December 2002

Design environments for Intelligent Systems

Don Kerr
Griffith University

John Gammack
Griffith University

Design Environments for Intelligent Systems

Don Kerr

John Gammack

School of Management
Griffith University
Nathan, Queensland, Australia
d.kerr@mailbox.gu.edu.au

Abstract

There is a need to balance the quality of professionally designed information systems with the end user's current knowledge of specific decision contexts. This is particularly so for intelligent systems. This paper looks at some theoretical underpinnings for the potential end-user development of intelligent systems. General requirements are characterised and the metaphor of a semantic spreadsheet is introduced. A two level process enabling end user development of knowledge-based systems is described. The first involves the development of a design environment that allows experts to develop the knowledge base. The second involves development within the design environment for the ultimate end users.

Keywords

GB0404 End users, UF Knowledge-based systems, GA0301 End-user programming, HA03 Decision Support Systems

INTRODUCTION

In their prospective for the next decade of decision support systems research, Carlsson and Turban (2002) note that "expert systems technology is now being replaced by a diversity of intelligent systems, which are built to fulfill … the support of an effective and productive use of information systems, which quite often are tailored to the needs and personality of the user." The development of systems by and for end users has seen much research activity, but the development of intelligent knowledge based systems by end users is as yet in its infancy. As Wagner (2000:12) asks, "Which new technology will enable end users to become knowledge system developers?" This paper addresses the prospects for such a technology.

Traditionally information systems development has required third party analyst/ developers identifying requirements and specifying a design for a system with particular functionality. Even with methodologies that embrace increased user participation in systems design, and considerable improvement in the technologies for and capabilities of end users, intermediaries are often necessary to the process. In the case of intelligent systems, such as expert and knowledge based systems, development has hitherto generally required a third-party knowledge engineer attempting to understand the domain knowledge and logically to order it for decision-making. Although a few successful end user expert systems are reported (Wagner, 2000), neither development difficulties faced, nor the extra complexity involved compared to spreadsheet developments, are described sufficiently to allow understanding of the potentials and limits in this area. Systems developed using expert system shells do not fully overcome key limitations, including content, size, quality and scalability (Wagner, 2000), which suggests a professional development effort is required for significant systems given current technologies. A general need for user-friendly decision support systems has also been noted (Kerr, 1996; Cox, 1996), and our research aims at identifying the design requirements for a technology that allows end user knowledge system development.

Although most end user development research has concerned spreadsheets as a representative decision support system, there have already been some attempts to involve end users in developing knowledge based systems. An early approach was TEIRESIAS (Davis and Lenat, 1982), a tool which took a developed rule based system and, for particular cases, interacted with the user (domain experts) to analyse and refine the conclusions. Apart from being limited to the cases considered, the knowledge representation was already fixed, and this line of work has not been pursued. Kreie *et al.* (2000) noted the problems with end user developed application quality, and introduced training in systems development for end users in a laboratory study. Although this increased the overall application design quality, other measures of quality showed no effect, and the study acknowledges sample and external validity limitations. Similarly Janvrin and Morrison (2000) in another experimental study identify risks associated with spreadsheet errors, including developer inexperience, poor design approaches, application types, problem complexity, time pressure, and presence or absence of review procedures. They compare using a structured design approach for spreadsheet development to *ad hoc* design, and show it has a positive effect on reducing errors. Wagner (2000) explicitly examined 25 expert system developments by MBA students following 15 hours of relevant instruction, a duration comparable with industry short courses, and unlikely to be less than companies might offer non-professional IS developers. Yet significant quality limitations were observed for this group too.

Yoon, Guimaraes and O'Neal (1995) identify the determinants of knowledge based system success, namely quality of developers, end user characteristics and involvement of end users in the development. Amongst several specific factors, higher success was associated with greater involvement of users, with highly significant correlations shown between assessments of satisfaction, reliability, timeliness and output value, and involvement at the level of designing those outputs. Although a small sample and single company study, there is no reason to suppose these do not generally apply. However, the applicable success measures used do not concern actual usage of developed systems, and figures on these are important when considering investments in specific developments, especially when such developments occur for wider deployment across an industry.

For example, in an agricultural context, Hayman and Easdown (2002) observe that decision support system (DSS) adoption is limited by poor engagement with end users, and despite good practices in systems development and accessibility via farm based computers, uptake remains low, at under 25%. Kuhlman and Broderson (2001) report similar results in Germany. It is not news to state that projects that involve end users closely in development activity have greater uptake, but continuing low uptake, particularly in this agricultural context, suggests a need to go further, towards greater end user development and ownership. Kahai *et al.* (1998) found that increased involvement in DSS model building led to higher expectations of success, which was associated with higher satisfaction, and increased likelihood to refine a model prior to making a decision. An approach that actively involves the end users' contexts in ongoing system design and usage is further motivated because the obsolescence of description in time as environmental changes occur implies increasing irrelevance, recurring re-engineering and mistrust in a system's applicability.

