TOWARDS SPECIALIZED EXPERT SYSTEM BUILDING TOOLS: A TOOL FOR BUILDING IRRIGATION EXPERT SYSTEMS

Samhaa R. El-Beltagy, Gamal Al-Shorbagi, Hesham Hassan and Ahmed Rafea

Central Lab for Agricultural Expert Systems
Agricultural Research Center
Ministry of Agriculture and Land Reclamation.
Cairo, Egypt
El-Nour St. P.O Box 438 Dokki,Giza, Egypt
E-mail: {samhaa, gamal_sh, hesham, rafea}@.claes.sci.eg)

Abstract: Expert system development is very often an expensive process which requires significant time and effort. This paper investigates the idea of building a specialized tool for a specific task in order to realize very rapid development times with very little effort on the part of the developer thus considerably reducing the overall development cost. The particular task that this paper addresses is that of irrigation scheduling. The work presented describes various aspects of a tool that was built for rapidly developing irrigation expert systems for vegetable crops in Egypt, and shows how greatly this tool simplifies the process of building such systems.

Keywords: Agriculture, Expert systems, Knowledge tools, Knowledge engineering, Modelling.

1. INTRODUCTION

Despite the availability of a wide spectrum of expert system tools and shells, developing an expert system is still often a time consuming and expensive process. Further more, the development of a 'good' expert system is still also highly dependant on the skills of its knowledge engineers. Reducing expert system development costs in terms of development time and required skills continues to be the goal of many research bodies. When addressing this issue for expert system development in general, improvements in cost and effort are likely to come solely from enhancements in development shells. However, much more significant improvements can be achieved in domains where expert system development is a recurring activity for a set of recurring tasks. Building agricultural experts systems for a wide variety of crops and a number of tasks (diagnosis, treatment, irrigation, fertilization, etc) is an example of such domains. Instead of replicating the same effort every time an expert system is being developed, commonalities can be captured and re-used. Such re-use can be achieved by standardizing ontologies and by building specialized tools for specific tasks within a specific domain. Such tools would not only contain a problem solving model for some given task, but would also include all knowledge that can be re-used for this particular task. The goal of this paper is to present an example of such a tool built specifically for irrigation expert systems. Section 2 of this paper briefly reviews some related work, section 3 explains the goals of irrigation expert systems as well as the goal of the tool to be presented, section 4 describes the tool that has been built, and section 5 concludes this paper and presents some ideas for feature work.

2. RELATED WORK

Knowledge system technology has been applied to a variety of agricultural problems since the early 1980s. Generally we can classify agricultural activities into: activities that are done on the farm prior to cultivation and activities, which are done during cultivation operations. The scope of this research concentrates on an activity type that is done during cultivation. Specifically, this work focuses on the irrigation scheduling activity. Many systems have been built to address the irrigation scheduling problem. These include for example the NEPER Wheat expert system (Kamel, *et al.,* 1995). The NEPER Wheat expert was developed for handling most of the production management activities including irrigation for the wheat crop. The generic task knowledge system development methodology (Chandrasekaran, 1988) has been utilized to develop this system. The generic task methodology is one that has recognized the importance of task re-use for speeding up the process of building knowledge based system. The methodology has identified a number of problem solving building blocks that target specific problems. For example, the hierarchical classification problem solver can be used to build a diagnosis system while a routine design problem solver can be used to build a scheduling system.

Other examples of irrigation expert systems include the barley crop management expert system (Boner I., 1992) which was designed to produce water and fertilizer recommendations to maximize yield., and the CALEX/Cotton expert system (Ostergard M.and Goodell P. 1992; Plant R. 1989) which was designed to link diverse production components including irrigation, pest management and agronomy. The CUPTEX (El-Dessouki, *et al.,* 1993) and CITEX (Edrees S., 1997) systems for cucumber and citrus production management respectively cover most of the agricultural practices for cucumber crop under tunnel and citrus cultivation in open fields including irrigation and fertilization. References for other expert system in this domain may be found in (Rafea, 1998). However each of these systems was built from scratch to target a specific crop. The extent to which these system could make use of previously built ones was limited on identifying a suitable knowledge representation scheme, and/or a suitable problem solving model. While these do in fact reduce the effort involved in building an irrigation expert system, they do not capitalize on the fact that one generic model containing common knowledge, equations and a problem solving model, can be applied to all. Through this application the reduction in both time and effort can be much more significant.

