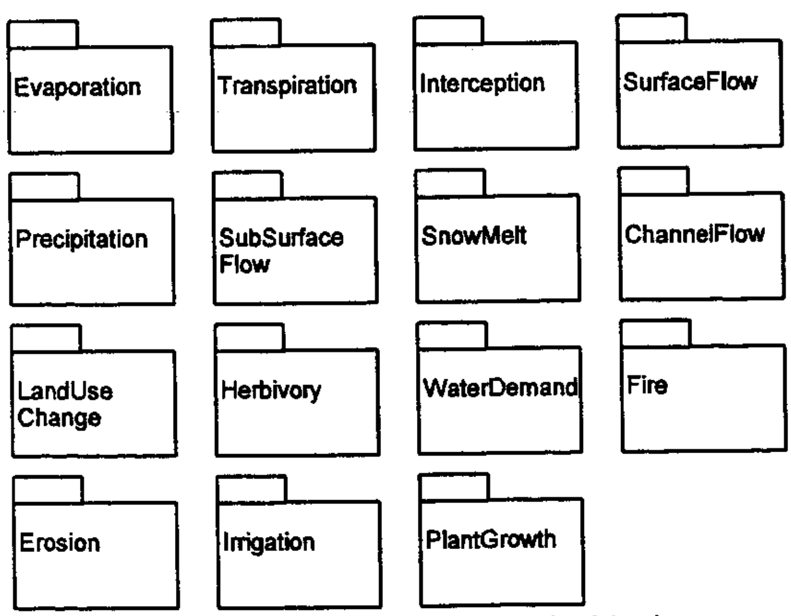


Figure 4.8 Objects and sub-packages inside the Processes package

4.1.1.2 TECHNIQUES OF IMPLEMENTING THE CODE

The next step after objects have been designed and linked with other required objects is the creation of computer code to implement the objects into the *ACRU*2000 model. Initially, test versions of *ACRU*2000 were developed in the Fortran90 and Java languages to test the compatibility of the UML designs in Rational Rose and subsequent code construction. A distinct advantage was seen in using Java over Fortran90 to implement object oriented design into practical computer code. While the actual Fortran90 and Java code appeared quite similar, it was found that Java had many more advanced capabilities in implementing the UML designs. One of the most important differences in the two languages was provided in the treatment of the inheritance ("is a type of") relationship. At this point, Java has a more powerful and robust inheritance system built into its code properties. The Java language was designed from the start to be object oriented and provides all the required capabilities to successfully implement designs into code. The Rational Rose Design Software has the ability to generate Java code and user reports (in Word format) directly from the UML object designs.

In addition, the "reverse" engineering function of the Rose software allows a user to make changes directly to the Java code and then "reverse" the altered code back into the model designs. All changes made in the code are then added to the design drawings. This tool is extremely powerful in keeping the design diagrams in step with code changes. With the Java code constructed, any Java compiler or development environment can be used to compile the source code. For this project, all Java code was compiled and debugged in the Java Workshop Version 2 (developed by Sun Microsystems).

4.1.2 USER INFORMATION

Input/Output Files

The *ACRU*2000 model is compatible with the *MENUs* and files constructed by the *ACRU*2000 utilities. *ACRU*2000 uses standard *ACRU* single and composite formats for climate files and creates output in both ASCII file and dBASE IV file formats.

One primary difference between inputs to the old and new *ACRU* models is the use of a control file which reports the flow network of land segments (or subcatchments.) Each land segment is described by its own menu file. The *ACRU*2000 model file uses the control file and the individual menu files to construct the required objects to be used in subsequent hydrological simulations. A user-defined list of output variables is read after the menu files to coordinate output of desired variables.

Sequence Order

The sequence of the various objects and processes can be altered as the components and processes are modular and this allows a high degree of freedom in setting the sequential order of execution. For this initial version of *ACRU*2000, the sequence of the *ACRU* (Version 328) model was used with a few additions. Fig. 4.9 shows a flow chart of the current sequence of processes.

