18th World IMACS / MODSIM Congress, Cairns, Australia 13-17 July 2009
http://mssanz.org.au/modsim09

Simulation of whole farm management decisions

de Voil, P.G.[1], D. Rodriguez[1], B. Power[1], and W.A.H. Rossing[2]

[1] Queensland Department of Primary Industries & Fisheries, 203 Tor St, Toowoomba, QLD 4350.
[2] Wageningen University, Biological Farming Systems Group.
Corresponding email: Peter.deVoil@dpi.qld.gov.au

Abstract: The simulation model APSFarm is an extended configuration of the systems model APSIM, and has been in development since 2005 to support the analysis of farm case studies in environments ranging from Central Queensland to the Victorian Riverina. One of the major extensions to APSIM was the implementation of dynamic farm management as a set of state transition networks, with each network representing the operation of a particular paddock or management unit. Each paddock has a current state (eg. fallow, crop), "rules" that allow transition to adjacent states, and "actions" that are taken when such a transition is made. These rules represent feasibility (eg. whether it is the correct planting season for this particular crop, whether machinery is available), tactics (eg fertiliser management), and strategy (eg. crop sequencing and mix of enterprise). Each day, the model examines all paths leading away from the current state to adjacent states. Should the mathematical product of all rules associated with a path be non-zero, the path becomes a candidate for action. Should more than one candidate be present, the highest ranking path is chosen, and its associated management actions to change state are undertaken (eg. sowing or harvesting a crop). There may be cascading events that flow from a state change, so the process is repeated until nothing more can be done for that day.

The benefits of the network approach are multifold; gross patterns can be identified from the network structure that allows comparison between farmers and farm types. From a software perspective, details of the transition rules and their associated actions are readily accessible, instead of being encoded in complex logic code constructs. Encapsulating management strategy as a data structure instead of a series of specific instructions allows the construction of dynamic analysis tools in which complex decisions can be more clearly described as sets of simple and measurable rules of thumb. Further, this data structure can make the task of providing a cohesive user interface much easier, facilitating graphical representations of farm management.

This paper presents a software tool used in studying diverse management systems drawn from ongoing case studies, discusses how the management systems are elucidated from farm managers and demonstrates some of the diagnostics available from such tools.

Keywords: *Simulation, APSIM, Management Strategy, APSFarm, state transition network.*

1. Introduction

Agriculturalists make decisions under risky and uncertain conditions – droughts, floods, changing markets, and new government policies. Primarily, a farmer's tactics and management strategy to deal with such uncertainty is developed through years of experience and practise, known as "gut feeling" or intuition (Schwartz and Sharpe, 2007). Unfortunately, intuition may be of little use when farmers are exposed to new operational environments, for example climate change, which is expected to add new dimensions to this already complex problem of managing a farm business.

Simulation models are valuable tools in the exploration of alternative farm management strategies. They allow alternative strategies to be evaluated without risk of negative physical on-farm impact, and provided the model is built on sound biophysical processes, it is possible to extrapolate the behavior of these management strategies under change scenarios. Our requirement to model management strategy *in silico* does not entail a model of human decision making processes, it involves reducing real world complexity to a simple level; balancing the need for reduced complexity while remaining responsive to the multitude of factors affecting farm management.

A farm management strategy can be described as the set of planned tasks that incorporates capabilities to adapt to perturbations (Cros et al, 2004), be they in the physical environment, or markets. While a simple "linear" (or calendar based) approach to representing farm management provides a readily understood framework, the method quickly becomes unwieldy in practice due to the opportunistic and adaptive nature of farming enterprises. Alternatively, management tasks can be aligned in a hierarchy whose structure is dictated by consequential relations between subtasks, a structure which closely relates to how individual tasks have temporal horizons. This concept of "management as a network" has significant advantages, in that portions of the network can be examined in isolation with known boundaries.