3. IRRIGATION EXPERT SYSTEMS: A CLOSER LOOK

The main goal of most irrigation expert systems is to produce an irrigation schedule for a particular crop in a particular farm. The output schedule is a plan of water quantities to be applied and the time of application according to the requirements of the plant and the affecting factors like the soil type, climate, source of water, etc. Determining a crop's water requirements is no trivial task, but is one that involves many equations that in turn involve an extensive number of complex variables relative to the soil, water, climate, crop etc. This work capitalizes on the fact that even though the irrigation requirements for various crops may vary, a number of basic concepts and irrigation determination techniques are shared among all. By recognizing this fact, a tool was built to hide all the complexities of an irrigation's system equations and knowledge from the developer, while highlighting any missing knowledge that can vary from one crop to another thus guiding the developer as to 'what' knowledge needs to be acquired. In a sense, the developer is simply offered an empty template for specifying the inputs and rules for determining or calculating the value of a property of some pre-identified concept. This greatly simplifies the knowledge acquisition task, and makes it possible for any person with some computer knowledge to easily build an irrigation expert system. In the next section we present this tool.

4. THE DEVELOPED IRRIGATION TOOL

A typical irrigation expert system is made out of concepts, relations, and equations on top of which a task layer is built. The developed specialized irrigation tool was built on top of an expert system development tool called KSR. KSR was designed and implemented at the Central Laboratory for Agricultural expert systems in Egypt. In this work, we have aimed to identify and capture all knowledge that is related to the irrigation task regardless of the crop, as well as identify knowledge that may vary from one crop to another. The steps for developing the specialized irrigation building tool can be summarized as follows:

1. Identify the main tasks involved in the production of an irrigation schedule
2. Acquire and model knowledge related to each of these tasks (concepts, equations, relations, etc)
3. Identify concepts and relations that may vary from one crop to another
4. Develop a generic irrigation task layer
5. Build an interface that hides non-changing knowledge from the user as well as enables him/her from accessing modifiable knowledge and editing it.

The result of following these steps was a tool capable of assisting developers in rapidly building an irrigation scheduling system for any vegetable crop in Egypt.

4.1 Tool features and description

The tool enables the production of an irrigation schedule on a weekly basis provided that climate reference data is available. The irrigation model caters for flooding, or drip irrigation systems. The model also includes equations to cover the case of planting within an open field, high tunnels, or low tunnels. Since the method of determining the value of potential evotranspiration (et0), which is essential for determining a crop's water requirements, differs across planting environments, two well known methods for calculating this value known respectively as Hargreaves (Hargreaves and Samani, 1982) and Pennman (Doorenbos and Pruitt, 1984), have been adopted. The Hargreaves model applies only to high tunnels, while the Pennman model is applicable to the other two planting environments.

The tool currently has about ninety five built in concepts, thirty five relations, eleven tables, and fifty two equations as well as the irrigation scheduling problem solving model.