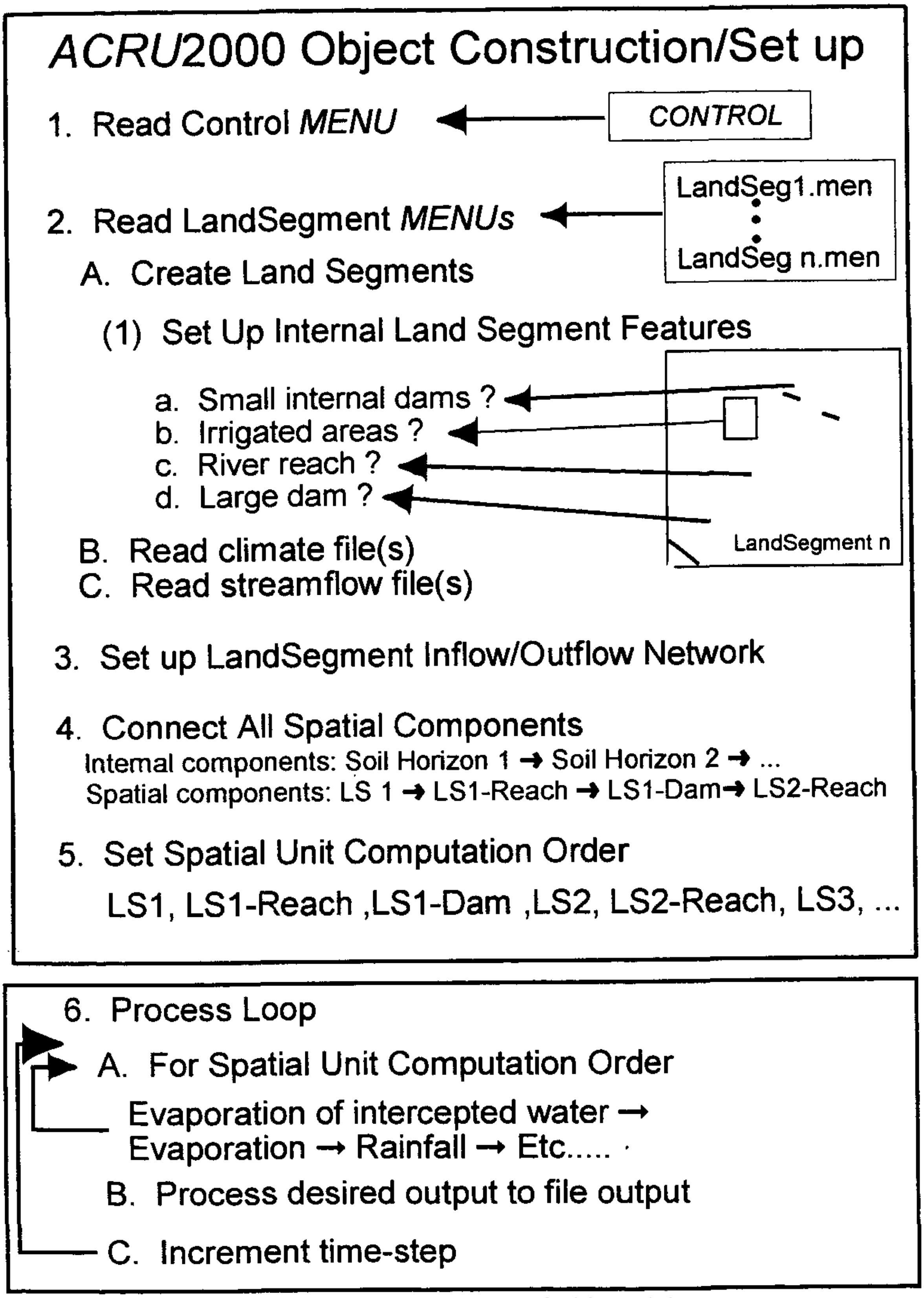


Figure 4.9 Execution sequence of the *ACRU*2000 model

The first action that occurs is that the *ACRU*2000 model reads the control file to create the proper number of CLandSegment objects to be simulated. This can be any number of CLandSegment objects since objects are created dynamically as needed in the model.
The second action reads the individual menu files for each CLandSegment object and creates any internal flow structures within a CLandSegment that may be of interest to the user. For example, irrigated fields or small internal dams can be constructed inside a CLandSegment object if required. Also, a CReach object can be created along with a larger CDam object to route and store water within the CLandSegment object. These internal flow routing objects (irrigated fields, small dams, reaches and larger dams) are only constructed when specified by the user. This action also reads the appropriate climate files where required to set up the CClimate object and may also read an observed streamflow file when available and when desired by the user.

The *third action constructs the land segment inflow and outflow network* so that water will flow from a land segment, or reach, as the user has designated. The fourth step connects all the spatial components within the model. This connection occurs within a CLandSegment object where one soil horizon is connected to the horizon below it as well as among different CLandSegment, CReach and CDam objects. The fifth action sets up the computation order of CLandSegments, CReaches and CDams to assure that the parallel flow moves progressively downstream.

The last action of this sequence begins the actual process of flowing water within the model. As all the proper spatial elements have been constructed and networked, the water will now move throughout the system by simulating the uphill land segments and systematically moving downhill, or downstream, for each time-step, which in the case of the initial *ACRU*2000 version is daily. As the hydrological processes of a land segment, reach or dam are simulated for the time-step, the desired output is processed for the file output. This process loop continues until the user-defined stop time is reached.

4.2 CONVERSION OF *ACRU* INTO *ACRU*2000

*ACRU*2000 development is a continuing process as both "previous" (*ACRU* 300 series) and the restructured modules are being added to object designs. In constructing the various component, process and data objects of the *ACRU*2000 model, first priority was given to existing *ACRU* (Series 300) functions. After the re-design and coding of the functions, the new objects were tested with an existing *ACRU* model (Version 325) to verify that each object's performance was correctly representing *ACRU* simulations. The following *ACRU* model concepts are currently incorporated into the *ACRU*2000 model. Most of these components are found in the *ACRU* version 300 series.

- Reads *MenuBuilder*2000 menu files
- Reads *OutputBuilder* files for model output
- Reads single and composite hydroclimatic files
- Calculates reference potential evaporation and maximum evaporation
- Vegetative canopy interception
- Canopy-intercepted-water evaporation
- Soil water evaporation

- Transpiration
- Rainfall
- Stormflow
- Infiltration
- Saturated soil water movement
- Recharge to intermediate zone
- Baseflow water storage and generation
- Water transfers from a dam or a reach to another dam or reach
- Option of an irrigated area within a land segment
- Detailed irrigation routines for irrigated fields
- Hydrological influence of small, upstream dams
- Flow routing generation
- Crop yield models
- Soil loss/sediment yield processes
- Writes user selected output to dBASE IV or ASCII Files.

4.3 *ACRU*2000 EXPANSION ATTRIBUTES

If one of the fundamental objectives of the *ACRU*2000 model is to be expansion-friendly, in order for new concepts and objects to be added in a systematic and orderly fashion with minimal impact on existing code modules. To achieve this goal, a methodology is required to systematically guide a potential developer through the various stages from preliminary cooperation, to object design, to final code generation and testing. The overall methodology in developing new objects is shown by the following steps:

- Define partnership roles
- Become familiar with Object Oriented design concepts and its application in hydrological modelling
- Review relevant object designs in *ACRU*2000
- Propose additions to *ACRU*2000 object design
- Implement design additions in Java language
- Testing new objects for correct operation.

This methodology is not a linear one where each step is started and then completed before moving on to the next phase. All of the stages are iterative in that, as a group learns and modifies its modelling approach, certain steps may be re-visited to sharpen the overall product. For example, a group may alter some of its proposed additions to the *ACRU*2000 design as it advances in its skill in developing object oriented designs. As the *ACRU*2000 development methodology is made to be cycled and revised, it is easy to improve and change designs to meet changing demands.

4.3.1 DEFINING PARTNERSHIP ROLES

This first step is one of the most important in creating useful communication and teamwork between the development groups and the *ACRU*2000 development team to meeting the research objectives. Communication is very important as one would find it difficult to work in isolation with object oriented concepts. Setting up specific roles and ground rules between groups can help devise a practical research plan. Some of the following questions could be answered in the collaborative research plan:

- Define partnership roles
- What are the objectives of the research collaboration ?
- What research objectives are shared among the research groups and which are exclusive to each research group ?
- What are the benefits for each participating group ?
- Which participants will be responsible for certain deliverables ?

