



APSIM: a Novel Software System for Model Development, Model Testing and Simulation in Agricultural Systems Research

R. L. McCown, G. L. Hammer, J. N. G. Hargreaves,
D. P. Holzworth & D. M. Freebairn

QDPI/CSIRO Agricultural Production Systems Research Unit, Toowoomba, Australia

(Received 9 March 1994; accepted 31 October 1994)

ABSTRACT

APSIM (Agricultural Production Systems Simulator) is a software system which allows (a) models of crop and pasture production, residue decomposition, soil water and nutrient flow, and erosion to be readily re-configured to simulate various production systems and (b) soil and crop management to be dynamically simulated using conditional rules. A key innovation is change from a core concept of a crop responding to resource supplies to that of a soil responding to weather, management and crops. While this achieves a sound logical structure for improved simulation of soil management and long-term change in the soil resource, it does so without loss of sensitivity in simulating crop yields. This concept is implemented using a program structure in which all modules (e.g. growth of specific crops, soil water, soil N, erosion) communicate with each other only by messages passed via a central 'engine'. Using a standard interface design, this design enables easy removal, replacement, or exchange of modules without disruption to the operation of the system. Simulation of crop sequences and multiple crops are achieved by managing connection of crop growth modules to the engine.

A shell of software tools has been developed within a WINDOWS environment which includes user-installed editor, linker, compiler, testbed generator, graphics, database and version control software. While the engine and modules are coded in FORTRAN, the Shell is in C⁺⁺. The resulting product is one in which the functions are coded in the language most familiar to the developers of scientific modules but provides many of the features of object oriented programming. The Shell is written to be aware of UNIX operating systems and be capable of using the processor on UNIX workstations.

INTRODUCTION

Among the many changes taking place in the culture of Western agricultural research and development institutions, there is an increased recognition that a 'systems approach' is needed to meet the challenges presented by the complexities, uncertainties and conflicts in modern agricultural production systems — systems which are increasingly perceived to include future generations of farmers and to extend well beyond the farmlands. There is a widespread disenchantment with research and development aimed at the quick technological fix and an increasing interest in the development of research methodologies that address the long term economic and ecological issues. Recent progress in the use of models in the search for strategies for more efficient production, improved risk management, and more sustainable production systems (Littleboy *et al.*, 1992; McCown *et al.*, 1992; Thornton & McGregor, 1988; Hammer *et al.*, 1987) has raised expectations for an operational research approach in agriculture that complements an experimental approach. Certain attempts to use models to aid tactical decision making have been sufficiently successful to similarly raise expectations (Hamilton *et al.*, 1991).

However, there is a significant risk that these unprecedented opportunities may be squandered because our models are not up to the job. If Loomis was correct in 1985 in his observation on the state of simulation modelling that 'the field is still young and chaotic', this is not likely to be wildly incorrect in 1994. Seligman's (1990) conclusion that even after 25 years of work, models had produced few sustained successes in practical planning and decision making is sobering. This era produced enormous progress in techniques for modelling the physical and physiological processes in crop production. But to meet the new needs and opportunities, new priorities are needed. One of these is better predictive performance; Loomis and Seligman emphasize that this is most likely to be achieved by testing and improving the best of existing models rather than inventing new ones. This requires concentration of investment in collection of good field data. A second priority is to move from the performance of single crops to performance of cropping systems in terms of both crops and soil. A third priority is better software that reduces the overheads of simulation modelling in research, facilitates efficient convergence of modelling effort both within and among teams, and allows flexible and efficient reconfiguration for simulating different production systems.

This paper deals primarily with these latter two priorities. We first address needs against the backdrop of current prominent software packages. We then describe the Agricultural Production Systems Simulator (APSIM) designed and developed to meet these needs.

The need for better software

We begin definition of software needs with the general need for tools to aid the search for better farming strategies and development of aids to decision making. In many of the farming systems where this need is most pressing, rainfall is uncertain and often deficient, and soil degradation threatens the economic future of crop production. The required simulation package must deal credibly with both the season-to-season variability of production and the long-term trends in production in response to changes in the soil resource. To do this requires

- (1) crop models with sufficient sensitivity to extremes of environmental inputs to predict yield variation for analyses of economic risks;
- (2) models to simulate trends in soil productivity and erosion as influenced by management, including crop sequences, intercropping, and crop residue management;
- (3) software that enables efficient evolution of the modelling system by research teams.

No existing cropping systems model of which we are aware provide all three features. DSSAT (Uehara & Tsuji, 1991) brings together a number of crop models, i.e. the CERES models (e.g. Jones and Kiniry, 1986), GRO models (e.g. Hoogenboom *et al.*, 1994), and SIMPOTATO (Hodges *et al.*, 1992) with useful utilities for simulation studies (e.g. Hunt *et al.*, 1993). Models of this type have the sensitivity to environmental extremes needed for risk analysis (Carberry *et al.*, 1989; Keating *et al.*, 1993). But although certain soil and crop management effects on water and nitrogen supply to a crop can be simulated (Keating *et al.*, 1991), the lack of a ready means to simulate cumulative effects of crops on the soil prevents DSSAT from being a cropping systems simulator. Because each crop model has its own soil routines, a change of crop brings a soil whose properties and initial states must be specified.