The lack of adoption of developed systems have lead the DSS and modelling development group, Agricultural Production Systems Research Unit (ASPRU) to develop an alternative view on the use of decision support in rural industries in Australia. ASPRU has put more of an emphasis on training end users in the use of developed research tools such as simulation models rather than developing intelligent systems for end user use. Proponents of the ASPRU approach called FARMSCAPE (Farmers', Advisers', Researchers', Monitoring, Simulation, Communication And Performance Evaluation) (Carberry *et al.*, 2002) suggests that farmers and officers directly advising farmers should under go training programs in the use of research tools that will help them make decisions on their own farm. Other members of ASPRU express concerns over the past development of expert systems for farmers and consider that decision support development holds promise in only four directions, namely a tool for aiding tactical decisions, a simulator for a consultant, a simulator in a learning laboratory and a formal framework supporting regulatory objectives and documenting farm practices (McGown, 2002). We consider this to be a very pessimistic view of the technology, given its usage in other industries and consider that intelligent systems could at the very least be a useful tool for aiding farmers in making tactical or even strategic decisions such as the resource allocation problem discussed in this paper.

One major motivation for shifting the burden of design onto end users in their working contexts is exactly the issue of their superior and current knowledge of the business domain.

Users' sensitivity to interpretive nuances, to contextual factors, to tacit components and relevancy issues are all areas unamenable to simple description, quantification and communication using symbols, certainty factors and other instruments predicated on the assumption that such expertise can be relayed reliably to external systems developers. Also, for many domains, users' knowledge has a vague or probabilistic quality, rather than the precise quantifications that classical systems require to be explicated. Allowing for these within the design is critical, since systems that neglect an experienced decision maker's own intuitions will lose acceptability and credibility, affecting uptake.

McGill (2001) in her review of the literature concerning user developed applications (UDA) outlined several risks associated with user developed spreadsheets including ineffective use of resources, threats to data security and integrity, solving the wrong problem and unreliable or incompatible systems. However, McGill concluded that the potential risk of inefficient use of personnel time is outweighed by superior decision making by end users and this was related to insights gained from system development. This research indicates that for end user developers, the benefits are substantial in that they can gain insights to the problems being addressed but it doesn't alter the fact that the ultimate end user is still faced with the problems of poor design and unreliable or incompatible systems.

Numerous other limitations of using end users as traditional developers have been identified. For example, Wagner (2000) describes the particular difficulties associated with end user development of knowledge-based systems, both in terms of design quality and knowledge content. His analysis of 25 expert systems written by non-professional developers revealed that there were "significant quality and size limitations that indicate limited feasibility of end user expert system development" and further contends that "the lack of design quality may not be easily compensated for by a "knowledge advantage" of the end users. His pessimistic conclusion, together with the other literature cited above suggests that end user knowledge based system development is unlikely to progress within a model of traditional development, using training strategies, or traditional shells as the vehicle. Instead, there is a need for a different type of technology, one that allows knowledge to be explicated from a user's concepts to allow useful formalizations of a knowledge base, but retaining quality processes of system development. We now go on to consider a general solution strategy that is not bound by the above limitations.

CHARACTERISTICS OF A KNOWLEDGE SYSTEM DEVELOPMENT TECHNOLOGY

We consider several aspects are important in ensuring the success of end user developed knowledge systems. These incorporate the measures of KBS success identified by Yoon *et al.* (1995) where increased levels of user involvement led to greater satisfaction and usage. But beyond these are factors that engage with the psychology of the decision maker, and particularly in domains where holistic, intuitive and heuristic judgements are required the need for plausible explanation and decision rationale is paramount.

A strength of the symbolic approach to KBS development is that an explanation facility can be provided, minimally a trace of a logic path, but also more sophisticated model based argumentation. Moulin *et al.* (2002) notes a lack of confidence in decisions by DSS users, and cite evidence for this. For example decisions a user does not fully understand will be discarded (Hollnagel, 1987 (in Moulin *et al.* 2002); Cox 1996), and justifiable recommendations that are not among the alternatives a user foresaw (Guida *et al.* 1997 (in Moulin *et al.* 2002)) may meet a similar fate. Importantly, if a system's recommendations are based on a decision making process different to that of human decision makers, reluctance to accept them may ensue (Jiang *et al.*, 2000).

The traditional approach of intelligent systems development has been adapted and hybridised over the years and there are now a variety of intelligent systems approaches possible for decision support. It is, however, important to provide the ability for industry end users to determine the relevant variables and provide explanations in the language understood within the industry. It is thus appropriate to use a symbolic approach in developing an intelligent system for end users. Systems that process content and provide explanation in the language of their users' domain "not only builds belief in the results, but also increases the reliability of the system" (Wooley, 1998). Other approaches such as the development of flexible expert systems (Chan and Lau, 1997) and the use of soft computing to build hybrid intelligent decision support systems (Zeleznikow and Nolan, 2001) still require specific expertise in areas such as multiple-criteria decision models, fuzzy logic, neural network theory and Bayesian inference. Many of these areas of knowledge are not well known by domain experts, and require computational expertise in systems development beyond what can be expected of industry end users. They often suffer from a lack of transparency in reasoning, and associated trust in their argument. It is also arguable that systems that require the assignment by end users of numerical co-efficients as a surrogate for certainty are context bound, subject to variability and numerous cognitive biases (Kahneman, Slovic and Tversky,1982; Kahneman and Tversky, 1996). Numbers generated from sample sets are likewise sample bound, subject to the range of variables selected, and liable to be mistrusted if their recommendations do not agree with domain experts experience or trust in the sample quality. In specialised fields, a system developed in one context may have no relevance to an apparently similar one. For example, in the domain of dairy farming, the types of inputs used to produce milk in Queensland differ significantly from those in Victoria. Measures based on the use of technologies such as application of nitrogen fertiliser or water use efficiency when using irrigation are less relevant in one state than another, even within the same nation.