4.3.2 LEARNING OBJECT ORIENTED DESIGN AND THE JAVA LANGUAGE

One of the primary purposes of the *ACRU*2000 model design is to allow organised and systematic expansion of the code without causing conflicts within the existing code structure. The basic requirements for those wishing to add to the modular designs and Java code include a familiarization with the UML design methodology and computer programming in the Java language. The UML is the primary object oriented design methodology used world-wide, with considerable literature and many references available. Some of these resources have been mentioned in the literature review in Chapter 2. In addition, the Java programming language has many books and on-line tutorials to aid in its understanding.

As this project has progressed, it was found that users with no experience in computers or programming were some of the easiest adopters of this methodology. Traditional functional programming does not provide useful experience as programmers often must "unlearn" many habits to interact with the new code designs. The overall UML/Java approach does not reward "quick fix" or "*ad hoc*" code development, but favours a systematic development process where design and integration plays the primary role in development. With the automatic code generation properties of Rational Rose, actual code writing and debugging occupies approximately 20% of the total project time.

4.3.3 REVIEWING RELEVANT OBJECT DESIGNS IN *ACRU*2000

After becoming familiar with the various object oriented terminology and its application in the *ACRU*2000 model design, users should begin to review the object designs that have relevance to their research objectives. Most additions to the *ACRU*2000 model add component, process and data objects to various packages and sections. Using the overall package diagrams to guide their additions, users can review the existing objects and propose new objects where required.

4.3.4 PROPOSING ADDITIONS TO *ACRU*2000 OBJECT DESIGN

Once the user has reviewed some of the relevant object designs, they may be ready to propose object additions to achieve their modelling objectives. Depending on their research objectives, there may be a few, or many, additional objects to add. In addition, the users may also request small additions to existing objects. At this point the *ACRU*2000 development team can aid users in the design of these new objects if required.

4.3.5 IMPLEMENTING DESIGN ADDITIONS IN THE JAVA LANGUAGE

The next step after objects have been designed and linked with other required objects is the creation of Java code to implement the objects into the *ACRU*2000 model. The Java language was designed from the start to be object oriented and provides all the required capabilities to successfully implement designs into code.

The Rational Rose Design Software has the ability to generate Java code and user reports (in Word format) directly from the UML object designs. In addition, the "reverse" engineering function of the Rose software allows a user to make changes directly to the Java code and then "reverse" the altered code back into the model designs. All changes made in the code are then added to the design drawings. This tool is extremely powerful in keeping the design diagrams in step with code changes.

Users are not strictly limited to code generation software as many objects can be constructed from other existing objects and changed only slightly to achieve new uses. This "code re-use" is a regular characteristic of object oriented design and can be utilised with or without code generation software. Once again, the level of Java code interaction depends directly upon the research objectives of the development group. With the Java code constructed, any Java compiler or development environment can be used to compile the source code. For this project, all Java code was compiled and debugged in the Java Workshop Version 2 (developed by Sun Microsystems) and the Jbuilder 3 Enterprise Edition (developed by Borland).

CHAPTER 5

THE SCOPE AND THE ROLE OF THE ACRU USER CONSULTANT

A PIKE and RE SCHULZE

5.1 OBJECTIVE

It is the wish of the WRC that products developed through their funding be applied as widely as possible. In 1995 the WRC therefore created the post of *ACRU* User Consultant within the "*ACRU* model development and restructuring" project. The objective was for the incumbent of this post to provide "excellent user support" for the many current and prospective users of the *ACRU* modelling system as well as the many other hydrological products applied frequently by outside users and which have been developed over the years in BEEH, through WRC funding.

This objective has been met through various general and specialised courses on the *ACRU* modelling system, one-on-one tuition, assistance via telephonic and Email discussions, undertaking collaborative work for the other WRC projects and for government and private clients, and developing a number of software programs to assist in the configuration of the *ACRU* model (for a list of these, see the Technical Report) as well as in the analysis of results.

5.2 CLIENTS SERVED

Table 5.1 presents a chronological overview of the various actions undertaken by the *ACRU* User Consultant. Where appropriate, the work of the *ACRU* User Consultant has been published in the *ACRU*cons report series, where the "cons" stands for the *cons*olidation of research and *cons*ultation with, clients requesting model applications by the *cons*ortium of team members within BEEH. Many of these *ACRU*cons reports are in the public domain, requests for some other require the collaborating organisation's permission before they are made available on request.

In summary, the activities listed in Table 5.1 illustrate the scope of activities of the *ACRU* User Consultant through

- provision of public and in-house model training
- model demonstrations and the interpretation of the results
- development of new internal model routines
- development of new routines to assist in the interpretation of the results
- development of a potential inter-model linkage
- research collaboration with South African partners
- research collaboration with international partners
- applications particularly in land use impacts on water resources studies, and for the RDP and
- assisting private consultants both within and outside South Africa, as well as the state and para-statal sectors.

Table 5.1. User Groups Assisted by the *ACRU* User Consultant.

Organisation	Person	Purpose/ Title of Report	*ACRU* Report
Users assisted in 1995			
Partners in Development	-	Reservoir Yield Analysis : Sikoto Dam, KZN	1
Hawkins, Hawkins & Osborne	-	Water resources assessment : Water Supply to Ndundulu Service Centre, KZN	2
Partners in Development	-	Reservoir Yield Analysis : Esidumbini Dam, KZN	3
Venter Forestry Consultants	K Venter	Hydrological impacts assessment of proposed afforestation in the Franklin area, KZN	4
Keeve Steyn Inc.	-	Impact of potential Mooi River water transfer scheme in KZN : Design flood estimation in the Mgeni catchment upstream of Midmar Dam	7
Institute of Hydrology (UK)	J Butterworth	Romwe catchment simulations (Zimbabwe)	-
DWAF	G Sewell	PC *ACRU*	-
Univ. Orange Free State	B Grové	Winterton study	-
GFJ	S Mallory	Rietvlei	-
MBB Inc.	J Brodie	Video Mapping of land use	-
Friedrich Schiller Univ. (Germany)	D Herpertz	Research into chill units	-
Univ. Natal	K Goddard	Postgraduate Project (Umvoti)	-
Univ. Natal	G Mathews	Postgraduate Project (Umvoti)	-
Univ. Natal	-	AUTOSOILS-K factors added	-
Forestek	C Everson	Cathedral Peak, *ACRU* simulations	-
Quinn & Assoc.	N Quinn	Mozambique simulations	-
Univ. Cambridge (UK)	M New	Water Budgeting Problems	-
Univ. Natal	-	Water Budgeting Subroutine	-
SASEX	E Schmidt	*ACRU* Demo	-
Mineral & Energy Affairs	-	*ACRU* Demo	-
Eksteen, V/D Walt & Nissan	-	*ACRU* Demo	-
Univ. Natal	-	Course (Nov 95')	-
Univ. Natal - ICFR	P Roberts	Forestek Decision Support System Unix-PC	-
Cornell Univ. (USA)	J Peel & G Kiker	*ACRU* Data Requirements	-
Dept. Agric (Glen)	D v/d Merwe	Interpretation of output from *ACRU*	-

