NEPER WHEAT: AN INTEGRATED ARCHITECTURE FOR IRRIGATED WHEAT CROP MANAGEMENT

Ahmed Kamel, Kris Schroeder,
Eman El-Sheikh & Jon Sticklen

Intelligent Systems Laboratory
Computer Science Department
Michigan State University
East Lansing, MI 48824, USA
kamela@cps.msu.edu

Rick Ward, Joe Ritchie,
& Urs Schulthess

Crop and Soil Sciences Department
Michigan State University
East Lansing, MI 48824, USA
22857MGR@msu.edu
Rhines%Staff%CSSDept@banyan.cl.msu.edu
urs@psssun.pss.msu.edu

A. Rafea & A. Salah

Central Lab for Ag Expert Systems
Agricultural Research Center
Ministry of Agriculture and Land Reclamation
Cairo, EGYPT
ESIC%EGFRCUVX.BitNet@pucc.princeton.edu

Abstract: In this report, we discuss the development of an integrated system for irrigated wheat crop management in Egypt. The goal of our work is to develop a system that will address the various aspects of crop management including varietal selection, planting/harvest date selection, sowing parameters decisions, insect/disease/weed identification and remediation, irrigation/fertilization management and harvest management. The approach we take to solve this problem is the Generic Task Approach to expert systems development pioneered by Chandrasekaran et al. By using the Generic Task (GT) approach, we set out to model the behavior of an expert in wheat crop management. Previous research in the Generic Task approach has typically focused on the application of a single generic task problem solver to solve a particular problem. However, as a multi-task problem, wheat crop management calls for the cooperation of multiple problem solvers, each tackling a portion of the larger problem. Thus, wheat crop management provides a testbed for the idea of integrating multiple generic tasks together with the numeric simulation capabilities of CERES Wheat pioneered by Ritchie et al. We use the Knowledge Level Architecture proposed by Sticklen as the template by which we will organize our system.

Keywords: Expert Systems, Agriculture, Computer simulation, Problem solvers, Management systems.

1. INTRODUCTION

In this report, we discuss the development of an integrated problem solving architecture to capture all relevant aspects of crop management. Specifically, we discuss the development of an expert system to support the management of irrigated wheat in Egypt. Our system, named Neper[1] Wheat, tackles the problem of irrigated wheat crop management through the integration of expert systems technology with a well known crop simulation model. In particular, our work brings together the Generic Task approach to expert systems development (Chandrasekaran, 1986) and one of the premier crop simulation models used in agriculture today, CERES Wheat (Ritchie et al., 1985).

Neper Wheat addresses the various aspects of crop management including varietal selection, planting/ harvest date selection, sowing parameters decisions, insect/disease/weed identification and remediation, irrigation/fertilization management and harvest management. As the length of this list implies, crop management is a complex, multi-task problem. To manage the complexity inherent in this problem, the Generic Task (GT) approach proposes a process of task decomposition, with the assignment of a "method" to each subtask identified during the decomposition. These methods are the heart of the GT approach as described in section 2. Previous research in the GT approach has resulted in the identification of several such methods denoted "generic tasks". However, the GT approach is not limited by only the set of previously identified methods. For the problem of irrigated wheat crop management, we have identified CERES Wheat, as well as multiple generic tasks as the methods needed to perform the task of wheat crop management. The GT approach alone does not provide a template for the integration of these problem solvers. We will use the Knowledge Level Architecture proposed by Sticklen in (Sticklen, 1989) as the template by which we will organize our system.

This paper briefly describes the Generic Task, Knowledge Level Architecture and CERES Wheat methodologies, focusing on those aspects relevant to wheat crop management. In section 2, we give an overview of the Generic Task Approach and in section 3, we discuss the ideas of the Knowledge Level Architecture. Next, we discuss our approach to building Neper Wheat, our wheat crop management system for Egypt. We highlight the architecture of our overall system in section 4 and discuss the decomposition of the strategic planner into the modules that address the various facets of wheat crop management in section 5. We discuss the use of CERES Wheat during the planning process in section 6. Neper Wheat has several modes of invocation which are discussed in section 7. Finally, in section 8, we conclude the paper with a discussion of the advantages of our approach, as well as our plans for continuation of our research.