Knowledgeable specialists in their own industry *are*, however, best placed to identify and assess the factors relevant to their own specific decisions. Contextual variables, which cannot be predicted in advance, affect any particular decision, and this precludes any approach which prescribes how those factors apply in a statically modelled situation. Traditional expert systems, whilst providing logical explanation and decision rationale do not handle changing situations where other factors and information come into play (Crowe *et al.*, 1996). In addition, providing an expert system shell, that in principle allows end users to develop systems by providing their knowledge base, not only fails to overcome this limitation, but also still requires some knowledge engineering expertise. Goodell and Traynor (1997) suggest that end users program as a means to an end, and rarely have time or inclination to learn the tools and skills of a professional programmer. Some balance between the rigorous development of systems, and the typically pragmatic, contextualised and real time decisions made daily in industry is required.

The concept developed in this paper aims to take the strengths of the expert system paradigm, but to go beyond it to allow industry end users to shape their own decision support systems in actual decision making contexts. The explanation facility within our approach also allows description in industry-specific terms, which is not possible with statistically oriented "black box" techniques. For end user acceptance and trust in the system's outputs this is considered important (Wooley, 1998). In addition, Ye and Johnson (1995) have shown empirically that explanation facilities can make expert system generated advice more acceptable to users.

AN ALTERNATIVE APPROACH

Although spreadsheets enabled end users to build business related applications, there is still no equivalent "killer application" for knowledge-based systems. In his conclusion, Wagner (2000:12) asks, "Which new technology will enable end users to become knowledge system developers?" In this section we describe some general characteristics of such a technology.

Traditional systems development approaches identify and freeze requirements in conceptual models, which often fail to adapt to changing circumstances. Much of the literature on end user development has been framed within traditional systems development paradigms, and entailed initiatives such as training users in systems development techniques, or using professional designers as intermediaries. There are however, other approaches which show promise for the DSS application area. Winograd (1995) detailed the case for a philosophy of design environments, which support the use of conceptual models of a user domain in forms meaningful to the user, providing a "responsive prototyping media". Such design environments do not simply model a snapshot of a business, but provide for evolutionary development by domain expert end users to permit new practices and innovations to occur.

Gammack *et al.* (1992) describe a prototype design environment for intelligent decision support called IDIOMS. This research project was able to overcome several of the limitations of engineered expert systems. These limitations included their lack of context sensitivity, the requirement for intermediary engineering, built-in obsolescence, static form, and user passivity. The prototype developed through IDIOMS included utilities for symbolic specification in a business user's own terminology, avoiding a representational pre-commitment to terms originating with, and understood by, a knowledge engineer. A constraint based (relational) knowledge representation was used from which decision rules could be derived from databases, without requiring humans to express their (perhaps tacit) knowledge in an unfamiliar, and possibly inappropriate, target language. Such rules then formed the basis of an intelligent system for a specifically developed business application.

The IDIOMS research demonstrated how end user developed applications could be dynamically developed from an historical and distributed data resources, while enhancing this with the contextual information embodied in the human knowledge resource. This form of systems development is able to recognize the nature of contemporary information and its usage and instead of embodying "pure intelligence" in a system, as the original expert system conception held, the design recognises that decision support needs to integrate available information within its context of use, which is essentially located in the social world.

Although the architecture of IDIOMS was conceptually and physically distributed (Kerridge, 1994) allowing scalable implementation with massive data sets, and any number of information agents, the prototype management decision support environment did not require this, and was effectively developed for a single virtual machine. The decision models built within the IDIOMS research project required an individual model builder and although intelligent systems were developed within the IDIOMS research for classifying benchmark datasets on mushrooms and irises (Fisher, 1936), the only business applications developed were for the banking industry, e.g. home loan decisions (Gammack *et al*., 1992). The research however, does indicate that a design environment can be tailored to specific industry requirements.

Going beyond the ideas suggested by the IDIOMS research project, integration of knowledge from multiple sources remains a challenging general problem. This paper proposes that using the developed design environment, a number of particular knowledge based applications will be readily developed by reference to databases and statistics available to any industry and qualified by industry experts' knowledge of their appropriate usage in decision making.

This approach has the potential to facilitate the development of an innovative environment within which specific systems relevant to any industry can be developed. This development can be done by knowledgeable, domain-specific industry experts, and made available for operational end-users in making decisions. One assumption in this approach is that much of the knowledge needed to make industry specific decisions is available as human expertise within specific industries. It is further assumed that this knowledge may not be readily available to operational end users. These assumptions can easily be verified through studies of the dissemination of knowledge in small rural businesses. For example, in the dairy industry there have been many attempts to capture human expertise for the development of expert systems, both in Australia (Kerr *et al*., 1992; Kerr *et al*., 1999a; 1999b) and overseas (Schmisseur and Gamroth, 1993; Esslemont and Kossaibati, 2000). Other significant aspects of this approach is in being able to extend the dynamic capabilities of intelligent rule based systems, whilst enabling rapid systems development within industry contexts by domain experts. In the next section we propose a specific development approach suited to these identified characteristics.