Organisation	Person	Purpose/ Title of Report	*ACRU* Report
Users assisted in 1996			
Venter Forestry Consultants	K Venter	Assessment of hydrological impacts of proposed *afforestation in the Nyambathi catchment*	9
Forestry Industry Assoc	B Ferguson	Hydrological impacts of land use practices in the Pongola-Bivane catchment	11
Univ. Potchefstroom	W du Rand & J Booysens	Primary Production Simulations	-
Swaziland Dept Agric.	J Pape	*ACRU* Demo	-
Univ. Regensburg, Germany	T Scholten	*ACRU* Demo	-
Mottram & Assoc.	R Mottram	*ACRU* Demo	-
Charles Sellick & Assoc.	B Bonthuys	*ACRU* Inputs and Interpretation of Output	-
Quinn & Assoc.	N Quinn	*ACRU* Inputs and Interpretation of Output	-
CSIR	P Donald	*ACRU* Demo	-
Farm Investments	J Oberholzer & K Rolf	Forestry Permit Applications	-
Univ. Natal	S Lorentz	Simulation of Sediments in Zululand Catchments	-
Univ. Stellenbosch / CCWR	G Jewitt & R De Vos	Linking *ACRU* & HSPF	-
Forest Industries	B Ferguson	Use of Aerial Photography for Land Use Classification	-
MBB Inc.	K Meier	AUTOSOILS Spreadsheet	-
Umgeni Water	M Summerton	AUTOSOILS Spreadsheet	-
Umgeni Water	W Schäfer	Mvoti Project	-
Umgeni Water	W Schäfer	North Coast Project	-
KNPRRP	G Jewitt	Sabie Project Phase I	-
KNPRRP	M Dent	Sabie Project Phase II	-
Consult 4	G Quibel	Ethiopian Water Resources Study	-
European Commission	W Flügel	Water Resources Study in the Umkomaas Catchment	-
ICFR	P Roberts	*ACRU* Presentation	-
IGBP	AFRICAN GAIM	*ACRU* Presentation & Training, Kenya	-
Univ. Natal	Postgraduate Class	*ACRU* Training on Umvoti Catchment	-
SRK	C Schultz	Assisted in Organising WRSM90 Course	-
DWAF	J Moolman	Identification of sediment producing areas in the Olifants catchment	-

Organisation	Person	Purpose/ Title of Report	*ACRU* Report
Users assisted in 1997			
Umgeni Water	W Schäfer	Hydrological study of water resources of the Mdloti, Tongati and Mhlali catchments	16
KNPRRP	G Jewitt	Simulation of streamflows and sediment yields in the Sand and Sabie catchments	17
Anton Froneman Inc.	-	Hydrological impacts of land use practices on the Lenjane catchment : Forest Permit Application	19
Forestry Industry Assoc	B Ferguson	Re-assessment of land uses in the Bivane catchment and hydrological impacts thereof	20
Alpine Trout	D Trotman	Reservoir yield analysis : Alpine Trout	21
Appropriate Information Technology (ATI)	J Rivett-Carnac	Water resources assessment in the Nadi catchment for community requirements at Ehlanzeni	22
LHA	H Swart	Analysis of potential timber yield and of hydrological impacts from exotic plantations in the former Homelands of South Africa	23
Bradford Conning	A Nicol	*ACRU* Information	-
Umgeni Water	D Mnguni	*ACRU* Information	-
Umgeni Water	P Sithole	*ACRU* Information	-
Umgeni Water	M Summerton	Analysis of ISCW soils information	-
Umgeni Water	G Jewitt	*ACRU* Information	-
MBB Inc.	K Meier	Impacts of removing riparian vegetation	-
MBB Inc.	A Pott	Mvoti land use project	-
ICFR	R Kunz	Hydrological impacts of Afforestation	-
Scott Wilson Kirkpatrick & Co, Zimbabwe	J Butterworth	Transmission losses	-
IWRMS	Zimbabwe, Swaziland, Italian and German Partners	*ACRU* course and catchment configuration	-
Univ. Orange Free State	B Grové	Hydrological & economic Impacts of irrigation planning - Winterton (Planned releases from Bell Park Dam)	-
Mottram & Assoc.	R Mottram	Land Use and Irrigation EIA in Midmar	-
Centre of Water in the Environment, Univ. Wits	J Mackenzie & A van Coller	Sediment production in the Sabie catchment	-
B Bonthuys	-	Water use by plantations - Sand catchment	-
N Bezuidenhout	-	*ACRU* Information	-
Univ. Potchefstroom	W du Rand	Primary production simulations	-
CCWR	J Thorpe &	HSPF links	-

Organisation	Person	Purpose/ Title of Report	*ACRU* Report
	R de Vos		
Venter Forestry	K Venter	Forestry permit applications	-
Farmers' Support Group	R Auerbach	Water resources of small agricultural catchment	-
Users assisted in 1998			
Private	M Reineke	Assessment of the impact of natural vegetation clearance for commercial protea farming on streamflow and sediment yield in the Upper Molen River catchment, Eastern Cape	24
POLTECH	-	Hydrological study of the Bedford and Braamhoek hydro-electric pump storage scheme	25
Forestry Industry Assoc	B Ferguson	Impacts of afforestation on streamflows and of irrigation abstractions on Paris Dam	26
ZAI Inc.	G Sim	*ACRU* software upgrades	-
Council for Geosciences	G Botha	Assistance with reformatting input data, checking menus and running *ACRU* software	-
Various Consultants	-	Replied to queries regarding the SCS model and estimation of design rainfall	-
MBB Inc.	A de Klerk & K Meier	Assistance and advice with regard estimating the impacts of clearing riparian vegetation	-
MBB Inc.	A Pott	Assistance with the estimating the impacts of Land Use changes in the Mvoti catchment	-
SRK	R Breytenbach & P Reid	Sizing of water holes of golf course in SW Cape	-
IWRMS	V Taylor & D Dlamini	Assistance in the simulation of streamflows and sediments in the Umkomaas & Mbuluze catchments for the IWRMS project	-
Univ. Natal	R Schulze	Modelling Hydrological Responses to Land Use and Climate Change	27
Users assisted in 1999			
SASEX	E Schmidt	Regional simulation analysis of hydrological and yield responses of sugarcane under dryland and irrigated conditions (using *ACRU*)	28
SASEX	E Schmidt	Forecasting of sugarcane yields using *ACRU* and seasonal rainfall forecasts	-
Sutherland & Assoc.	G Matthews	Queries regarding setup and running of *ACRU*	-
DWAF/BKS	C Langhout	*ACRU* course & study of waters resources of the Olifants catchment	-