2. GENERIC TASK APPROACH

The Generic Task (GT) approach of Chandrasekaran and his colleagues is one of the earliest and one of the most fully developed of the task specific approaches to knowledge-based systems. The assumption of the GT approach is that there are certain basic "tasks" which make up complex problem solving. Associated with each task is an inferencing strategy (or method) which is capable of efficiently performing the task. The knowledge needed to perform each task takes different forms depending on the particular method identified to address the task (Chandrasekaran, 1986; Chandrasekaran, 1987).

The Generic Task approach sets out to identify generic tasks - basic combinations of knowledge structures and inference strategies capable of performing the tasks which make up complex problem solving. Research in the Generic Task approach has involved the development of expert systems tools to perform each individual generic task.

A number of generic tasks are currently available. However, for purposes here, the most significant are:

- **Structured Matching**. Structured Matching is a simple inferencing mechanism for performing inferences of the form: "If conditions A, B, C,... hold, then condition X holds." Rules of the previous form are organized into "structured" sets that correspond to some domain structure.

- **Hierarchical Classification** (Gomez and Chandrasekaran, 1981; Mittal, 1980). Hierarchical Classification (HC) is intuitively a knowledge organization and control technique for selecting among a number of hierarchically organized options. The abstract engine used for hierarchical classification, known as CSRL, was the first TSA shell and is described in (Bylander and Mittal, 1986).

- **Routine Design** (Brown and Chandrasekaran, 1986; Chandrasekaran et al., 1989). Routine Design (RD) was proposed by Brown as an architecture for performing design and planning tasks in which substantial experience is available (not for design or planning in totally novel situations).

Following the GT approach, when a knowledge engineer is faced with a new problem, he/she performs a task decomposition of the problem. Decomposition proceeds until a subtask matches an individual generic task, or another method is identified to perform the subtask. If a subtask matches one of the individual generic tasks, the knowledge structures and control strategies are specified. The knowledge engineer must only obtain the appropriate domain knowledge to fill in

---

1. The name "Neper" comes from an early Egyptian god of agriculture.

the knowledge structure. Having a pre-enumerated set of generic tasks from which to choose gives the knowledge engineer significant direction during the analysis phase of system development.

3. Knowledge Level Architecture

The problem of irrigated wheat management is a multi-task problem with several differing task types needed to solve the problem. Therefore, there is a need to integrate multiple Generic Tasks into one problem solver. The Knowledge Level Architecture (KLA) proposed by Sticklen (Sticklen, 1989) provides an organizational overlay to the basic Generic Task Approach to facilitate integration.

The KLA is based upon the Knowledge Level Architecture Hypothesis (KLAH). This hypothesis builds on what Newell proposed in his AAAI presidential address of 1980 (Newell, 1982). Newell's proposal was the existence of a distinct level of analysis for systems, the “Knowledge Level” which existed above the symbol level. What the Knowledge Level provides is a way of understanding a problem solving agent apart from the implementation of the agent. Although this allows deeper understanding of problem solving, Newell recognizes in his address that the behavior of an agent cannot always be predicted at the knowledge level. The reason for this deficiency is the lack of any discussion of problem solving control. The KLAH on the other hand, allows discussion of the control issue, but only in terms of *knowledge organization and control*.

Knowledge organization and control are captured in the Knowledge Level Architecture according to the Knowledge Level Architecture Hypothesis as stated in (Sticklen, 1989, p.343):

- **Knowledge Level Architecture Hypothesis**: A problem solving agent may be decomposed into the cooperative efforts of a number of sub-agents, the larger agent can be understood at the Knowledge Level by giving a Knowledge Level description of the sub-agents and specifying the architecture the composition follows.

This hypothesis leads to the specification of a system by explicitly representing the interactions between its agents. There are two defining aspects of KLA:

- First, there is a distinct message protocol that exists between problem solvers. The message protocol between two cooperating agents is defined in terms of the functionality of the agents. In other words, the protocol provides a means for the agents to request work (from other agents) and respond in a vocabulary that the other agents can understand.

- Second, to allow communication between cooperating problem solvers, communication channels are established. By decomposing the agent into subagents and fixing the communication paths, the KLA provides a way of organizing the knowledge of the agents.