In knowledge based systems research it has often been observed that users do not express themselves naturally in the target language, usually logical rules (Barr, 1999). However, rules remain a powerful representation with natural logic of computation, and which allow validation and explanation by reference to humanly understood concepts. It is also recognised that people know more than they can tell, that recognition is easier than explication, and that rules embody conceptual associations that domain experts make and can justify. One approach to overcome the requirement to express knowledge in rules is to ask for simple relationships to be described, and to generate rules from those. There are several mechanisms that allow such knowledge to be elicited in common use already, such as repertory grids (Gammack, 1987; Bradshaw, Ford, Adams-Webber and Boose, 1993), and constraint representations (Bowen and Bahler, 1993). By representing knowledge in a simpler target language than rules, more flexibility, both computationally and in explanation capabilities is possible. This is because relations between domain variables, numeric or semantic, are simply expressed. This can form the basis of a "semantic spreadsheet".

For example, for the conversion between Fahrenheit and centigrade scales, a constraint is formally expressed as the relation F=9/5C+32. This single expression is sufficient to compute values in either direction, so that knowing a value for C will allow computation of a value of F, without a requirement to provide a procedural description of the inverse relationship that other representations would require. For more complex systems of variables, expressing the knowledge of their interrelationship as constraints is simpler and more natural and justifiable, and complex, multivariate rules can be expressed by generating them from systems of simpler constraints. Such computational mechanisms are inherent in spreadsheets where variables are linked, and values update by reference to known information. Building semantic models of the domain around non-numerical (knowledge) concepts allows similar types of computation to occur in specialised expert domains. For example, if we know that (Aaron is a boys name) (boys are male) (males wear ties (low probability)) (tie wearing for interview is higher) and that (someone called Aaron is at the reception desk for interview), we can infer that the person with the tie at reception is the one expected to be Aaron. This is an everyday type of human computation, but the computational semantics is simple, requiring only a representation of the relevant knowledge, propagated by simple bindings and probabilities. But because such type categories are not easily available in spreadsheets, a different technology is required, which is our aim in this research. Providing an environment in which domain knowledge is represented and linked in such expert user developed models is the first stage.

A specific intelligent system corresponds to a semantic spreadsheet, formed by selecting functions and importing informative materials from other sources as a domain expert user chooses. The produced spreadsheet template can be made available to ultimate end users, who can use it to provide information for supporting decisions in their own context, and for sophisticated users, to amend the formulae quite specifically.

METHODOLOGY FOR DEVELOPMENT

Because expert systems developed by non-professional developers have significant design quality problems, other ways of assisting non-professional developers need to be derived. Our general solution to the problems is proposed with the following characteristics. It is proposed that the design environment can be developed in two phases, the first being based on the development of several traditional expert systems. These initial intelligent systems can be developed using traditional evolutionary prototyping methods (Berry 1994; Jojo and O'Keefe 1994; Luqi *et al.* 1998; Kerr *et al.* 1999), as this will involve the proposed end users in the development process. The method is also analogous to iterative prototyping as recently described by Alter (2001). This method has the advantage of reducing risk by basing each successive layer of development on the success of the previous layer, and builds in user validation at each stage. It is acknowledged that this initial development will involve third party intermediaries and it is expected that these people will play a significant role in the development of the design environment.

Knowledge that will facilitate the development of the knowledge base necessary for the initial intelligent system development can be acquired by a series of unstructured interviews (McGraw and Westphal, 1990; Blignaut and Venter, 1998) and focus groups (Gibbs, 1997) of relevant experts in the fields of expertise sought, as well as by various cognitive techniques specific to identifying the categories used in making decisions. This will identify concepts, relations, constraints, and other parameters relevant to decision making. These initial systems can be developed using an expert system shell as they provide a convenient method of separating the knowledge base from the program control mechanisms.

These will then assist in the second phase, as they will form the basis of a design environment in which further specific and evolvable intelligent systems can be developed by industry end users. The knowledge elicited in developing the initial systems can then be abstracted to identify the parameters involved in decision-making, and the relations among decision categories. These categories identified form the basis of a design environment in which the parameterised sources of information and knowledge can be brought together by industry experts to produce a system specific to making decisions in a field of current relevance. The extension of this will be an industry specific template developed using the constraint-based representation described by Gammack *et al.* (1992). This will enable the domain expert to concentrate on the knowledge base while program control and the template will facilitate organisation of the knowledge.