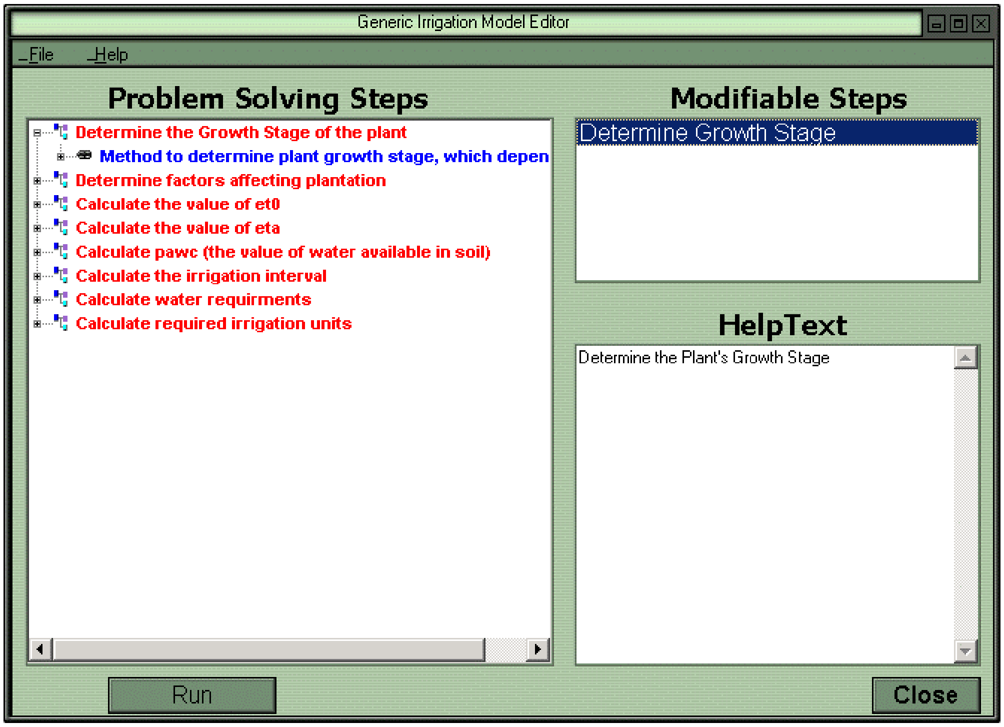
4.2 *Using the tool*

To develop an irrigation expert system, all the developer has to do is to examine the major tasks, identify any missing knowledge and enter that into the tool. Within the developed tool, eight major tasks for generating an irrigation schedule have been identified. These are as follows:

1. Determine the Growth Stage of the Plant
2. Determine factors affecting the Plantation
3. Calculate the value of et0 (potential evotranspiration)
4. Calculate the value of eta (exhaustion of actual water)
5. Calculate the value of pawc (water available in soil)
6. Calculate the irrigation interval
7. Calculate water requirements
8. Calculate required irrigation units

For each of these tasks, any relations or rules that may vary from crop to crop were identified. These have been termed 'modifiable steps' as they can be modified depending on the crop under consideration. For each modifiable step, the output is determined before hand, but the inputs that influence that output are left for the developer to specify. Each modifiable step is either represented by a set of rules constituting a relation, or a table. The expert system developer need only inspect the 'modifiable steps' associated with each of the eight major identified tasks and fill in the missing knowledge for each. These are clearly represented in the developed tool as shown in figure 1. After examining each step, the task of the developer is to acquire the knowledge which will accurately determine the required output of the step. So, the steps for developing an irrigation system can be summarized as follows:

- Determine the factors that influence any of the output(s) of a 'modifiable step'.
- Understand how the variation of these factors can influence the output.
- Through the developed tool, locate the concepts and properties representing the identified factors and use them to write the rules or to fill in the tables associated with each step, so as to reflect the acquired knowledge



Determine the Growth Stage of the plant
Method to determine plant growth stage, which depen
Determine factors affecting plantation
Calculate the value of et0
Calculate the value of eta
Calculate pawc (the value of water available in soil)
Calculate the irrigation interval
Calculate water requirments
Calculate required irrigation units