A series of project activities between researchers and farming practitioners (farmers, extension and advisory staff) that have used whole farm models has now been expanded to four locations in Eastern Australia. These activities take place as a series of workshops, using locally relevant case studies as discussion material, with a specific focus on long term adaptation strategies. The case study material from these workshops will explore questions of farm management strategy using a simulation model to provide insight on system wide interactions at these new levels.

2. From APSIM to APSFarm

APSIM, the Agricultural Production System sIMulator (Keating et al, 2003), was derived from the CERES (Jones and Kinry, 1986) family of models, which at the time were primarily focused on replicating experimental plots in a "single year, single crop" manner. The need to scale up to production systems required the introduction of short term processes such as fallow nutrient cycling via a surface residue pool. Simulation of longer term processes such as the effect of soil erosion on soil fertility was then possible. The advances in science were accompanied by advances in software engineering, particularly modular construction, which allowed users to "plug in" the necessary components of their particular system.

However, implementing the seemingly mundane tasks of managing crop establishment, harvest and death led to a manager component with capabilities that are "far beyond anything envisaged by the early developers" (Keating et al, 2003). The component provided a basic-like language interpreter, which implemented conditional logic driven by state variables (collected from other modules plugged into the system) and local variables. This simple language was easily understood by non-programmers, supporting the everyday simulation of field scale crop rotations, intercropping, and even whole farm systems (Gaydon et al, 2006).

While these applications demonstrate the capacity of a simple interpreter to undertake such tasks, its limitations are apparent. The only datatypes are scalar strings and real numbers. There are no flow control structures beyond if/then/else structures, and there is no ability to define procedures or function calls. Importantly, the lack of comprehensive error detection in its language parser (a table driven parser written in fortran) makes developing reliable simulations risky even for experienced users. Since the implementation of this interpreter in 1992, advanced language tools that are able to reliably parse text and produce syntax trees for subsequent evaluation have become available (Parr, 1995), yet the time constraints involved in rewriting the module (to one of C++/ Java/ C#) to use such techniques have been prohibitive.

Within the APSIM user community, the ongoing use of TCL (Ousterhout, 1998) as a tool for managing simulations in a "batch" manner led to the realisation that the same language could be embedded as a module within the APSIM framework; able to interact with the simulation framework in the same manner as the earlier manager component, yet offering a richer, more expressive language able to undertake more demanding tasks. The developers of OpenALEA (Pradal et al, 2008), went further, embedding a set of modules for structural plant modeling within a Python interpreter, such that a simulation is a python program. Significantly, both of these approaches avoid a considerable amount of software maintenance, the core language implementation is managed by the respective user communities, and in each language there exists a large amount of library code to support scientific application development. Each user community distributes under "open source" style licenses.

The use of a rich language allows us to reduce a complex set of if/then rules to a single data structure, representing the operation of the farm as a connected set of states (eg. particular crops), and actions that are undertaken when we wish to change a state (eg. sow a crop). The management strategy is implicit in the rules that allow a state transition to be made. The management process becomes a generic procedure that operates on the data structure.

The price is that these more advanced languages have a more advanced syntax. Code written in TCL is not as readily understood as the original "basic like" language, leaving non-specialists with a considerable barrier to understand and manipulate these more complex simulations.

3. Implementation of APSFarm

APSFarm extends APSIM in 3 areas: the use of multiple paddocks in a single simulation, the implementation of a cashbook module responsible for maintaining a farm's economic state, and the use of farm level management rules that specify resource sharing at the farm level. Multiple paddocks were introduced to APSIM since version 3, though communication between modules at the farm level was static (ie. pathways defined before the simulation commenced) until version 6. The cashbook module simply subscribes to economic events coming from the system, when they arrive, it uses its own tables to find the associated cost/price information, allowing a cash balance to be kept.