These aspects provide a means of organizing the knowledge of differing agents. Furthermore, since control is passed to an agent only when another agent sends a request, the KLA provides a means of understanding the problem solving activity taking place among the cooperating agents of an integrated system.

4. INTEGRATED ARCHITECTURE

Previous research in the GT approach typically focused on the application of one generic task to one problem. However, analysis of the problem of irrigated wheat crop management reveals a multi-task problem, with several different task types needed to address the problem. Therefore, there is a need to integrate multiple generic tasks into one problem solver using the ideas of the Knowledge Level Architecture.

Figure 1 shows a high-level overview of the system. To understand the processing performed by the system, we describe the architecture according to the Knowledge Level Architecture paradigm upon which it is based. The architecture follows a classic organization for planning type problem solvers (i.e., generate, test, and critique). To perform planning in this manner, our system is composed of three cooperating agents, the Strategic Planner, CERES Wheat and the Plan Critic whose tasks are plan generation, plan testing and plan critiquing respectively. The strategic planning module uses compiled[2] knowledge of wheat crop management to propose a plan. Next, the CERES Wheat Module simulates the growth of the wheat under the circumstances set forth by the plan. Finally, the outcome of testing is handed off to the Plan Critic, where experience-based knowledge, possibly in the form of the farmer’s or extension agent’s opinion, can be used to critique the plan. If the plan is found unsatisfactory, a redesign request is sent to the strategic planner. The farmer may also be asked to change any preferences that have a negative impact on the plan’s performance. Finally, if the plan is approved, it is handed off to the farmer.

---

[2] Compiled level expertise refers to knowledge based on previous experiences.

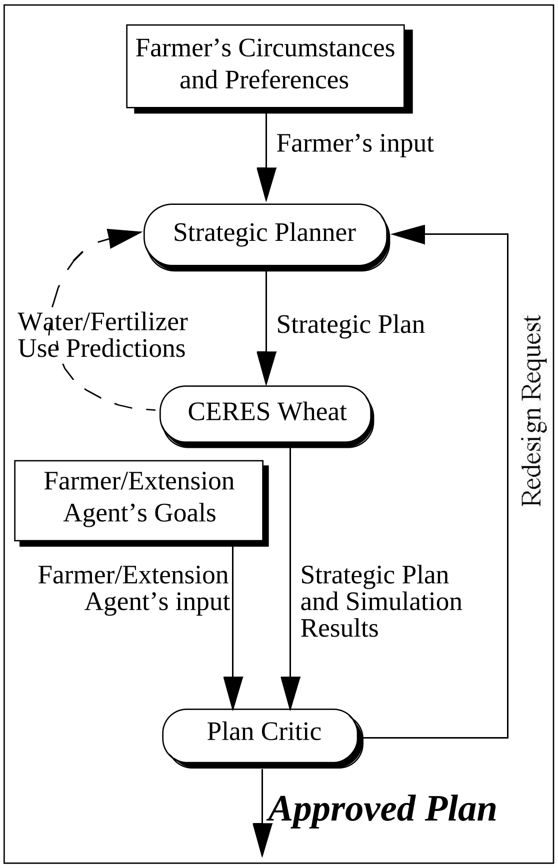

Fig.1. Irrigated Wheat Crop Management System

Our plan tester, CERES Wheat (Ritchie et al., 1985), is one of a family of dynamic process-orientated models which simulate the growth, development, and yield of major cereals. Our objective is to exploit the model's simulation capabilities as a dynamic knowledge base for prediction of input demand and yield as influenced by management decisions made by the strategic planner. The model is sensitive to crop management decisions including choice of variety, date of planting, fertility levels, and irrigation amounts and timings. With recently incorporated modifications, it can also simulate the impact of long-term climatic changes on yield, crop duration, and nutrient losses. Modeling can thus be used to evaluate long-term agricultural productivity and sustainability in light of management decisions.

5. STRATEGIC PLANNING MODULE

From an expert systems viewpoint, the heart of our system lies in the Strategic Planning Module. Therefore, we focus on this module and the development of a plan for the management of a wheat crop during an entire cropping season. To design this planning module, we followed the Generic Task and Knowledge Level Architecture approaches.