5.3 LESSONS LEARNT

The Water Act of 1998 and the resulting growing awareness of the need for the hydrological fraternity to move to daily, physical-conceptual based models has proved that the decision to fund a post to support the product of many years of WRC funded research was timely, very necessary and has been highly successful. This is evident from the list of users who have received support since 1995 (Section 5.2) and the more recent interest shown by major role players in integrated water resources management, for example, DWAF, BKS, Umgeni Water and others.

The increase in both the scope and number of applications of the *ACRU* model has taken one of two forms. On the one hand, many users have approached BEEH for personal tuition with the aim of building internal expertise with the model (e.g. Ninham Shand, MBB and BKS). Alternatively, they have recognised the potential for a model such as *ACRU,* but have felt that it would take too long to develop the experience base necessary to operate the model with consistent accuracy (e.g. INR, Umgeni Water, Forestry Industry Assoc). These organisations have, for the most part, opted to contract the application to BEEH and appoint a "steering committee" of DWAF and other specialists to give advice, supply information and provide personal contacts as well as ensuring that the deliverables are forthcoming in a form that is appropriate to and understandable by the relevant decision makers and managers of the organisations and their clients.

As positive as the WRC funded post of *ACRU* User Consultant within BEEH is seen to be (in the past and in the future) for the assistance to users and dissemination of WRC derived hydrological products, it is not without frustrations. These occur mainly for two reasons:

- Firstly, it is a sad reflection on South African operations hydrology that many, even experienced, water resources practitioners lack insight into the complexities of the terrestrial hydrological system and its interrelationships/feedbacks. This is becoming increasingly apparent in applications of land use hydrology and the practical enactment of many aspects of the National Water Act of 1998.
- Secondly, certain WRC funded projects at other institutions have sought to call for assistance in supplying them with hydrological information and *ACRU* simulations at very short notice and at a stage that their project has just about come to an end and results are required. This situation becomes very frustrating since the time constraints place severe restrictions on the quality of the study and the *ACRU* User Consultant is placed under great pressure with minimal returns in terms of experience gained and ideas generated to improve the modelling system while literally "bailing out" outside WRC projects. This situation has resulted in BEEH referring most "conventional" applications (applications which lack research or verification components) to local and well-trained consultants who have built up a reasonable level of expertise with the *ACRU* model.

Both these sources of frustration, however, testify further to the necessity of having the *ACRU* User Consultant in place.

CHAPTER 6

TECHNOLOGY TRANSFER

SD LYNCH

Some BEEH research documents pertaining to this project have been converted into the HTML (Hyper Text Markup Language) format and placed on the CCWR WWW server. For each of the published articles distinction has been made between the abstract and that of the full text of the paper. If users select the abstract icon, their Web browser will display only the abstract of the article.

The assumption that accessing the full paper icon would indicate an interest being expressed in the particular WWW article is valid as time is of the essence when searching for information in the present computer age. The WWW downloads that are recorded reflect the number of times that the full paper icon was activated. Counting the number of times that the abstract icon was activated would not be a true reflection of the usefulness of a particular article as the icon might have been activated out of curiosity (Lynch, 1996). During the course of this project the research staff were fortunate enough to be able to present their research findings nationally and internationally due to the generous financial assistance from the Research Office of the University of Natal and other sources. All the digital articles can be viewed by following the links from **http://www.ccwr.ac.za/~lynch2/dpub0.html**

The products of this research project will be made available to interested parties via the WWW or on CD-ROM. The dynamic nature of this research tends to favour the distribution of the technical programs and documents via an electronic medium and interested parties can then be kept abreast of enhancements and additions as this project forms a cornerstone of research within BEEH.

CHAPTER 7

THE FUTURE

SD LYNCH

Not many years ago an IBM compatible 80286 PC with 1 MB of RAM was considered a very powerful computer. The South African one minute by one minute of a degree altitude dataset of 781 rows and 1021 columns was read in approximately 10 minutes. At present, the South African 7.5 arc minute altitude dataset of 5677 rows and 7329 columns is read within one second on a 350MHz Pentium II PC with 256 MB of RAM.

When reflecting on future developments the word *utopia*, an imagined perfect place or state of things (Tulloch, 1993), comes to mind:

> If all the agrohydrology and -climatology information were available for the whole world at an infinitesimal spatial resolution, then automatic catchment delineation programs would be able to successfully delineate the catchment boundaries from the digital elevation model, daily rainfall and temperature amounts would be available at any point and the *IMenubuilder*2000 program could be used to create an automatic input *MENU* that would allow one to model the hydrological characteristics of any area in the world using the *ACRU* or any other appropriate simulation model.

With more than 200,000 users Worldwide, ArcView GIS is the world's most popular desktop mapping and GIS software (Theodore, 1999) and a vast number of these users will be developing their own software and will also share their software, as they have done in the past.

The suite of *ACRU* utilities need to be updated periodically as new software becomes available and the establishment of a committee to maintain and manage time-series analysis software is of paramount importance.

7.1 FUTURE GUI-GIS PROGRAMMING TRENDS

ESRI, the developers of the ARC/INFO and ArcView systems, has indicated that their support of multi-platform hardware systems will cease when the current version of their software is replaced in 2001. Their systems will run primarily under the Microsoft Windows environment with database access remaining platform independent.