Determine Growth Stage

Determine the Plant's Growth Stage

Figure 1: Irrigation Expert system developer interface

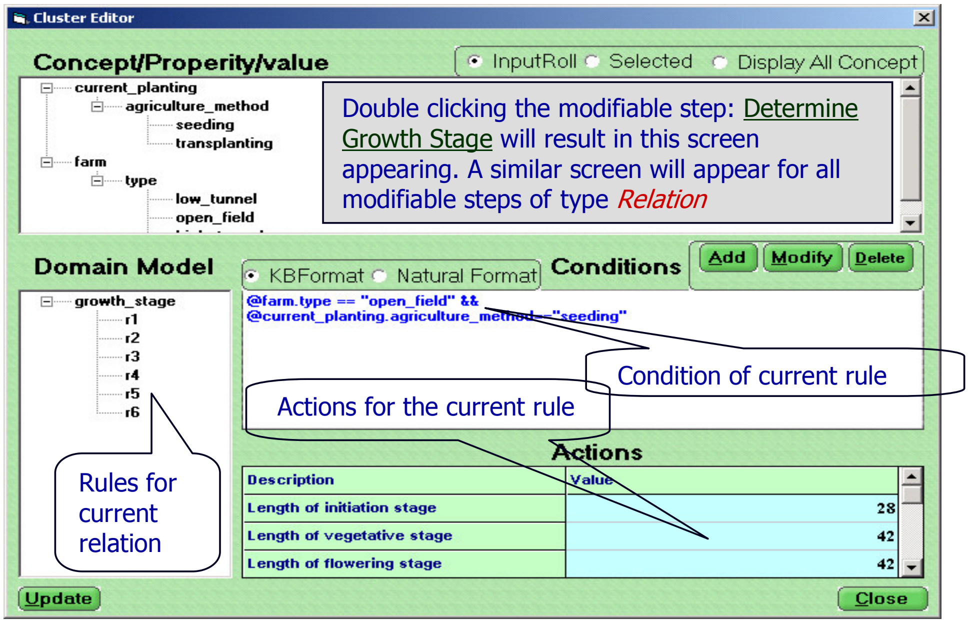
To further assist the developer in this task, each of the modifiable steps is described and the requirements from the developer are clearly laid out

Steps associated with Task 1: Determine the Growth Stage of the Plant
There is only one step associated with this task the name of which matches the name of the task itself "Determine Growth Stage". The step is of type 'Relation' meaning that it is composed of a set of rules. The purpose of this step is to be able to determine the length of each of a crop's growth stages in days. Example growth stages include: Initiation, Vegetative, Flowering, etc. Factors that can affect these, can include the crop's variety and/or other factors depending on the crop itself.

Steps associated with Task 2: Determine factors affecting the Plantation
Step1: Calculate Expected Fruit Yield (efy):This modifiable step is represented by a table. In this table you should place any factor that can affect expected fruit yield, as a column. There is only one possible output (efy) the value of which is specified based on the variation of the affecting factors.
Step2: Calculate Average Expected Fruit Yield:This step is also represented by a table, and steps similar to those described for efy should be followed to fill it. In this table the only possible output is that of the average expected fruit yield
Step3: Determine Variety Characteristic Bases: This step is of type 'Relation' and its output is the value of the 'variety factor'. The developer should thus aim to write the rules the determine the 'variety factor'.
Step4: Determine Optimal Number of Plants: This step is also of type 'Relation' and its output is the value of of the 'optimal number of plants factor'

Figure 2: A sample of help provided for filling in modifiable steps

for him/her. An example of the description of modifiable steps associated with two of the identified major tasks is shown in figure 2. In total the developer needs only modify five relations and four tables. While the developer can view any of the built in equations, he/she can not edit any of these. For modifying relations and tables, two specialized editors are provided. The relation editor is shown in figure 3. A separate interface is also provided for the entry of reference data (for the farm, soil, water, climate, etc.) by either the expert system developer or the end user.