A complaint of many systems approaches is that the "plumbing" involved in a component communicating with the system obscures the function of the component itself. In recent years, attempts have been made to "hide" this detail within component wrappers (Holzworth et al, 2007) and language extensions (Holzworth and Huth, *these proceedings*) that provide a simple interface to the rest of the system via four paths: getting and setting variables in components plugged into the system, publishing and subscribing to events in the system. The "basic like" manager in APSIM further hides plumbing by lazy variable scoping: if a variable requested by the user is not available from the APSIM infrastructure, it is assumed to be a local variable of the manager itself.

The "farm manager" used within APSFarm implements these same four interface points as TCL commands. Thus, the user of this module is able to inquire about states in the system, make decisions based on those states, and monitor or initiate events in the system. Interpretations can be made about the state of other components: whether a crop is alive or dead, whether the current date of the simulation is within a window, whether soil water is above some threshold. From these interpretations, the language can implement complex data structures that represent "meta" states, or compounds of physical states and their relations, for example a "would probably plant a late crop" state can be composed of "it is late in the season, few crops are planted in the farm, there is a positive seasonal climate forecast" states.

The use of such structures allows us to reduce the existing set of complex if/then rules to a single data structure, representing the operation of the farm as a connected set of states (eg. particular crops in particular paddocks), and actions that are undertaken when we wish to change a state (eg. sow a crop). The management strategy is implicit in the rules that allow a state transition to be made.

Implementing management rules in APSIM has been quite straightforward, seldom requiring more than 1 page of code for simple tasks. When moving to whole-farm simulations in APSFarm, the amount of code required to represent management rules for all crops and for the use of farm-resources grew to 30Kb (7 pages) or more, yet has two distinct parts: generic "used everywhere" support procedures and specific farm configurations, such as crop varieties, fertiliser rates, and the combined rules and actions described above. For non-specialist users, a GUI can present the farm specific information with contextual help.

4. Graphical Representation

APSFarm uses the APSIM User Interface (ApsimUI), which stores the entire simulation as an XML tree – and presents the tree as a way of navigating between components of the simulation. Each component of the simulation has its own UI tab. Most "science" components access their parameters as name/value pairs. The parameters (or "rules") of the manager are plain text nodes, each associated with a simulation event. If or when the event occurs during the simulation, the system will invoke the manager with this text.

Several UI tabs are available for the manager component. The simplest UI tab presents rules as plain text with one text node for each event subscription. More advanced UI tabs present the user with alternative views of the underlying code, while still presenting text (rules) to the model. For the management networks outlined above, an editable connected graph is displayed (figure 1). The location and contents of the nodes and arcs are explicit in the components XML tree. When the UI is asked to "save" itself it saves not only the nodes and arcs but also a TCL rule to build the structure in a simulation, which is read by the model at runtime.

5. Case studies

The use of APSFarm in case studies follows an iterative process of engagement between model builders and farming practitioners. Over a series of interactions, the farm physical properties (area, soil types, water and nutrient characteristics), crop and livestock components, work and resource capacities (labour/machinery capacities), agronomic "rules", and operational preferences are described. Each step represents a confirmation of the previous steps, a confidence building process whereby model builders gain insights into the details of the farm, and practitioners see that the model can reproduce their direct experience. As the exercise is not to mimic "human decision making", but to explore strategy, we do not expect the model to predict the exact operation of the case study, but to demonstrate "similar behavior", typically measured by similar cropping intensities and seasonal responses.

The farm level management processes of two fundamentally different farming enterprises are presented here demonstrating very different strategies: a rainfed grain production system from Central Queensland, and a conventional irrigated cotton/grain system from Southern Queensland.