At the highest level, the task of creating a strategic plan is seen as a design problem - designing a plan for the management of a wheat crop. Thus, a Routine Design problem solver is incorporated as the top level problem solving agent and will act as a controlling agent during problem solving. After determining the top level problem solver, we decompose the task of wheat management into its component tasks. Through the process of task decomposition, we identify the task of wheat crop management as being composed of Varietal Selection, Planting Date Selection, Strategic Pest Management, Preplant Tillage, Planting Parameters, Fertilizer/Water Regime, and Harvest management. Analysis of each subtask brings forth the assignment of problem solving modules appropriate for each task as shown in Figure 2.

Figure 2 depicts the top level controller as a Routine Design problem solver. Routine Design performs a form of design known as parametric design. The goal of this form of design is as follows: given a set of predefined design parameters, assign values to the parameters to specify a complete design. To accomplish this task, a Routine Design problem solver makes use of hierarchical structures of design specialists to perform design, each specialist is responsible for a particular part of the overall plan. As Figure 2 shows, the specialists are not required to be Routine Design specialists. In our system, they are the problem solvers which best matched the subtasks defined in our task decomposition.

We view the individual modules, as well as the top level problem solver, as cooperating agents each performing a portion of the overall task according to the Knowledge Level Architecture paradigm. Communication channels are then set up between the cooperating agents. The communication channels define the input and output that will be passed between two problem solvers. Furthermore, the communication channels define the points of interaction between two agents. These points indicate when in the processing a problem solver will be invoked. This provides considerable explanation capabilities as to how problem solving will proceed.

In our system, the communication channels are straight forward. The top level Routine Designer, acting as a controlling agent, hands input to each problem solving module and receives output in a linear order. Thus, the top specialist, (Strategic Planning Module) makes use of the subspecialist (Varietal Selection, Planting Date, Strategic Pest Management, Preplant Tillage, Planting Parameters, Fertilizer/Water Regime, and Harvest Specialists) to perform the task of plan generation. In the following, we present an abbreviated description of each module from our system.