Java, Delphi and Visual BASIC have found favour amongst the GUI-GIS programmers recently as this is in line with the new software development route that is being followed by ESRI, one of the world's top GIS software developers. The *ACRU* User Consultant will maintain a WWW site listing the software that is developed to assist users of the *ACRU* modelling system to manage their input and output data requirements.

CHAPTER 8

CONCLUSIONS: LESSONS LEARNT

SD LYNCH

A major challenge, in regard this project, was anticipating the advances in the fields of hardware and software developments. One can read the previews of what the software and hardware will be capable of doing, but only experience and in-house bench-marking will really determine if these advances are acceptable to the particular research being undertaken. Trends that influence the hardware market are often affected, in South Africa, by the strength of outside currencies. The cost of permanent hard disk storage has shown a dramatic decline since the start of this project and disk space is therefore not an issue anymore.

At the start of this project there were only a handful of programming languages that catered for GUI and the manipulation of geo-referenced data. Beta versions and initial releases of programming languages are often dependent on what hardware and software are commonly being used. Microsoft, for example, shelved their Fortran90 compiler after two years in production and ESRI has indicated that their support for the Unix platform will cease at the end of 2000. The initial acceptance of the ArcView environment was based primarily on the fact that it was supported on a large variety of hardware platforms. The geo-processing capabilities of the future single operating system ArcView far out-weigh the multi-platform support that was initially offered.

Ancillary software is only developed when the need arises and software such as *GenScn* only became available towards the end of this project. This highlights the fact that researchers all over the world are working together to develop common software to display and analyse spatial and time-series data.

Users of the ArcView system have currently developed in the order of 700 Avenue scripts that are in the public domain environment. The development of ArcView was at an initial stage at the onset of this project and the users and developers were hard at work developing and testing their programs.

The acceptance of operating systems also has a huge impact on what programming environments the developers of software tend to use. Microsoft Windows has become the *de facto* operating system on desktop computers today and the developers have tended to concentrate their programming in languages that are supported in this environment. Programming languages that were reviewed and rejected at the onset of this project may, therefore, today have been accepted due to the large user base and a prime example of these include Tcl/Tk, Visual BASIC and Delphi.

The *ACRU* User Consultant position, funded by the WRC, has had enormous success, as shown in Chapter 5. It is, however, worth noting that a considerable amount of the *ACRU* User Consultant's time was taken up "educating" users of the *ACRU* modelling system on the basic hydrological principles. The fact that these users did not always study, in advance, the major two references that accompany the *ACRU* modelling system, *viz.*

- Hydrology and Agrohydrology : A Text to Accompany the *ACRU* 3.00 Agrohydrological Modelling System (Schulze, 1995), and
- the *ACRU* User Manual (Smithers *et al.*, 1995)

resulted in so much time being spent in "educating" these users. This trend of "wishing to be shown how" is likely to continue, however, with the establishment of Catchment Management Agencies the *ACRU* User Consultant is likely to be even more fully engaged in what has become a noticeable trend, *viz.* that of outsourcing *ACRU* modelling to BEEH, where the model expertise resides, rather than doing the specialist modelling oneself.

The design of an object oriented structure of the *ACRU* model was a time consuming process that culminated in a hydrological model that can now be re-coded into virtually any object oriented programming language with ease. If a more advanced or applicable object oriented programming language were to replace Java then the re-coding of the *ACRU*2000 would proceed without due delay. The object oriented design has also resulted in a structure that allows for transparent and straightforward inclusion of new modules.

ACKNOWLEDGEMENTS

The research team wishes to express its appreciation to the following organisations and individuals for their valuable assistance during the project;

- the Water Research Commission for their financial assistance
- the Steering Committee of this project for their valuable input and guidance
- Jack Dangermond for the donation of ArcView3 and Spatial Analyst
- the staff of GIMS for their technical support
- the technical staff of the CCWR for valuable computing assistance
- the Research Office of the University of Natal for financial assistance to present research findings overseas and for further financial assistance in procuring and maintaining a state of the art desktop computer system
- Dr Olaf David of the Institute of Geography, Friedrich-Schiller University in Jena, Germany for object oriented modelling and Java support
- the technical and professional staff of BEEH and
- the post-graduate students in BEEH for assistance in Beta software testing.

REFERENCES

Acocks, J.P.H., 1988 Veld types of South Africa, Memoirs. Botanical Survey. South Africa, 40. Department Agricultural Technical Services, Pretoria, South Africa.

Bicknell, B.R., Imhoff, J.C., Kittle, J.L., Jr., Donigian, A.S., Jr. and Johanson, R.C., 1997. Hydrological Simulation Program - FORTRAN, User's manual for version 11: US Environmental Protection Agency, EPA/600/R-97/080, National Exposure Research Laboratory, Athens Ga., United States of America.

Booch, G., 1991. Object Oriented Design. Benjamin Cummings.

Booch, G., 1995. Object Solutions. Addison-Wesley. Reading Ma., United States of America.

Booch, G., Rumbaugh, J. and Jacobsen, I., 1998. Unified Modeling Language User Guide. Addison-Wesley. Reading Ma., United States of America.

CCWR, 1999. The Computing Centre for Water Research (CCWR), c/o University of Natal, Pietermaritzburg, South Africa.

Dent, M.C., Lynch, S.D. and Schulze, R.E., 1989. Mapping mean annual and other rainfall statistics over southern Africa. Water Research Commission, Pretoria, South Africa. WRC Report, 109/1/89. pp 230.

ESRI, 1997a. ARC Version 7.1.1. Environmental Systems Research Institute, Inc., Redlands Ca, United States of America.

ESRI, 1997b. ArcView Dialog Designer Version 3.1. Environmental Systems Research Institute, Inc., Redlands Ca, United States of America.

ESRI, 1998a. ArcView GIS Version 3.1. Environmental Systems Research Institute Inc., Redlands Ca, United States of America.

ESRI, 1998b. ArcView Spatial Analyst Version 1.1. Environmental Systems Research Institute Inc., Redlands Ca, United States of America.

Flügel, W-A., 1996. Hydrological Response Units (HRUs) as modelling entities for hydrological river basin simulation and their methodological potential for modelling complex environmental process systems. - Results from the Sieg catchment. Die Erde 127:43-62.

Flynn, K.M.,1996. US Geological Survey, 437 National Centre, Reston Va 20192, United States of America. Personal communication.