Rainfed grain production systems in Central Queensland are characterised by their highly opportunistic nature, due largely to the extreme climatic variability of the region: "When it rains, do something" was a simple strategy from one participant. In brief, crops are grown in summer and winter, with a preference for summer crops. Double cropping is rare, if unheard of. "Best practise" avoids planting the same crop year after year, as the incidence of disease increases with each subsequent planting. Provided machinery is available, crops are planted when rainfall events fill the soil surface layer/s with enough water to ensure germination. Should these criteria be satisfied, the final impediment to planting is the farmers attitude to risk: given the high chance of an unfavourable season, the chances of financial loss from crop failure are likewise high. If a sizeable fraction of the farm is planted to a crop at any one time, the consequences are serious, hence the farmer is reluctant to "go all out" to a single crop. Discussions with farmers indicate these attitudes are known as conservative (risk-averse) or aggressive (risky) behavior, and there was much interest in quantifying the impact of both behaviors. The quantification of this "attitude" is simple: an upper limit to the area planted for a particular crop.

After several rounds of discussion, APSFarm was able to provide insight into these (and other) questions about management strategies. Invariably, the model provides a wealth of discussion support material, usually in the form of tradeoffs (eg "more income, higher risk"), and sometimes as rules of thumb, for example the chance of some particular event occurring. Current discussions are examining the impact of climate change projections on whole farm productivity.

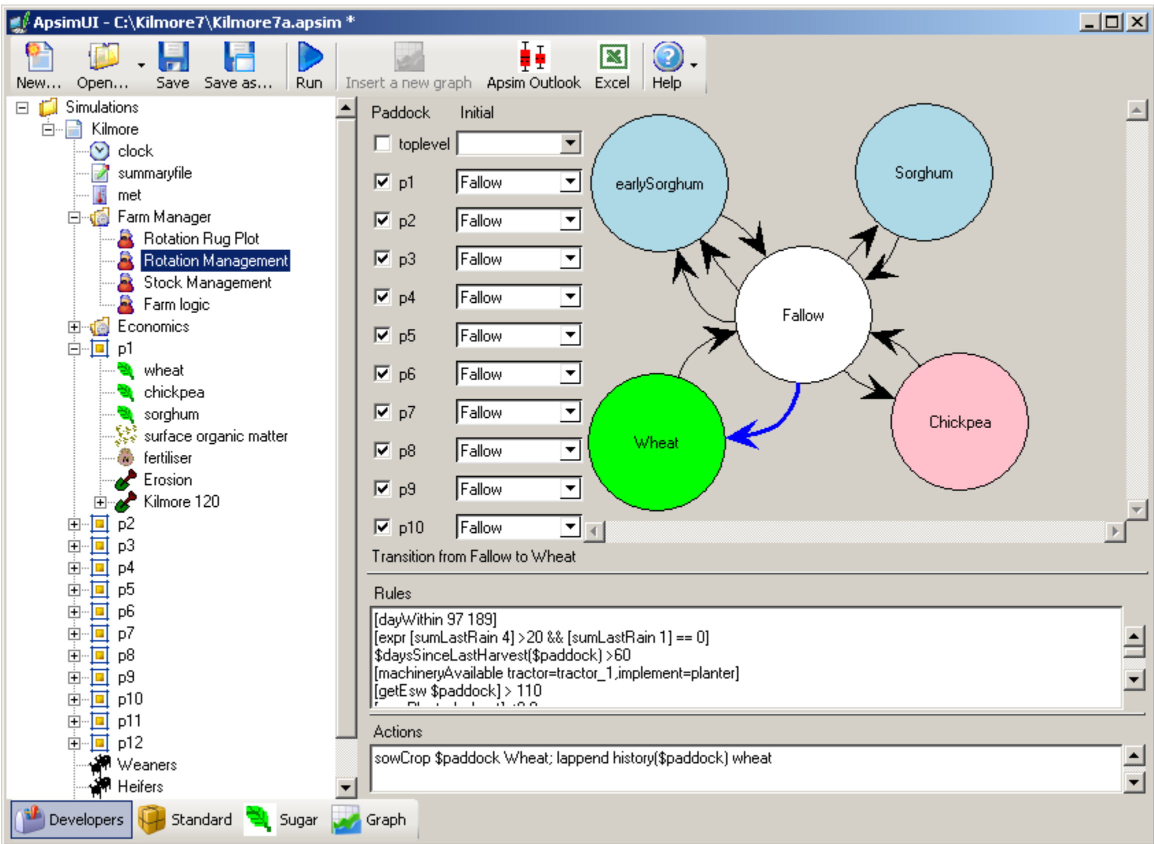

Figure 1. ApsimUI screenshot of the CQ case study.