Alternatively, EPIC and NTRM represent a class of cropping system models designed to simulate cumulative effects of cropping systems on the soil (Cole *et al.*, 1987). But the simple generic crop models used to provide flexibility in changing crops without changing soils have been shown to lack the sensitivity needed to enable them to be used for analysis of risk or simulation of performance at specific locations and seasons (Williams *et al.*, 1989; Steiner *et al.*, 1987).

Production of high quality software has rarely been a high priority in model development. In a research environment, code is generally written by scientists and it is efficient to use old code that suits (and works) in new programs, often with patched enhancements. Although such a

process contributes to efficient development of prototype models, truncation of the software development process here has resulted in code which is generally difficult to read, unreliable, and expensive to maintain in an on-going model testing/improvement process. An exceptional investment in software has been made in the development of DSSAT. From a collection of crop models from diverse sources, considerable progress was made in attaining similarity in appearance to users, standard input and output files, etc. But without comprehensive re-engineering, the lack of modularity has hindered the transfer of improvements from one model to another and hindered modification of the models for new applications (Hodges *et al.*, 1992).

APSIM results from a convergence of two previous efforts to achieve the combination of features 1 (high sensitivity of crop models), 2 (ability to simulate a wide range of configurations of crops, sequences, mixtures and management practices and effects on trends in soil productivity), and 3 (software which is designed and tested). The first, PERFECT (Littleboy *et al.*, 1992), was developed primarily to simulate the effects of erosion on the productivity of vertisols in the Australian subtropics, as influenced by soil management. The approach was to utilize existing crop routines with high sensitivity together with enhanced routines of soil management, soil water movement and erosion. While providing a useful tool for system analysis, e.g. effects of erosion on productivity (Littleboy *et al.*, 1992) and climatic risk (Hammer *et al.*, 1987), continued development and extension to other users has been hampered by inadequacy in feature 3. A second development, AUSIM (McCown & Williams, 1989), adopted CERES-Maize as a crop template to achieve feature 1. Feature 2 was achieved by design of a 'plug in-pull out' system and a comprehensive re-engineering of CERES-Maize to achieve highly independent modules of crop growth, soil water and soil nitrogen, allowing flexible recombination of crop routines in order to simulate rotations, intercrops, weeds, etc. and to facilitate transfer of model improvements. The investment in feature 3, in addition to making new modules highly structured and highly logical in terms of function content, was a comprehensive testing of new code. The efforts of the AUSIM and PERFECT development teams have been combined to produce APSIM, which goes beyond its predecessors in achieving all three features.

General design features of APSIM

Structure of the conceptual model

The key concept is the central position in the model of the soil rather than the crop, in spite of the fact that the output generally of greatest

interest is crop yield. Changes in the status of soil state variables are simulated continuously in response to weather and management. Crops come and go, each finding the soil in a particular state and leaving it in an altered state. All crops share the same soil and aerial space in which various processes take place, e.g. soil water and nitrogen transfers and transformations, surface residue decomposition, and radiation interception. This structure allows ready simulation of the effects of one crop on another *via* its effects on the soil, in both sequences and mixtures of crops.

Structure of the program

This model concept is implemented in the APSIM program using the structure shown in Fig. 1. Various high order processes, e.g. production of a crop, soil water balance etc. are represented as modules which relate to each other only through a central control unit, the 'Engine'. Plant growth modules are interchangeable, and more than one growth module can be connected simultaneously. This plug in-pull out capability enables