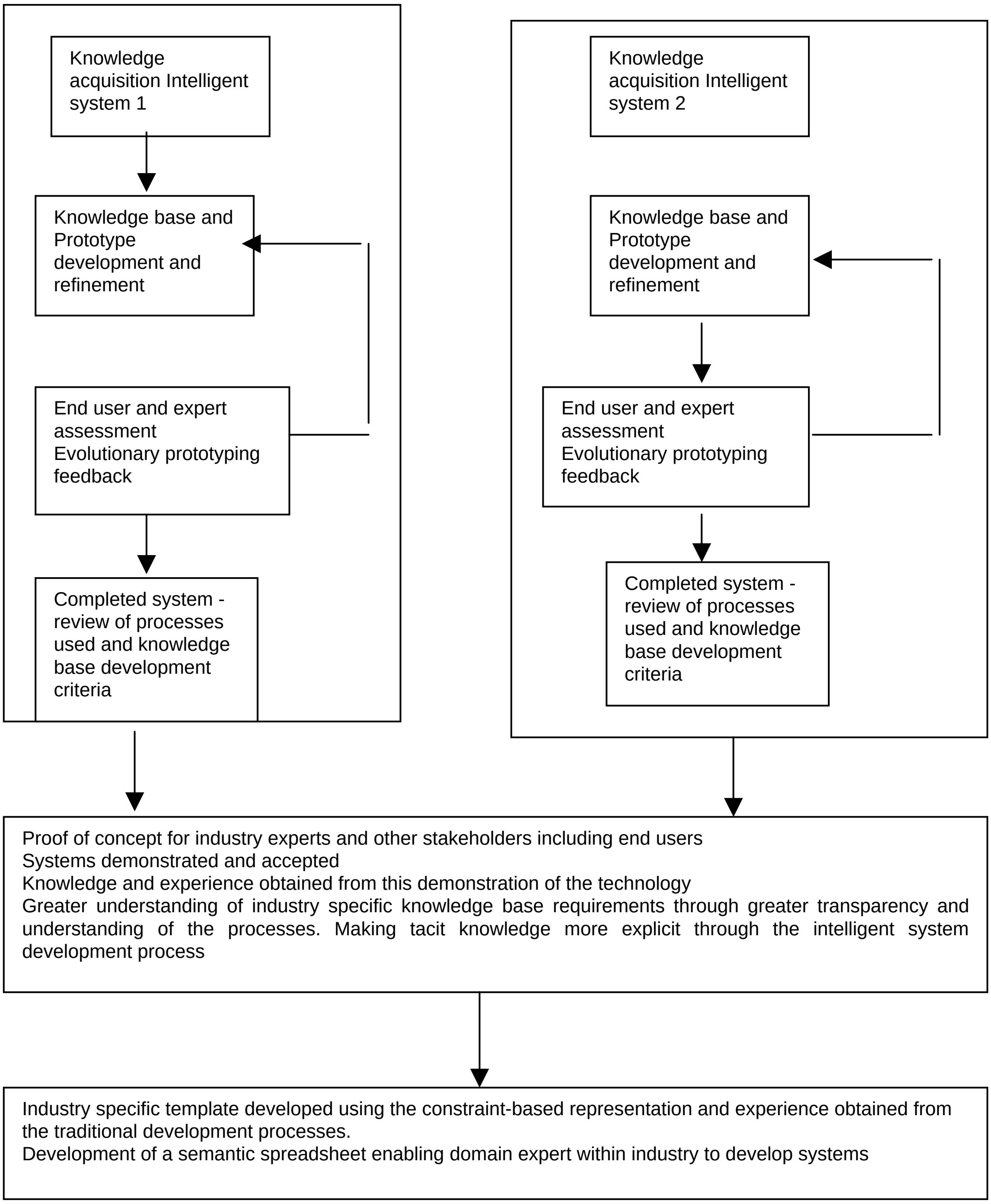
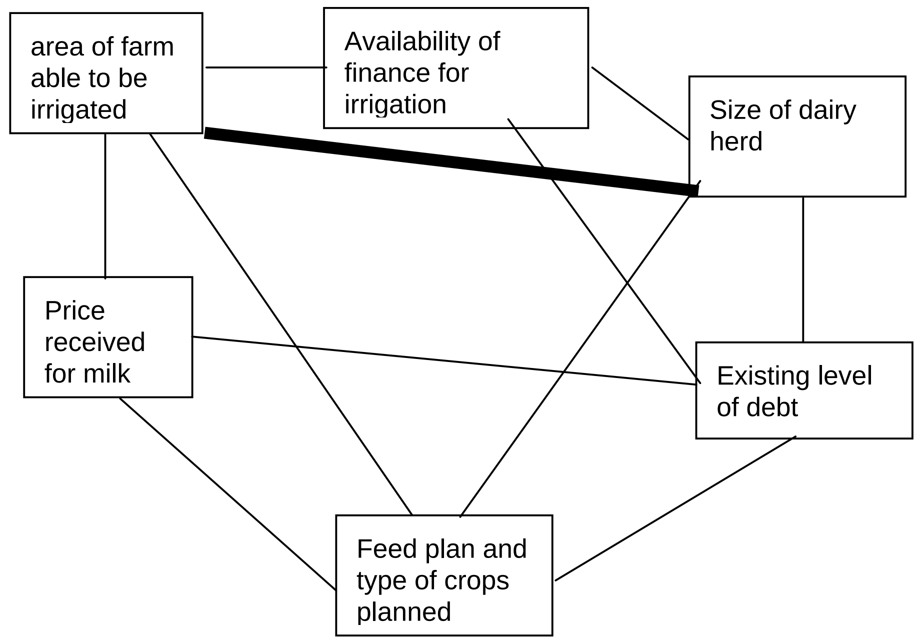