Double clicking the modifiable step: Determine Growth Stage will result in this screen appearing. A similar screen will appear for all modifiable steps of type *Relation*

```
@farm.type == "open_field" &&
@current_planting.agriculture_method=="seeding"
```

Condition of current rule

Actions for the current rule

Rules for current relation

Description	Value
Length of initiation stage	28
Length of vegetative stage	42
Length of flowering stage	42

Figure 3: A snapshot of the relation editor

4.3 Knowledge Representation

For the irrigation task, mathematical dependency graphs were chosen to model the irrigation knowledge. The concept of dependency networks (DNs) was first introduced in the context of developing truth maintenance systems (Forbus and Kleer, 1993). A DNs is a directed graph consisting of nodes- representing variables or concepts- and links- representing the dependencies between these nodes. A mathematical dependency graph is a novel knowledge representation scheme suitable for representing mathematical knowledge in a declarative manner such that it can be traversed to calculate a quantity of a numeric object. A dependency graph consists of a set of nodes related to each other by directed arcs. An arc between a node A and a node B means that the value of node A depends on the value of node B or in other words, that there is a depend-on relation between these two nodes. The mathematical dependency between a node and any other nodes means that the value of this node is computed through a mathematical function, which includes the other nodes as parameters of this mathematical function. This form of knowledge representation was chosen for the following reasons:

1. To avoid recalculations.
2. To enhance readability and maintainability.
3. To promote reusability.
4. To enable explanation.

Figure 4 shows a simplified diagram for part of the dependency graph used in our tool. 𝔽 in the diagram denotes a function while ℝ denotes a relation.