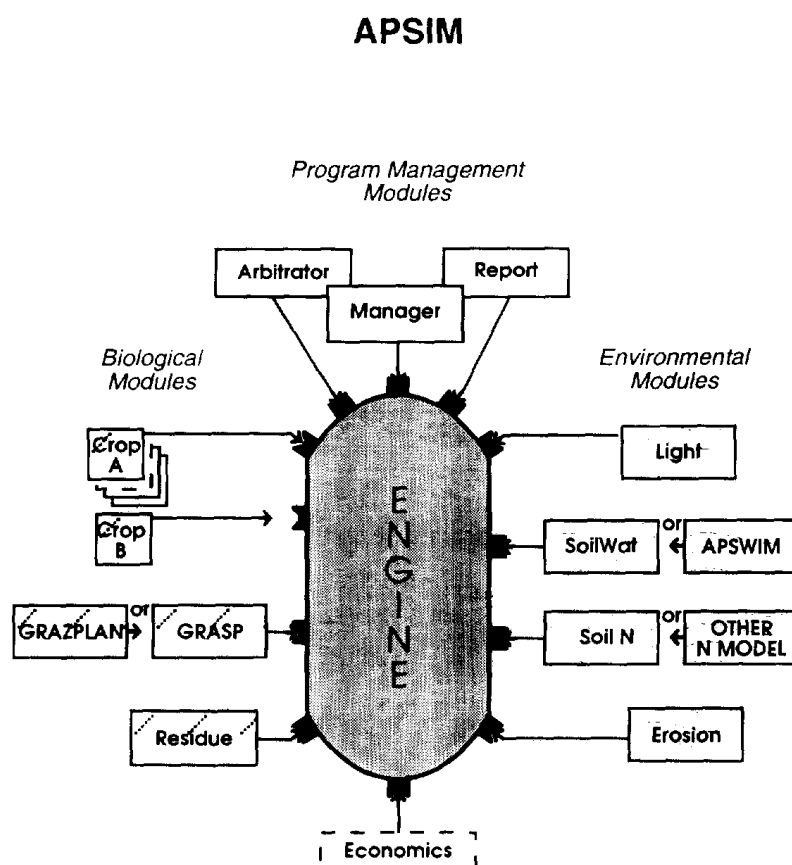


Fig. 1. The structure of the APSIM program. Modules are readily pulled out or plugged in. (See Table 1 for origin of modules; dashed box indicates module still under development.)

the achievement of flexible simulation of cropping systems (sequences and mixtures) while using the crop models most capable of accurate yield prediction. Before a simulation run, the growth routine for the required crops are selected from a library of crop models available and plugged in by selection in a screen menu (Fig. 2).

Dynamic simulation of a cropping system requires representation of relevant management actions realistically taken in response to conditions. In APSIM, this is accomplished by the 'Manager' module (Fig. 1). Actions (e.g. choice of crop, planting, application of fertilizer, tillage or irrigation) can be either scheduled or controlled using conditional rules. The language for expressing rules is 'If....condition(s) satisfied., then...action(s)'. This form allows great flexibility and enables ready construction of complex rules. The 'System Log' records interventions of the Manager.

Other 'Program Management' modules are also shown in Fig. 1. The 'Report' module implements the output of variables nominated in the control file. An 'Arbitrator' module controls competition for resources among crops in mixtures (intercrops, weeds and crops, pasture components). An 'Interactive module' will provide a visual-interactive option similar to that developed for CERES-Maize by Hargreaves and McCown (1988).

A high degree of flexibility is achieved with 'Biological and Environmental' modules. Different versions of modules can be interchanged or a module can be absent without causing disruption (Fig. 1).

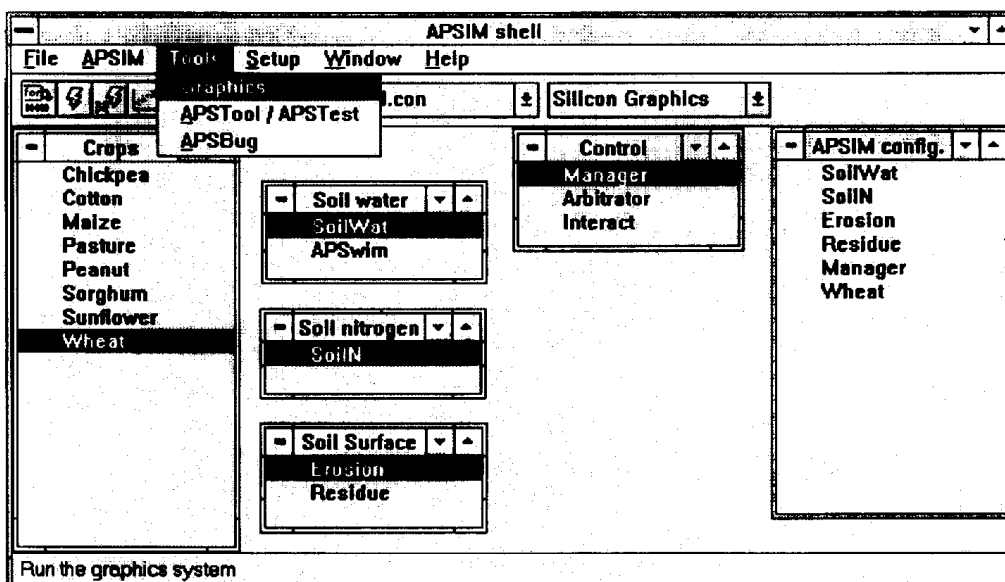


Fig. 2. The main screen of APSIM showing pull-down menus and modules available for a run configuration.

The Engine has been designed to perform mainly one function, i.e. the passing of messages to modules from either itself or other modules. Program functions, such as reporting, which might well be included in the Engine, appear instead, as Program Management modules (Fig. 1). This forcing of all functions other than communication into modules means that enhancements to APSIM can be added with minimal change to existing code. For example, the Interactive module will be added with no change to the Engine or other modules. However, to provide some structure to description, we distinguish among three types of modules in Fig. 1, the Engine sees them identically and, contrary to the impression given in Fig. 1, there is no limit to the number of modules the Engine can accommodate (but there is a cost in run speed as the number increases).

The modules and the engine are programmed in FORTRAN 77. In spite of certain sacrifices in programming opportunities, this is a concession to (a) the fact that FORTRAN continues to be the predominant programming language of simulation modeling in agriculture and (b) the high cost of re-coding extant FORTRAN routines.

The user interface

A user interface, developed in C++ and Visual BASIC, provides a suite of tools for developing, testing and maintaining module code, running the model, and presenting and analysing output. This computing environment is provided as a Microsoft WindowsTM program, with the user-friendly features of screen displays of multiple windows, mouse and keyboard input, pull down menus etc. The model is configured for a task from the main menu and shown in the Configuration window (Fig. 2). The several tools which are available are located on pull-down menus.