Figure 1: Diagrammatic representation of the proposed methodology for development

An example of how the semantic spreadsheet would be developed is adapted from Gammack *et al.* (1992) and shown as a constraint graph in Figure 2 below. In this case certain values in categories shown below can constrain the feasibility of implementing an irrigation system on a farm. A line between two variable represents a constraint, where the value of one (e.g. acreage of farm able to be irrigated) constrains the possible values for the other (e.g. size of dairy herd) as one value becomes fixed, it impacts on others linked in the graph. The value of one variable may be of particular interest at a given time, e.g. price received for milk. The size of dairy herd does not directly affect the price received for milk, but a value for this target variable can be predicted by propagating known values through the constraint graph. In the example shown in Figure 2, the percentage of the farm that can be irrigated would be a major constraint to implementation.



Size of dairy herd (number of cows)	Area of farm able to be irrigated (hectares)
100	40
80	32
60	24
40	16
20	8
0	0

Figure 2: Constraint graph showing the relations between irrigation potential of a dairy farm and milk production, with details of the constraint shown as a matrix

Rather than develop application-specific intelligent systems for particular ends, each developed intelligent system can provide greater transparency to the processes involved in decision-making, as explanation facilities describing the processes and assumptions used to derive an answer can be incorporated in the final product. This can then be abstracted into generic templates in a design environment in which industry end users can supply current information of relevant sorts for decision support. The environment will allow incorporation of various knowledge sources to build models, which can be used directly for specific problems. Due to this generality, the design environment can be validated through extension to other domains, and where applicable, other industries with analogous problems.

The methodology for both initial expert systems and design environment development will need to include an initial demonstration of the capabilities of the technology. The participants of the focus groups and domain experts will then determine the key components needed for the development of the initial expert system. The design environment developer will liaise with participants of the focus group meetings and domain experts. The aim is to develop intelligent systems that are simple to operate and yet capture the expertise of a range of domain experts. It is important that there is a strong focus on operational end users and other key stakeholders having ownership in the development of each intelligent system. These key stakeholders will be able to provide feedback on each prototype and this will ensure that future system developers are aware of the development process. This phase of the development process will be an on-going iterative one, facilitated by the existence of flexible development tools such as expert system shells and the design environment developer's own work on the development of the template.

This process will allow expert industry end users to select, source, query and combine information relevant to specific decisions facing them in the critical areas, and to set in motion processes to generate an intelligent system that can provide answers based on this information. Such systems may then be made available to operational end users for use in their own settings.

The semantic spreadsheet metaphor introduced earlier provides a useful analogy with the design environment. A spreadsheet enables decisions to be made by end users who can supply values to identified formulae. Sophisticated users can add formulae, and express other relations among known variables, and protect or enable more sophisticated operations. Clearly the knowledge of the spreadsheet developer can vary on a continuum from IS professionals through to end user data entry clerks. Spreadsheets tend to use numerical formulae, and do not accommodate semantic relationships. Our proposed technology encourages domain knowledge embodied in relational expressions to be described in the form of constraints. Such constraints, which are simple relationships between variables, can then be used as a basis for deriving sophisticated rules, and can make use of information from a variety of sources. End users can then, depending on their level of sophistication, provide values or add qualifying relationships to the representation in the decision model.

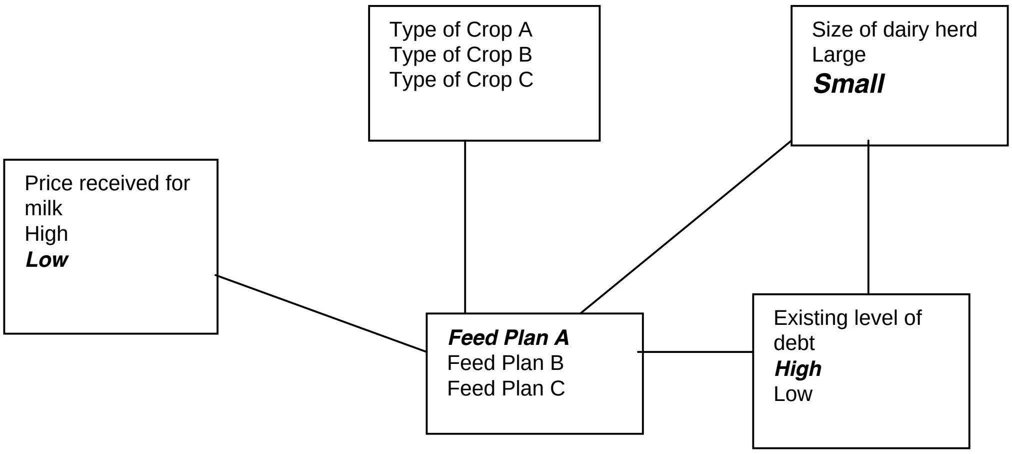


Figure 3: The sort of input an end user might supply to a rule-based model derived from the constraint graph in Figure 2. Heuristic values are indicated, but numerical quantifications are also possible.