A screenshot of a simulation from this case study is shown in figure 1 – the components of the simulation are shown in the left hand tree, with the crop management component highlighted. The initial state of each paddock is described, and the rules and actions for a wheat planting shown. The transition network for this system is star-shaped, reflecting the opportunistic nature of the system.

Irrigated systems present a reverse scenario; ordered crop rotations that center around the need to utilise a limited amount of water in the most efficient manner. Unlike rainfed systems, the planting requirements for irrigated systems are largely temperature driven –for both seed germination, and the need to avoid seasonal temperature extremes during sensitive phases of the crop. Topics of discussion have included the tradeoffs of introducing rainfed crops into an irrigated farm when irrigation water is in short supply, quantification of the value of efficiency measures that reduce evaporative and drainage losses, and relations between prices that make one crop more profitable than another.

A case study example was developed for an 800 ha farm in Southern Queensland reliant on a combination of overland flow and bore water for irrigation, under economic pressure from a declining bore water allocation. Over several interviews, a management strategy was elucidated that described farmer rules for planting individual crops. The rules describing crop sequences are shown in Figure 2, rules describing resource requirements for each crop are itemized:

Irrigated Maize

- Planting Window: Oct 1 – Oct 15
- Dam storage water greater than 80%
- Soil water greater than 150 mm
- Existing farm area planted to maize or sorghum less than 50%

Rainfed Sorghum

- Planting Window: Oct 16 – Dec 15
- Last 4 days rain greater than 25
- Soil water greater than 100 mm
- Existing farm area planted to maize or sorghum less than 50%

or, late in season

- Planting Window: Dec 16 – Jan 14
- Existing farm area planted to maize or sorghum less than 25%

Irrigated Cotton

- Planting Window: Oct 16 – Oct 31
- Dam storage water greater than 50%
- Existing farm area planted to cotton less than 50%
- Existing farm area planted to maize or sorghum less than 50%

Rainfed Wheat

- Planting Window: Apr 16 – Jul 1
- Last 10 days rain greater than 25
- Soil water greater than 100 mm
- Days since last harvest greater than 14 days

The key feature of this strategy is its flexibility in response to water availability, be that moisture stored in the soil for rainfed crops, or water stored in dams for irrigated crops. In discussions, the flexibility of the system has shown a maintenance of profitability under change scenarios, though with considerably reduced income.

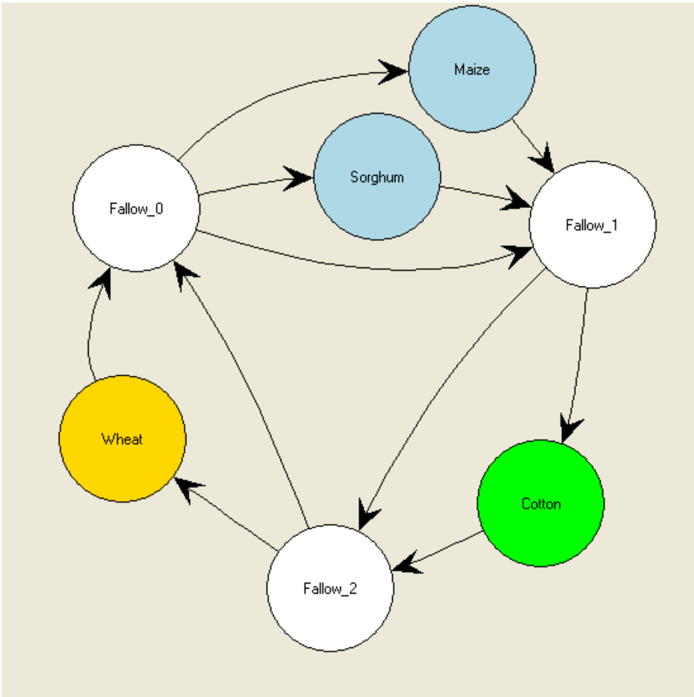

Figure 2. Crop sequencing diagram from the Southern Queensland case study.