The shell has been written to be aware of UNIX operating systems and be capable of using the processor on UNIX workstations for APSIM runs. This offers speed advantages for large runs without much loss of the convenience of the PC Windows environment.

Although seamless in practice, description of the user interface is simplified by considering separately the two major applications: (a) program development and maintenance and (b) simulation.

Simulation

The environment for using APSIM for simulating production system performance is shown in Fig. 3. This is designed for ease in manipulating the configuration of the production system model, presenting appropriate input data, making simulation runs, presenting output, and making comparisons and analyses of both physical and economic outputs.

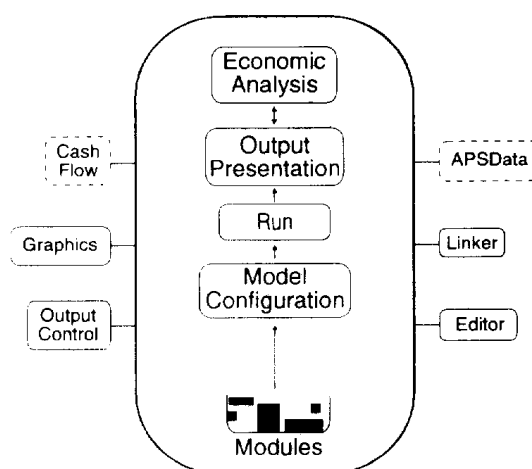


Fig. 3. The user environment for system simulation. Flow of operations are shown within the boundary and tools and resources outside. (Dashed lines indicate element still under development.)

Modules are selected for inclusion in a run by dragging labels of required modules into the APSIM Configuration window (Fig. 2). A run is specified by construction of a 'Run Control File' using the Editor. This file automatically shows the modules and files that have been selected on the screen. The name of the output file is designated and output characteristics can be specified, e.g. variables reported, intervals, or event cues. Information for controlling the run is supplied as Start and End dates and by creating the 'Manager' file.

The executable file is created by selecting 'Compile APSIM' from the 'APSIM' pull-down menu (Fig. 2). This links selected modules and builds a runtime file. A run is initiated by selecting 'Run APSIM' from the 'APSIM' menu. Output tables can be examined using the editor or graphed. From the 'Tools' menu, simple plots of time series, x - y , actual versus predicted, concentrations versus soil depth, or cumulative probability can be selected. From within the shell, output tables can also be imported into Microsoft ExcelTM where graphs or statistical analyses can be generated using pre-written macros.

Program development and maintenance

The environment for program developers is illustrated in Fig. 4. The utilities shown outside the main boundary are available for aiding the operations shown inside. The user can install any preferred editor, compiler, debugger or linker.

While adoption of highly structured programming provides a simplified logical framework for a model which makes maintenance easier, it

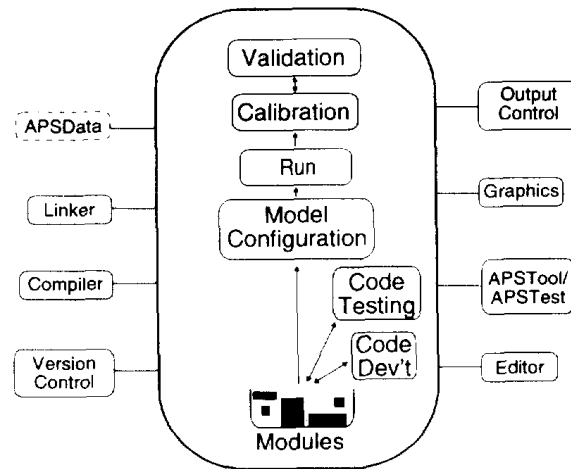


Fig. 4. The user environment for program development and maintenance.

also results in an increased number of subroutines with increased internal documentation text. To facilitate navigation among the numerous elements with relatively voluminous code, 'APSTool' provides a means of visualizing both the high- and low-level design of code and a ready means of selecting and accessing the code for given elements. APSTool is a Windows-based FORTRAN source code editor that provides a flexible dual-presentation of the program structure tree and the source code of a selected subroutine. Code appears in the editor window following selection by clicking on that element in the tree window. The tree is updated automatically as new code is developed.

A novel aid to code development is provided by the test bed generator, 'APSTest'. APSTest analyses selected code for variables used and writes a driving routine that reads inputs and produces an output file. The APSIM graphics utility enables output to be readily graphed and the behaviour of mathematical functions observed.

The test bed created by APSTest is also used to trap logical and coding errors in the source code. The aim is to make it crash or produce incorrect results by using very wide ranging inputs. The test bed reads test input files and generates output that can be evaluated for errors in mathematics, extrapolation of equations, interactions between equations and processing order.

The validation and calibration operations require iterative sequences of code development, model configuration, model runs and analysis of model output. Selection of the Editor and commands to compile, compile and link, and run the model are all on a pull-down menu. Flexible output control routines are provided to construct two-way tables of

observed experimental data and simulation data output and to graphically display or print observed-predicted comparisons and pertinent mathematical statistics.