The example shown in Figure 3 demonstrates how the constraints (shown in bold and italicised) of a small herd size, high levels of debt and using feed plan A may result in a low price received for milk. This is feasible in the present deregulated dairy market where there is a price differential for payments based on the quality of the milk, (a reflection on the type and amount of food available to the milking cows) and quantity of milk (a reflection on the small herd size). A high level of debt would also be a major constraint to increasing herd size. As mentioned above an end user could change values or qualifying relationships to conduct “what-if” scenarios of change. For example in the situation shown in Figure 3, a change in feed type enabling the farmer to implement feed plan B could result in an increase in the price received for milk through improved milk quality. Alternatively, a restructuring of debt might allow the farmer to increase herd size resulting in an increase in milk volume and subsequent increases in total receipts and the price received for the milk.

CONCLUSIONS

Klein and Methlie (1990) state that in an uncertain world it is important to distinguish between decisions and outcomes. A good decision does not necessarily lead to a good outcome and the quality of a decision depends on the correctness of the actions taken by the decision maker. As Damon Runyon put it, “The race is not always to the swift, nor the battle to the strong, but that’s the way to bet.” Decision-making quality can be measured in terms of time spent to make a decision, alternatives evaluated, and information searched.

The use of intelligent decision tools developed within design environments will allow operational end users to explore more options, search through more information and consider many more variables associated with changes in their enterprise than they would have under normal circumstances. This has implications both for the quality of the final decision (since more information and variables can be considered) and for developing the ultimate end user's (e.g. a farmer's) long-term ability to make decisions. The development of a design environment that allows industry experts to develop intelligent systems that will assist operational end users explore more options will help in the implementation of more informed decisions and ultimately improve the competitiveness of businesses within a given industry.

We believe that there is scope for the development of intelligent systems for farmers and farm advisors and that the relevant domain expert can undertake this development. We also consider that the ASPRU approach will end up being too resource intensive with workshops having to be directed to each farmer at the individual farm level and that a more general approach should provide both farmers and their advisors with tools that will assist them with resource allocation problems specific to their farm. By having a professionally developed design environment, with parameters informed by industry professionals, specific intelligent systems sensitive to end user knowledge can be built, and Figure 3 shows an example of this in action.

This approach will provide domain experts in any industry with a product analogous to a semantic spreadsheet and should provide significant advantages to traditional expert system shells. It is hoped that this approach will overcome many of the limitations of expert system shells such as content, size, quality and scalability as described by Wagner (2000) and ineffective use of resources, threats to data security and integrity, solving the wrong problem and unreliable systems as described by McGill (2001). It is anticipated that proof of concept testing can be done on resource allocation problems specific to the Australian dairy industry. If the concept is proven, the approach should be applicable to any industry. This is the subject of our continuing research.

REFERENCES

Alter S. (2001) Which Life Cycle - Work System, Information System or Software? Communications of the Association for Information Systems 7(17), October 2001.

Barr V. (1999) Applications of rule-based coverage measures to expert system evaluation: Knowledge Based Systems 12: 27-35

Berry, D.C. (1994). Involving users in expert system development. Expert Systems, 11:23-28.

Blignaut R.J and Venter I. M. (1998) Teamwork: can it equip university science students with more than rigid subject knowledge, Computers and Education 31(3):265-279.

Bowen J. and Bahler D. (1993) Constraint-based software for concurrent engineering. Computer, 26(1):66-68, January.

Bradshaw, J.M., Ford, K.M., Adams-Webber, J.R. and Boose, J.H. (1993). Beyond the repertory grid: new approaches to constructivist knowledge acquisition tool development. International Journal of Intelligent Systems 8(2) 287-33.

Carberry P.S., Hochman Z, McCowan, R.L., Dalgliesh N.P., Foale M.A.., Poulton P.L., Hargreaves J.N.G., Hargreaves D.M.G, Cawthray S., Hillcoat N., Robertson M.J., (2002) The FARMSCAPE approach to decision support: farmers', advisors', researchers' monitoring, simulation, communication and performance evaluation, Agricultural Systems, 74 141-177.

Carlsson C and Turban E (2002) DSS: directions for the next decade, Decision Support Systems 33 (2) 105-110

Chan C. and Lau P. (1997) An expert system architectural framework for engineering selection, Engineering Applications in Artificial Intelligence 10:4 357-367.

Cox, P.G. (1996). Some Issues in the Design of Agricultural Decision Support Systems. Agricultural Systems, 52:355-381.

Crowe, M., Beeby, R., and Gammack, J. (1996) Constructing systems and information: a process view. McGraw Hill, Maidenhead.

Davis R and Lenat, D.B. Knowledge-based Systems in Artificial Intelligence (McGrawHill, New York, 1982).

Esslemont and Kossaibati (2000), The use of databases to manage fertility, Animal Reproductive Science 60-61 pp 725-741

Fisher, R. A. (1936). The Use of Multiple Measurements in Axonomic Problems. Annals of Eugenics 7, 179-188.

Gammack JG (1987) Different Techniques and Different Aspects on declarative knowledge In AL Kidd (Ed) Knowledge Acquisition for Expert systems: a practical handbook, Plenum New York.