6. Discussion

Practitioners find it difficult to talk about their motives for decisions. Researchers find it difficult to fully understand the drivers of farm decisions. Here a whole-farm model is described for use in comparing decision strategies. "Parameters" of the model closely match the topics of discussion – both the physical characteristics of the farm and the farmer's attitude to risk, and it can simulate the impact of change in terms that are easy to discuss with farmers, such as yield distributions, risks and environmental factors.

While the process of building and parameterising the model is lengthy, it integrates a diverse range of sciences, allowing exploration of the combined interactions of system components. This systemic behavior can be counter–intuitive; for example one discussion in the Central Queensland case study concerned an estimation of predicted changes in sorghum yields in a climate change situation; in a monoculture situation, the effect of CO_2 fertilisation outweighs the reduced water supply, increasing yields. Yet in system simulations, it was found that the number of opportunities to plant sorghum became severely limited, and that the conditions for that planting would be quite unfavourable.

The APSFarm model remains a complex research tool that requires skill to operate, not unlike APSIM. Like APSIM, the majority of its users come from the research community,who use the model to support discussions with agricultural practitioners. Such discussions have taken place and have been valued by participants, and more are in progress. The APSFarm model has been an invaluable component in these activities, both to research participants as a container of knowledge, and to practitioners as a source of reliable information.

References

Cros M.J., Duru M., Garcia F., and Martin-Clouaire R. (2004), Simulating Management Strategies: the Rotational Grazing Example. *Agricultural Systems* 80:23-42.

Gaydon D.S., Lisson S.N., and Xevi E. (2006), Application of APSIM 'multi-paddock' to estimate whole-of-farm water-use efficiency, system water balance and crop production for a rice-based operation in the Coleambally Irrigation District, NSW, Proceedings of the 13th Australian Society of Agronomy Conference, 10-14 September 2006, Perth, Western Australia.

Holzworth D.P., Huth N.I., and DeVoil P. (2007), A Framework Independent Component Design: Keeping It Simple. In Oxley, L. and Kulasiri, D. (eds) MODSIM 2007 International Congress on Modelling and Simulation. Modelling and Simulation Society of Australia and New Zealand, December 2007.

Holzworth, D. and Huth, N. (2009), Reflection + XML Simplifies the Development of the APSIM Generic PLANT Model. MODSIM 2009 International Congress on Modelling and Simulation. Modelling and Simulation Society of Australia and New Zealand, July 2009.

Keating B.A., Carberry P.S., et al. (2003), An overview of APSIM, a model designed for farming systems simulation. *European Journal of Agronomy* 18, 267-288.

Jones, C.A. and Kinry, J.R. (1986), CERES-Maize: A Simulation Model of Maize Growth and Development. Texas A&M University Press.

Ousterhout, J.K. (1998), Scripting: higher-level programming for the 21st century. *Computer* 31:23–30.

Parr T. (1995) ANTLR - A Predicated ll(k) Parser Generator. *Software - Practice & Experience* 25:789-810 (http://antlr.org)

Pradal, C., Dufour-Kowalski, S., Boudon, F., Fournier, C., and Godin, C. (2008), OpenAlea: A visual programming and component-based software platform for plant modeling, *Functional Plant Biology* 35:751-760

Schwartz B., and Sharpe, K. (2006), Practical wisdom: Aristotle meets positive psychology. *Journal of Happiness Studies* 7:377-395