Porting to other platforms

Because of close adherence to standard FORTRAN 77, other computing platforms can be used to gain greater speed in large jobs without sacrificing much of the convenience of the Windows-based interface. This is accomplished by configuring the model for the production system using the APSIM Shell on the microcomputer then selecting the platform (DOS or UNIX) from the list box (Fig. 2 indicates Silicon Graphics UNIX).

Investment in control of software quality

A major investment has been made in software quality and in systems for maintaining quality as APSIM evolves. We are attempting to implement the software configuration and quality control management system described by Bershoff and Davis (1991). This is based on careful design, coding style standards, moderation of code by peers, testing of code by other programmers, change requests, version control and staged, scheduled development and releases of new versions.

Of the investments in software quality, the following deserve special comment:

Modules designed for independence and simplicity

First, the program is decomposed into a large set of small, highly independent, closed subroutines which can be called from any other subroutine and which can be separately compiled. Second, subroutine independence is maximized by minimizing the data relationships between modules and maximizing relationships within subroutines. Third, subroutines are defined so that each performs one function, and fourth, modules are kept small to aid independence, readability and ease of testing. Additionally, state variables are only changed in a single high level subroutine, with the lower level subroutines returning the rates of change. This reduces problems of order of processing.

Maintainability

Maintainability refers directly to the quality of the code and the documentation. All code is written so that the logic of the program is easily followed by another programmer. Structuring, disaggregation of program modules, use of meaningful variable and subroutine names, and meaningful clarifying comments in the code contribute to the readability

and thus the maintainability of the program. Standard documentation is used for all subroutine header templates and includes descriptions and units in the declaration of all variables/arrays.

Flexibility

Not only are modules easily added, removed and interchanged, but so are subroutines within modules.

Re-usability

To capitalize on past achievements in model development, select current models/sub-models are reverse-engineered and then redesigned and recoded into functional subroutines using the design philosophies described above. The redesign strategies are mainly those described by Elshoff and Marcotty (1982). A redesigned growth routine serves as a template for other crops when a similar level of elaboration of processes is wanted. This means that models of other crops are based on an existing well-designed and -tested code, thus reducing the potential for errors, and time taken for design, coding, debugging and testing. Familiarity of design, functional units and names facilitates recoding and maintenance. But this approach does not restrict the addition, replacement, or deletion of code needed for the development of a growth routine for a new crop.

The biological and environmental modules

Modules for two approaches to simulating soil water are available. *Soil-Wat* (Fig. 1) derives from the multiple store, cascading overflow, water balance routines in CERES Maize (Jones & Kiniry, 1986) and PERFECT (Littleboy *et al.*, 1992). Changes include

- (1) transfer of solute leaching from the soil N routine;
- (2) inclusion of a surface residue effect on evaporation and runoff.

The code for *Soil-Wat* has been redesigned and reverse-engineered to meet programming standards described above.

Recent advances in computing speed and power and in numerical methods for solving non-linear equations now make it computationally-feasible to use a physical process approach to water balance. From the main menu, a user can select APSWIM, a custom implementation of SWIM (Ross, 1990a), which efficiently solves the Richards' equation. (Williams *et al.*, 1991; Ross, 1990b). This approach introduces a substantial change in soil parameterization, which contributes to present lack of clarity of the relative cost-effectiveness of these two approaches for various applications (Williams *et al.*, 1991). APSIM enables these contrasting approaches to be compared readily.

SoilN (Fig. 1) derives from the nitrogen routine in the CERES models (Godwin & Jones, 1992). In addition to complete reverse-engineering to achieve the required structure and modularity, the main changes are the introduction of a labile soil organic matter pool and explicit carbon flows which govern nitrogen flows by C–N ratios.

The surface residue dynamics are simulated in a *Residue* module. All above-ground material is considered as a single pool which can be burnt, incorporated into soil as FOM, or left to decompose on the surface. This routine contains functions for

(a) decomposition of material in contact with the soil, as influenced by tillage and water supply, temperature, mineral nitrogen status of the surface layer of soil and C/N of residue;

(b) relationships between cover and dry matter;

(c) effects of tillage on dry matter placement and on surface cover.

Daily cover is provided to *Erosion* and to the operative soil water module. Values of carbon and nitrogen are transferred to soil pools of the nitrogen module.