Flynn, K.M., Hummel, P.R., Lumb, A.M. and Kittle, J.L., Jr., 1995. User's manual for ANNIE, version 2, a computer program for interactive hydrologic data management: US Geological Survey after-Resources Investigations Report 95-4085. US Geological Survey, 437 National Centre, Reston Va 20192, United States of America.

Graham, I., 1991. Object Oriented Methods. Addison-Wesley. Reading Ma., United States of America.

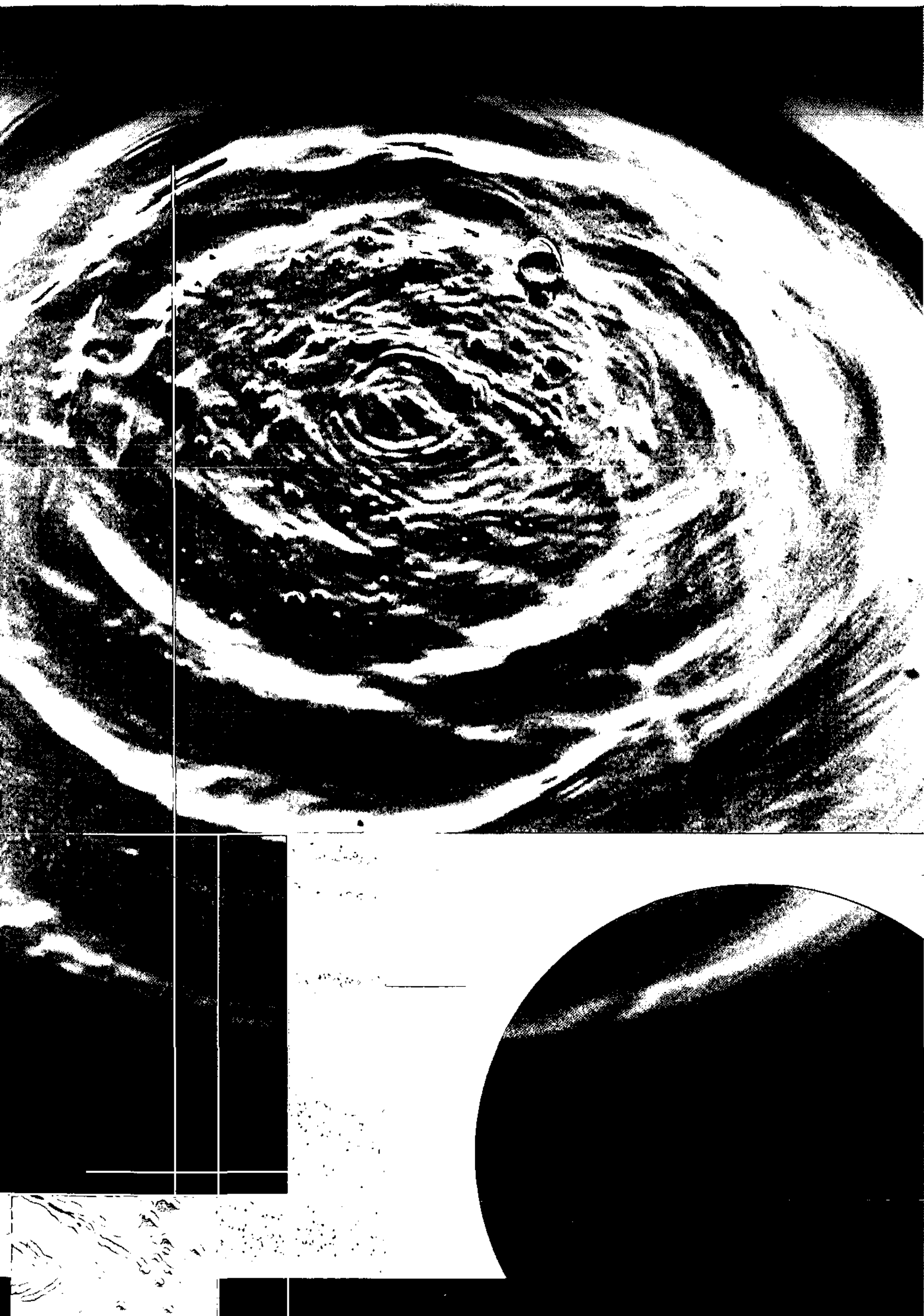
Hughes, D.A., 1999. Institute for Water Research, Rhodes University, Grahamstown, South Africa. Personal communication.

Jacobsen, I., Booch, G. and Rumbaugh, J., 1998. The Objectory Software Development

Process. Addison-Wesley. Reading Ma., United States of America.
Jones, C.A. and Kiniry, J.R., 1986. CERES-Maize: A simulation model of maize growth and development. Texas A&M University Press, College Station Tx., United States of America.
Kittle, J.L., Jr., Lumb, A.M., Hummel, P.R., Duda, P.B. and Gray, M.H., 1998. A tool for the generation and analysis of model simulation scenarios for watersheds (GenScn): US Geological Survey Water-Resources Investigations Report 98-4134. US Geological Survey, 437 National Centre, Reston Va 20192, United States of America.
Leavesley, G.H., Lichty, R.W., Troutman, B.M. and Saindon, L.G., 1983. Precipitation-Runoff Modeling System: User's Manual: US Geological Survey Water-Resources Investigations Report 83-4238, pp207.
Linacre, E.T., 1977. A simple formula for estimating evaporation rates in various climates, using temperature data alone. Agricultural Meteorology 18: 409-424.
Lynch, S.D., 1989. Geographical information systems in Hydrology. Proc. International Conference on Geographical Information Systems in southern Africa, University of Natal, Pietermaritzburg, South Africa.
Lynch, S.D., 1996. Using the Internet as a publishing medium. Unpublished material. Department of Agricultural Engineering, University of Natal, Pietermaritzburg, South Africa.
Lynch, S.D., 1997a. Developing a graphical user interface (GUI) in Fortran90 for the watershed data management (WDM) system. Proc. 8th Hydrological Sciences Symposium, Pretoria, South Africa.
Lynch, S.D., 1997b. Techniques for displaying PostScript files on the World Wide Web. Proc. 1997 ESRI User Conference, San Diego Ca, United States of America.
Lynch, S.D., 1998. Converting point estimates of daily rainfall onto a rectangular grid. 3rd International Conference on GeoComputation, University of Bristol, Bristol, United Kingdom.
Lynch, S.D., 1999. Automatic hydrological simulation model input using a GIS. Proc. 11th European Colloquium on Quantitative and Theoretical Geography, University of Newcastle, Newcastle-upon-Tyne, United Kingdom.
Lynch, S.D., Dent, M.C. and Schulze, R.E., 1989. Generation of a spatial database of climatic variables in a developing region. 12th International CODATA Conference 1990, Columbus Oh., United States of America.
Lynch, S.D., Lecler, N.L. and Schulze, R.E., 1996a. Using Real-Time Hydrological Data. Proc. 1996 ESRI User Conference, Palm Springs Ca., United States of America.
Lynch, S.D., Reynders, A.G. and Schulze, R.E., 1994. Preparing input data for a national-scale groundwater vulnerability map of southern Africa. Water SA: 20, 239-246.
Lynch, S.D. and Schulze, R.E., 1995. Techniques for estimating areal daily rainfall. Proc. 1995 ESRI User Conference, Palm Springs Ca., United States of America.
Lynch, S.D. and Schulze, R.E., 1996b. Digital Agrohydrological Atlas for Southern Africa. Proc. 1996 ESRI User Conference, Palm Springs Ca., United States of America.
McCown, R.L., Hammer, G.L., Hargreaves, J.N.G., Holzworth, D.P. and Freebairn, D.M., 1996. APSIM: A novel software system for model development, model