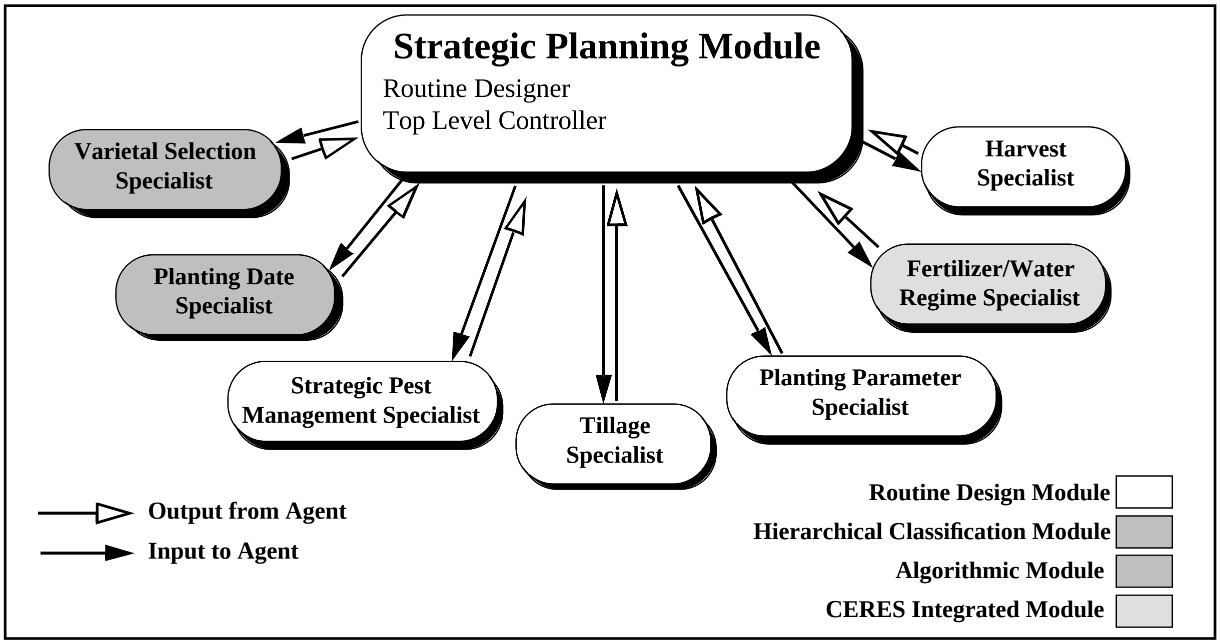

Fig.2. Strategic Planning Module

5.1 Varietal Selection Module

During the analysis of the wheat crop management problem, we identified varietal selection as a classification problem. Thus, we have chosen the generic task Hierarchical Classification (HC) to perform varietal selection. We composed a list of factors which experts consider when selecting a variety for planting. For example, the system must consider the desired planting and harvest dates, the region in which the field resides, as well as the type of wheat the farmer can market (either bread or durum). Furthermore, problems on the farm must also be considered including heat stresses, drought stresses, loose smut, rusts, and salinity. These problems are important since some varieties may be more resistant or susceptible to such stresses than others. A classification tree was then built which classifies available varieties in terms of these relevant factor.

When the user runs the system, he/she enters the circumstance on their farm. For example, the system will ask the user if there are rust problems or salinity problems on the farm. Furthermore, the farmer can enter preferences about various factors (such as planting date, seed color, time for maturity, etc.) as well as their willingness to change these preferences. The system will then classify the users situation to determine which varieties strongly match, match and/or weakly match their requirements. For example, if the user's field is located in the Reclaimed Desert Areas of Egypt and experiences problems with loose smut and yellow rusts, the system would classify these circumstances to find a variety that is appropriate for planting in the Reclaimed Areas, and is also resistant to loose smut and yellow rusts. Figure 3 shows a graphical depiction of the Information Processing Task of the Varietal Selection Module.

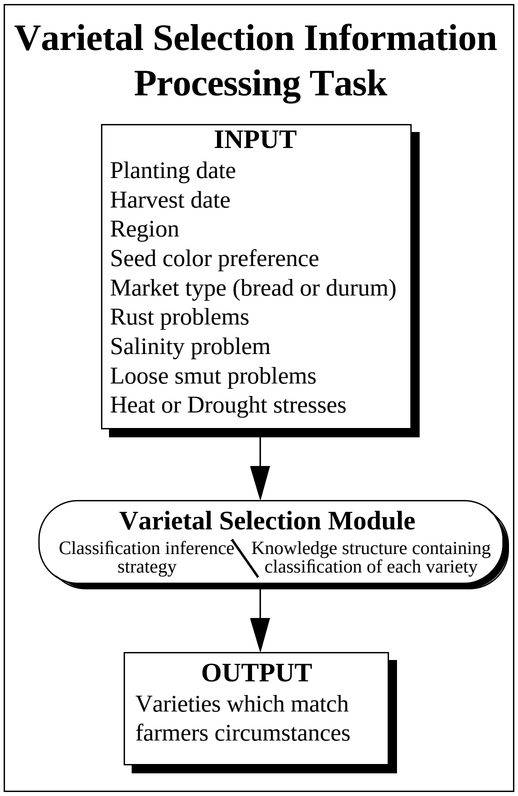

Fig.3. Varietal Selection Module

5.2 Planting Date Module

After choosing the appropriate variety, the planting date is selected. Planting date selection has been implemented as an algorithmic problem solver. The planting date determined at this point in the problem solving is the optimal planting date for the variety chosen by the Varietal Selection Module. Since an optimal planting date is associated with each variety, we use this date as our base recommendation. However, this date may be adjusted as needed by the other modules.

5.3 Pest Control Module

Strategic pest management includes the control of three factors: diseases, weeds, and insects. For each factor, the Pest Control Module analyzes the planting environment and constraints to determine the potential pest problems and the steps to control them. The Pest Control Module is implemented as a Routine Design problem solver which considers the history of weeds (monocotyledons and dicotyledons), diseases (loose smut, mildew and root rot) and insects (cut worms and crickets) in the field. This module must also take into account constraints on the solution, such as whether there is ample time to perform the treatment before planting, and whether the farmer has at his disposal herbicides and/or irrigation water to follow through with the recommendations.