The crop template results from combining, redesigning, and re-engineering of AUSIM-Maize (Carberry & Abrecht, 1991), a derivative of CERES-Maize (Jones & Kiniry, 1986) and QSUN (Chapman *et al.*, 1993), similar

TABLE 1
List of Current APSIM Modules and Their Origins

Group	APSIM module	Original model	Reference
Crop	Cotton ^a	OZCOT	Hearn & Da Rosa, 1985
	Cowpea	ASPI-M-Cowpea	Adiku <i>et al.</i> , 1993
	Maize	AUSIM-Maize	Carberry & Abrecht, 1991
	Peanut	QNUT	Hammer <i>et al.</i> , 1992
	Sorghum	QSORG	Hammer & Muchow, 1991
		AUSIM-Sorghum	Carberry & Abrecht, 1991
	Sunflower	QSUN	Chapman <i>et al.</i> , 1993
	Wheat1	Woodruff-Hammer	Hammer <i>et al.</i> , 1987
	Wheat2	CERES-Wheat	Ritchie <i>et al.</i> , 1988
Tropical grass pasture	GRASP ^a	GRASP	McKeon <i>et al.</i> , 1990
Temperate pasture	GRAZPLAN ^a	GRAZPLAN	Moore <i>et al.</i> , 1991
Soil water	SoilWat	CERES	Ritchie, 1985
		PERFECT	Littleboy <i>et al.</i> , 1992
		APSWIM ^a	Ross, 1990a
Soil nitrogen	SoilN	CERES	Godwin & Jones, 1991
Soil erosion	Erosion	PERFECT	Littleboy <i>et al.</i> , 1989

^aIntellectual property remains that of the original developer.

in concept to the soybean model of Sinclair (1986). Any existing crop model can be installed in APSIM as a module, simply by disabling the soil processes and adding the APSIM interface. But this 'patching' of existing model code into APSIM has obvious shortcomings compared to re-programming using the APSIM crop template, and suffers some limitations in the use of APSTool. However, it is often a justifiable expediency.

The *Erosion* module uses a modified event-based Universal Soil Loss Equation (Littleboy *et al.*, 1992). Parameters are erodibility, slope and slope length. Inputs are daily *runoff* from the water balance module and daily soil *cover* from Residue.

DISCUSSION

APSIM represents a major investment in improving predictive modelling in agricultural systems research which combines Farming Systems Research Methodology and Operational Research (McCown *et al.*, in press). APSIM contributes to better predictive modelling in a number of ways. First, improved representation of certain aspects of cropping systems enables important phenomena to be better simulated, e.g. the effects of crop sequences on soil N and the competition between intercrops. Second, good routines in different models can be easily recombined to provide a superior configuration for a given task. Third, APSIM provides an infrastructure that can support convergent effort by teams in testing and improving models, with change taking place simultaneously on many fronts.

An important feature of APSIM will be a customized relational database presently under development using Microsoft AccessTM. Parameter files for specified crop cultivars and soil taxons, weather files, and files of observed crop and soil data are automatically built by *APSDData* which is accessed using the *APSDData* pull-down menu. In addition, *APSDData* will provide cost and price information to enable an *Economics* module to keep track of cash flows during multi-period production runs. This enables management rules based cash flow to be incorporated in the Manager.

APSIM structure makes simulation of competition between mixed crops conceptually elegant and operationally efficient. Models of the crops to be included are selected at the beginning of the run (Fig. 2). The fact that they share soil water and N modules and that sharing of soil water and N is controlled by the Arbitrator module, code of a crop routine representing an intercrop does not differ from that of a crop growing alone. When appropriate relationships are specified in the Arbitrator, simulations of intercrop performance agree well with that measured (Adiku *et al.*, 1993; Carberry *et al.*, 1994).

Although designed for research on dryland cropping systems, APSIM is now being used for simulating other systems. Pasture and animal production modules are being added: GRAZPLAN (Moore *et al.*, 1991) has been interfaced with APSIM for the Mediterranean and temperate regions of Australia and the pasture model, GRASP (McKeon *et al.*, 1990) is now a module in APSIM for use in the subtropics and tropics. Modules are being developed and modified to adapt APSIM for sugar cane production systems in high rainfall areas. APSIM is also being used in research aimed at improving design of liquid waste disposal systems. This includes simulation of water and N cycling in forestry plantations with a grassy understory, which utilizes the intercropping capability.

In the research-funding climate which has developed in Australia, it has been necessary to adopt a business approach to the development and distribution of APSIM. The investment needed for development of quality software has proved to be very much higher than we anticipated. No longer can such costs be met from appropriation funding, nor will industry funding bodies support such development that seems to many of their stakeholders to be R&D core technology. This means that continuing development and maintenance of APSIM must come largely from generated income. To date, in addition to using APSIM to win industry funding for applications projects, this has taken the form of contracts to support the use of APSIM by others under license, generally in research in which APSRU has a common interest.

While this policy is dictated by economic realities of the developers, it appears to provide net benefits to all concerned. Our collaborators/clients benefit from a product and support whose quality and cost benefits exceed those of alternatives which might be produced in-house. Industry funding bodies see support of a software development effort 'with critical mass' as an efficient means of providing quality modelling support for their projects. They also value the convergence of effort in improving simulation capabilities that is resulting from research networks in which APSIM is the common 'language'. Importantly, this approach to management of the intellectual property aspects of APSIM does not interfere with the publication of the scientific advances that contribute to or emanate from APSIM in equitable arrangements.

ACKNOWLEDGEMENTS

Neil Huth, Brian Wall, Brian Keating, Peter Carberry, Robin Connolly, Holger Meinke and Greg Larkens have made important contributions to the development of APSIM.