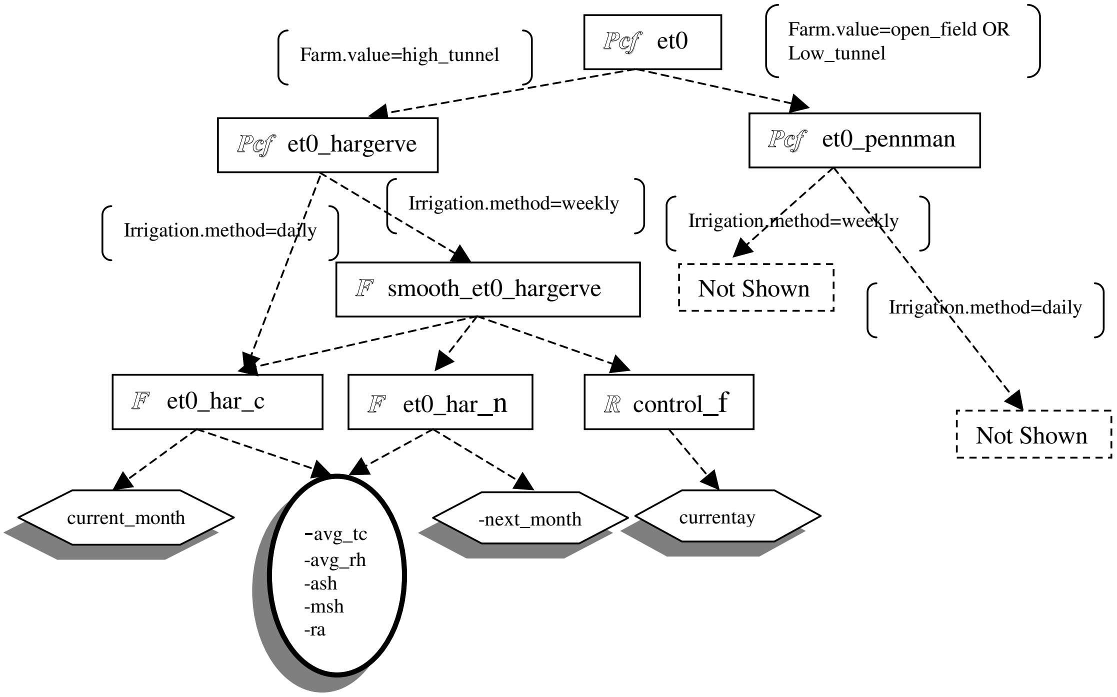


Figure 4: A simplified diagram of a section of the dependency graph employed

5. CONCLUSIONS AND FUTURE WORK

An irrigation scheduling expert system is a complex system that requires a lot of effort for building. A typical system will have a considerable number of concepts, relations, tables and equations built into it. By capturing re-usable knowledge across various irrigation systems, the required development time can be drastically reduced. This paper presented a tool that can achieve this for irrigation expert systems for vegetable crops in Egypt. Through this tool the process of developing an irrigation expert system is greatly expedited and the effort for developing such a system is greatly reduced. The developer does not need to concern himself/herself with the complexities of the built in model. All he/she has to do is focus on acquiring knowledge to determine the output a very small number of relations and tables. Of the thirty five relations built into the system, only 5 need be modified and of the eleven tables only 4, while none of the fifty two built in equations need be changed. The problem-solving model is also built in and hidden from the developer. This not only cuts down of the development time of an irrigation expert system, but also alleviates the need for a skilled knowledge engineer for developing a powerful system. By providing clear instructions for how to use the tool, almost any person with access to a domain expert can easily build an irrigation expert system. In fact, domain experts themselves can easily use this tool for research purposes. The tool was used for training non-computer specialists on developing irrigation expert systems as part of a one week workshop on expert system development in general. Only two days were designated for the purpose of training a developing a simple irrigation expert system. All workshop attendees were able to develop a working system within this limited time span.

Future enhancements to the tool will extend it to cover horticultural crops. Work is also currently being carried out in order to augment the developed tool with explanation facilities which is very likely to transform the tool into a powerful education instrument.

REFERENCES

Boner I., Parente A. and Thompson K. (1992). Knowledge-Based systems for crop management. In: *Fourth International Conference On Computers In Agriculture,* Orlando, Florida. American Society of Agricultural Engineers. .

Chandrasekaran,B. (1988). Generic Tasks as Building Blocks for Knowledge-Based Systems: The Diagnosis and Routine Design Examples. *The Knowledge Engineering Review* **3**, 183-210.

Doorenbos J. & Pruitt W.O. (1984. *Crop WaterRequirements*. Food and Agriculture Organization of the United Nations.

Edrees S., El-Azhary E. Tawifik K. Rafea A (1997). An Expert Systems for Citrus Production Management. In: *Proceedings of the Fifth International Conference on Artificial Intelligence Applications*, Cairo, Egypt.

El-Dessouki, A., S. El-Azhary, and S. Edrees (1993). CUPTEX: An Integrated Expert System For Crop Management Of Cucumber**.** In*: Proceedings of ESADW-93*, Cairo, Egypt. MOALR..

Forbus K. & Kleer J. (1993) *Building Problem Solvers*, pp 151-170. The MIT press, Cambridge, Massachusetts, London.

Hargreaves,G.H. and Z.A.Samani, (1982). Estimating potential evapotranspiration. *Journal of Irrigation and Drainage Division* **108**, 225-230.

Kamel,A., K.Schroeder, J.Sticklen, A.Rafea, A.Salah, U.Schulthess, R.Ward, and J.Ritchie, (1995). An Integrated Wheat Crop Management System Based on Generic Task Knowledge Based Systems and CERES Numerical Simulation. *AI Applications* **9**.

Ostergard M.and Goodell P. (1992). Delivering Expert Systems to Agriculture: Experiences with CALEX/Cotton. In: *Fourth International Conference On Computers In Agriculture,* Orlando, Florida. American Society of Agricultural Engineers.

Plant R. (1989). An integrated expert decision support system for agricultural management. *Agricultural Systems* **29**, 49-66.

Rafea A. (1998) Agricultural Expert Systems. In: *The handbook of applied expert systems* (J.Liebowitz (Ed)). CRC Press , LLC,USA.