Gammack, J.G., Fogarty, T.C., Battle, S.A., Ireson, N.S., and Cui, J. Human-Centred Decision Support: The IDIOMS System (1992) Artificial Intelligence and Society, 6 345-366.

Gibbs A (1997) Focus groups Social Research Update 19, University of Surrey

Goodell H. and Traynor C. (1997) End-User Computing SIGCHI Bulletin, 29 4, October 1997

Hayman P. T. and Easdown W. J. (2002) An ecology of a DSS: reflections on managing wheat crops in the northeastern Australian grains region with WHEATMAN, Agricultural Systems, 74: 57-77

Janvrin D. and Morrison J. (2000) Using a structured design approach to reduce risks in end user spreadsheet development Information & Management 37 1-12

Jiang J.J, Muhanna, W.A. and Klien G. (2000) User resistance and strategies for promoting acceptance across system types, Information & Management, 37 (1) 25-36

Jojo, L.P. and O'Keefe, R.M. (1994). Experiences with an expert systems prototyping methodology. Expert Systems, 11:13-21.

Kahai, S. S. Solieri, S. A. and Felo, A. J. (1998) Active involvement, familiarity, framing, and the illusion of control during decision support system use, Decision Support Systems, 23, (2) 133-148.

Kahneman, D., and A. Tversky. (1996). On the reality of cognitive illusions. Psychological Review 103:582-591.

Kahneman, D., P. Slovic, and A. Tversky, Eds. (1982). Judgment under Uncertainty: Heuristics and Biases. New York: Cambridge University Press.

Kerr D.V., (1996) DAIRYPRO - A Knowledge Based Decision Support System for Strategic Planning on Dairy Farms in Northern Australia, Unpublished PhD thesis, Griffith University, Australia.

Kerr D.V., Chaseling J., Chopping G.D. and Cowan R.T., (1999b) DAIRYPRO - A Knowledge Based Decision Support System for Strategic Planning on Sub-Tropical Dairy Farms - II. Validation Agricultural Systems 59 (3): 257-66.

Kerr D.V., Cowan R.T. and Chaseling J., (1999a) DAIRYPRO - A Knowledge Based Decision Support System for Strategic Planning on Sub-Tropical Dairy Farms - I. System Description, Agricultural Systems 59 (3): 245-55.

Kerridge, J (1994) The IDIOMS Parallel Database Machine: Design, Performance and Future Directions, In: S Noguchi et al (eds), pp33-52. 6th International Conference Transputer and Occam, 1993, Tokyo,

Klein, M. and Methlie, L.B. (1990). Expert Systems: A Decision Support Approach. Addison-Wesley Publishing Company, Workingham, England.

Kreie J, Cronan T.P., Pendley J and Renwick J.S. (2000) Applications development by end-users: can quality be improved? Decision Support Systems 29 143–152

Kuhlmann, F and Brodersen C. (2001) Technology and farm management: Developments and perspectives Computers and Electronics in Agriculture 30 71–83.

Luqi, Chang C.K and Hong Zhu (1998) Specifications in software prototyping. The Journal of systems and software 42:125-140.

McCown R.L. (2002) Changing systems for supporting farmers' decisions: problems, paradigms, and prospects, Agricultural Systems 74 179-220.

McGill T., (2001) The application development process: What role does it play in the success of an application for the user developer? Proceedings of the twelfth Australasian conference on information systems 4-7 December 2001 Coffs Harbour NSW, Australia

McGraw, K.L. and Westphal, C.R. (1990). Readings in knowledge acquisition: Current practice and trends. Ellis Horwood Series in Artificial Intelligence, New York, United States.

Moulin, B. Irandoust, H, B langer M and Desbordes G (2002) Explanation and Argumentation capabilities: towards the creatyion of more persuasive agents AI review 17 (3) 169-222

Nardi, Bonnie A., A Small Matter Of Programming, Perspectives on End User Computing. MIT Press, Cambridge MA, 1993.

Owei V. and Navathe S. (2001) A formal basis for an abbreviated concept-based query language Data & Knowledge Engineering 36 109-151

Schmisseur, E. and Gamroth, M.J. (1993). DXMAS: An expert system program providing management advice to dairy operators. Journal of Dairy Science. 76:2039-2049.

Surinder S., K, Solieri S. A.,and Felo A. J. (1998) Active involvement, familiarity, framing, and the illusion of control during decision support system use, Decision Support Systems 23: 133–148.

Wagner (2000) End Users as Expert System Developers?: Journal of End User Computing, Vol. 12, No. 3

Winograd, T. (1995) From Programming Environments to Environments for Design. CACM 38(6): 65-74

Wooley B.A. (1998) Explanation Component of Software Systems, ACM Crossroads 5 (1).

Ye, R.L. and Johnson, P. E. (1995) The impact of explanation facilities on user acceptance of expert systems advice; MIS Quarterly, Minneapolis; 19(2)157-171.

Yoon, Y.; Guimaraes, T.; and O'Neil, Q. (1995) Exploring the factors associated with ES system. MIS Quarterly, 19 (1) 83-106.

Zeleznikow J. and Nolan J.R. (2001) Using soft computing to build real world intelligent decision support systems in uncertain domains, Decision Support Systems 31 263-285.

COPYRIGHT