REFERENCES

- Adiku, S. K., Carberry P. S., Rose, C. W., McCown, R. L. & Braddock, R. (1993). Assessing the performance of maize (*Zea mays*) — cowpea (*Vigna unguiculata*) intercrop under variable soil and climate conditions in the tropics. *Proceedings of the 7th Australian Society of Agronomy Conference*, September 1993, Adelaide, South Australia, p. 382.
- Bershoff, E. H. & Davis, A. M. (1991). Impacts of life cycle models on software. *Commun. ACM*, **34**(8), 104–18.
- Carberry, P. S., Muchow, R. C. & McCown, R. L. (1989). Testing the CERES-Maize simulation model in a semi-arid tropical environment. *Field Crops Res.*, **20**, 297–315.
- Carberry, P. S. & Abrecht, D. G. (1991). Tailoring crop models to the semi-arid tropics. In *Climatic Risk in Crop Production: Models and Management for the Semi-arid Tropics and Subtropics*, eds R. C. Muchow & J. A. Bellamy. CAB International, Wallingford, pp. 157–82.
- Carberry, P. S., Adiku, S. G. K., McCown, R. L. & Keating, B. A. Application of the APSIM cropping systems model to intercropping systems. In *Proceedings International Workshop on Dynamics of Roots and Nitrogen in Cropping Systems of the Semi-Arid Tropics*, ICRISAT. (In press).
- Chapman, S. C., Hammer, G. L. & Meinke, H. (1993). A sunflower simulation model: I. Model development. *Agron. J.*, **85**, 725–35.
- Checkland, P. B. (1981). Rethinking a systems approach. *J. Appl. Syst. Anal.*, **8**, 3–14.
- Cole, C. V., Williams, J., Shaffer, M. & Hanson, J. (1987). Nutrient and organic matter dynamics as components of agricultural production systems models. *Soil Fertility and Organic Matter as Critical Components of Production System*, SSSA Spec. Pub., **19**, 147–66.
- Elshoff, J. L. and Marcotty, M. (1982). Improving computer program readability to aid modification. *Communications of the ACM*, **25**(8), 512–21.
- Freebairn, D. M., Silburn, D. M. & Loch, R. J. (1989). Evaluation of three soil erosion models for clay soils. *Aust. J. Soil Res.*, **27**, 199–211.
- Godwin, D. C. & Jones, C. A. (1992). Nitrogen dynamics in soil-plant systems. In *Modeling Plant and Soil Systems, Agronomy Monograph No. 31.*, eds J. T. Richie and J. Hanks. ASA-CSSA-SSSA, Madison, pp. 287–321.
- Hamilton, W. D., Woodruff, D. R. & Jamieson, A. M. (1991). Role of computer-based decision aids in farm decision making and in agricultural extension. In *Climatic Risk in Crop Production: Models and Management for the Semi-arid Tropics and Subtropics*, eds R. C. Muchow & J. A. Bellamy. CAB International, Wallingford, pp. 411–24.
- Hammer, G. L. & Muchow, R. C. (1991). Quantifying climatic risk to sorghum in Australia's semiarid tropics and subtropics: model development and simulation. In *Climatic Risk in Crop Production: Models and Management for the Semi-arid Tropics and Subtropics*, eds R. C. Muchow & J. A. Bellamy. Ch. 16. Wallingford, CAB International, pp. 205–32.
- Hammer, G. L., Woodruff, D. R. & Robinson, J. B. (1987). Effects of climatic variability and possible climatic change on reliability of wheat cropping — a modelling approach. *Agric. For. Meteorol.*, **41**, 123–42.
- Hammer, G. L., Irmansyah, Meinke, H., Wright, G. C. & Bell, M. J. (1992). Development of a peanut growth model to assist in integrating knowledge