testing, and simulation in agricultural systems research. Agricultural Systems 50: 255-271.
McKivergan, P.,1999. Geographic Information Management Systems (PTY) LTD, PO Box 652, Halfway House, Midrand, South Africa. Personal communication.
Meier, K.B., 1997. Development of a Spatial Database for Agrohydrological Model Applications in Southern Africa. Unpublished MSc Eng. dissertation, Department of Agricultural Engineering, University of Natal, Pietermaritzburg, South Africa.
Microsoft Corporation, 1995. Fortran PowerStation v4.0. One Microsoft Way, Redmond Wa, United States of America.
Ousterhout, J.K., Undated. Scriptics Corporation, 2593 Coast Ave., Mountain View Ca., United States of America.
Penman, H.L., 1948. Natural evaporation from open water, bare soil, and grass. Proc. Royal Society, London A193: 120-146.
Quatrani, T., 1998. Visual Modelling with Rational Rose and UML. Addison-Wesley. Reading Ma., United States of America.
Rational Software Corporation, 1995. The Unified Method, Draft Edition (0.8). Rational Software Corporation, 18880 Homestead Rd. Cupertino CA., United States of America.
Ritchie J.T., 1972. A model for predicting evaporation from a row crop with incomplete cover. Water Resources Research 8(5): 1204-1213.
Rosenzweig, M.L., 1968. Net primary productivity of terrestrial communities: prediction from climatological data. American Naturalist 102: 74-76.
Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F. and Lorenson, W., 1991. Object-Oriented Modelling and Design. Prentice Hall International, Inc.
Rumbaugh, J., Jacobsen, I. and Booch, G., 1998. Unified Modeling Language Reference Manual. Addison-Wesley. Reading Ma., United States of America.
Schulze, R.E., 1975. Catchment evapotranspiration in the Natal Drakensberg. Unpublished PhD thesis. Department of Geography, University of Natal, Pietermaritzburg, South Africa.
Schulze, R.E., 1984. Hydrological models for application to small rural catchments in southern Africa: Refinements and development. Water Research Commission, Pretoria, South Africa, WRC Report No. 63/2/84. pp 248.
Schulze, R.E., 1989. *ACRU* : Background, Concepts and Theory. Water Research Commission, Pretoria, South Africa. WRC Report, No. 154/1/89. pp 235.
Schulze, R.E., 1995. *Hydrology and Agrohydrology* : A Text to Accompany the *ACRU* 3.00 Agrohydrological Modelling System. Water Research Commission, Pretoria, South Africa. WRC Report No. TT69/95. pp273.
Schulze, R.E., 1997. *South African Atlas of Agrohydrology and -Climatology.* Water Research Commission, Pretoria, South Africa, WRC Report No. TT82/96. pp 276.
Schulze, R.E., Angus, G.R., Lynch, S.D. and Furniss, P.W., 1990. Primary productivity over southern Africa: An example of agricultural resource determination. Agricultural Engineering in South Africa, 22, 22-33.
Schulze, R.E., Kiker, G.A. and Kunz, R.P., 1993a. Global climate change and agricultural productivity in southern Africa: thought for food and food for thought. In: Climate Change and World Food Security. Edited by T. Downing. Published in cooperation with NATO Scientific Affairs Division. Springer. New York, United

States of America.
Schulze, R.E. and Hohls, B.C., 1993b. A generic hydrological land cover and land use classification with decision support systems for use in models. Proc. 6th South African National Hydrological Symposium, University of Natal, Pietermaritzburg, South Africa.
Schulze, R.E. and Lynch, S.D., 1992. Distributions and variability of primary productivity over southern Africa as an index of environmental and agricultural resource determination. Symposium on Impacts of climatic variations and sustainable development in semi-arid regions. A contribution to UNCED, Fortaleza, Ceara, Brazil.
Schulze, R.E., Lynch, S.D., Smithers, J.C., Pike, A. and Schmidt, E.J., 1995. Statistical output from *ACRU*. *In*: Schulze, R.E. Hydrology and Agrohydrology: A Text to Accompany the *ACRU* 3.00 Agrohydrological Modelling System. Water Research Commission, Pretoria, South Africa. Report No. TT69/95.
Smithers, J.C. and Schulze, R.E., 1995. *ACRU* Agrohydrological Modelling System, User manual, version 3.00. Water Research Commission, Pretoria, South Africa. Report No. TT70/95.
Sun Microsystems, Inc., Java Workshop Version 2. 901 San Antonio Road, Palo Alto CA 94303, United States of America.
Theodore, J., 1999. ArcView GIS 3.2 Planned for Summer Release. Environmental Systems Research Institute, Inc., Redlands CA, United States of America.
Tulloch, S.,1993. The Reader's Digest Oxford Complete Wordfinder. The Reader's Digest Association Ltd., London, United Kingdom.



Water Research Commission

PO Box 824, Pretoria, 0001, South Africa
Tel: +27 12 330 0340, Fax: +27 12 331 2565
Web: http://www.wrc.org.za