- from management and adaptation studies. In *Peanut Improvement: a Case Study in Indonesia*, ed. G. C. Wright & K. J. Middleton. *ACIAR Proceedings No. 40*, pp. 95–103.
- Hargreaves, J. N. G. & McCown, R. L. (1988) A Visual-interactive version of CERES-Maize. *Tropical Agronomy Technical Memorandum*, CSIRO Division of Tropical Crops and Pastures, Brisbane, Australia.
- Hearn, A. B. & Da Rosa, D. A. (1985). A simple model for crop management applications for cotton. *Field Crops Res.*, **12**, 46–69.
- Hodges, T., Johnson, S. L. & Johnson, B. (1992). A modular structure for crop simulation models: implemented in the SIMPOTATO model. *Agron. J.*, **84**, 911–15.
- Hoogenboom, G., White, J. W., Jones, J. W. & Boote, K. J. (1994). BEAN-GRO: a process-oriented dry bean model with a versatile user interface. *Agron. J.*, **86**, 182–90.
- Hunt, L. A., Pararajasingham, S., Jones, J. W., Hoogenboom, G., Imamura, D. T. & Ogoshi, R. M. (1993). GENCALC: software to facilitate the use of crop models for analyzing field experiments. *Agron. J.*, **85**, 1090–4.
- Jones, C. A. & Kiniry, J. R. (1986). *CERES-Maize: A Simulation Model of Maize Growth and Development*. Texas A&M University Press, College Station, Texas, 194 pp.
- Keating, B. A., Godwin, D. C. & Watiki, J. M. (1991). Optimising nitrogen inputs in response to climatic risk. In *Climatic Risk in Crop Production: Models and Management for the Semi-Arid Tropics and Subtropics*, eds R. C. Muchow & J. A. Bellamy. CAB International, Wallingford, pp. 329–58.
- Keating, B. A., McCown, R. L. & Wafula, B. M. (1993). Adjustment of nitrogen inputs in response to a seasonal forecast in a region of high climatic risk. In *Systems Approaches for Agricultural Development*, eds F. W. T. Penning de Vries *et al.* Kluwer Academic Publishers The Netherlands, pp. 233–52.
- Littleboy, M., Silburn, D. M., Freebairn, D. M., Woodruff, D. R., Hammer, G. L. & Leslie, J. K. (1992). Impact of soil erosion on production in cropping systems. I. Development and validation of a simulation model. *Aust. J. Soil Res.*, **30**, 757–74.
- Loomis, R. S. (1985). Systems approaches for crop and pasture research. *Proceedings of the 3rd Australian Society of Agronomy Conference, 1985*, Hobart, Tasmania, pp. 1–8.
- McCown, R. L. and Williams, J. (1989). AUSIM: a cropping systems model for operational research. *Proc. SSA IMACS 1989 Biennial Conference on Modelling and Simulation*, ANU, 25–27 Sept.
- McCown, R. L., Keating, B. A., Probert, M. E. & Jones, R. K. (1992). Strategies for sustainable crop production in semi-arid Africa. *Outlook on Agric.*, **21**(1), 21–31.
- McCown, R. L., Cox, P. G., Keating, B. A., Hammer, G. L., Carberry, P. S., Probert, M. E. & Freebairn, D. M. The development of strategies for improved agricultural systems and land use management. In *Proceedings of ISNAR/ICASA Workshop, Opportunities for Use of Systems Research Methods in Agriculture in Developing Countries*. The Hague, November 22–25, 1993. (In press).
- McKeon, G. M., Day, K. A., Howden, S. M., Mott, J. J., Orr, D. M., Scattini, W. J. & Weston, E. J., (1990). Management for pastoral production in northern Australian savannas. *J. Biogeog.*, **17**, 355–72.

- Moore, A. D., Donnelly, J. R. & Freer, M. (1991). GRAZPLAN: an Australian DSS for enterprises based on grazed pastures. *Proceedings International Conference on Decision Support Systems for Resource Management*, Texas A&M University, College Station, Texas, USA, April 15–18.
- Ritchie, J. T. (1985). A user-orientated model of the soil water balance in wheat. In *Wheat Growth and Modeling*, eds E. Fry & T. K. Atkins. Plenum Publishing, NATO-ASI Series, pp. 293–306.
- Ritchie, J. T., Godwin, D. C. & Otter-Nacke, S. (1988). *CERES-Wheat: a Simulation Model of Wheat Growth and Development*. Texas A&M University Press, College Station, Texas.
- Ross, P. J. (1990a). *SWIM: A Simulation Model for Soil Water Infiltration and Movement*. Reference Manual. CSIRO, Australia, Division of Soils, Townsville.
- Ross, P. J. (1990b). Efficient numerical methods for infiltration using Richards' equation. *Water Resources Res.*, **26**(2), 279–90.
- Seligman, N. G. (1990). The crop model record: promise or poor show? In *Theoretical Production Ecology: Reflections and Prospects*, eds R. Rabbinge, J. Goudriaan, H. van Keulen, F. W. T. Penning de Vries & H. H. van Laar. PUDOC, Wageningen, pp. 249–58.
- Sinclair, T. R. (1986). Water and nitrogen limitations in soybean grain production. I. Model development. *Field Crops Res.*, **15**, 125–41.
- Steiner, J. L., Williams, J. R. & Jones, O. R. (1987). Evaluation of the EPIC simulation model using a dryland wheat-sorghum-fallow crop rotation. *Agron. J.*, **79**, 732–8.
- Thornton, P. K. & McGregor, M. J. (1988) The identification of optimum management regimes for agricultural crop enterprises. *Outlook on Agric.*, **17**, 158–62.
- Uehara, G. & Tsuji, G. Y. (1991). Progress in crop modelling in the IBSNAT project. In *Climatic Risk in Crop Production: Models and Management for the Semi-arid Tropics and Subtropics*, eds R. C. Muchow & J. A. Bellamy. CAB International, Wallingford, pp. 143–56.
- Wilkerson, G. G., Jones, J. W., Boote, K. J., Ingram, K. T. & Mishoe, J. W. (1983). Modelling soybean growth for crop management. *Trans. ASAE*, **26**, 63–73.
- Williams, J. R., Jones, C. A., Kiniry, J. R. & Spaniel, D. A. (1989). The EPIC crop growth model. *Trans. ASAE*, **32**(2), 497–511.
- Williams, J., Ross, P. J. & Bristow, K. L. (1991). Parsimony, precision and pragmatism in modelling crop water supply. In *Climatic Risk in Crop Production: Models and Management for the Semi-arid Tropics and Subtropics*, eds R. C. Muchow & J. A. Bellamy. CAB International, Wallingford, pp. 73–96.