Depending upon the problems discovered, the Pest Control module presents the user with recommendations to control the insect, diseases or weeds in the field. This module also determines whether wet or dry planting is appropriate given the circumstances on the farm.

5.4 Tillage Module

The health of the wheat crop is determined long before the seed is planted. One of the main determining factors is the kind and frequency of tillage applied to the soil. Tillage is usually done to prepare a seedbed, prior to planting. We have identified three levels of tillage, (no till, minimum till, and maximum till). Choosing a suitable tillage scheme is crucial to the healthy growth of the wheat crop.

The primary objective of the Tillage Module is to determine the most suitable type of tillage to perform based on the field and farmer's situations. The tillage schedule is based upon experiential knowledge (i.e., what would the expert recommend in a given situation). The goal is not to invent new approaches to tillage. Thus, the Tillage Module is implemented as a Routine Design problem solver whose goal is to design the tillage schedule.

To produce the optimal tillage recommendation, the Tillage Module takes into account a number of factors which determine the state of the field. These include the soil type/texture, weed history of the field, crop residues, soil moisture content, and type of planting (wet or dry). The module must also contend with constraints placed upon tillage by the farmer. These include the time allotted for tillage, equipment availability and monetary constraints. The output provided by this module includes the type of tillage, equipment recommendations and tillage schedule, as well as when fertilizers and herbicides should be applied in accordance with the tillage schedule. Finally, an estimate of total cost for tillage is presented to the farmer based upon the field size, the tillage schedule and estimated labor and equipment costs.

5.5 Planting Parameters Module

Before the wheat can be planted, various planting parameters must be determined. These parameters depend on the overall condition of the land and the wheat variety to be planted. They are determined by the Planting Parameters Module, which is implemented as a Routine Design problem solver. The primary objective of this module is to determine the planting parameters for the field including sowing method, seeding rate, sowing depth, land surface preparation and any planting date adjustments necessary.

To determine these parameters, the Planting Parameters module must consider a number of interrelated factors. For example, the type of sowing (whether broadcast or direct drilling) depends heavily on whether a seeding drill is available to the farmer, the soil moisture content and the condition of the field. To determine other parameters, such as seeding rate and sowing depth, the module must consider the type of sowing as well as the wheat variety, soil type, soil moisture content, and the germination rate of the seed.

5.6 Irrigation and Fertilization Regime Module

The goal of the Irrigation and Fertilization Regime Module is to present to the farmer a recommendation on the amount of water, nitrogen, phosphorus, and potassium needed by the growing plant throughout the cropping season. To avoid nutrient leaching in the soil, this module takes into account the wheat variety, soil type and region in which the farmer is planting to determine the applications which can most efficiently deliver the required nutrients without leading to deficiencies during the growing season.

The output of the module is the schedule for water and nitrogen applications including amounts and timings, as well as the amount of phosphorus and potassium necessary for the given soil type.

5.7 Harvest Module

The final stage of the cropping season is harvesting. The harvest module is implemented as a Routine Design problem solver which presents the user with recommendations on harvest timing, grain storage and straw handling. The module takes into account the cost of the various tasks included in the harvesting process and estimates the time and cost of the entire harvest. Constraints, such as availability of a combine, are also taken into account by the module.

6. INTEGRATION OF EXPERT SYSTEMS WITH CERES WHEAT

The next phase of the project involves bringing together compiled expertise (i.e., knowledge based systems problem solvers) and the numerical simulation capabilities of CERES Wheat. From a purely expert systems viewpoint, the problem we set to integrate compiled-level expertise and numerical simulation is receiving wide spread attention. The reason is largely due to the perceived “naturalness” of an interaction between “experience” knowledge and numerical methods. Knowledge-based systems are useful for decision making in situations where experience-based knowledge is readily available. Most knowledge-based systems, however, are based on qualitative, heuristic knowledge. While such knowledge can be very useful in the absence of more detailed quantitative models, quantitative models can typically produce an improved accuracy and thus should be used whenever possible.

In section 4, we described the use of CERES Wheat as a plan tester. However, this will not be the only function of CERES Wheat. Close examination of Figure 1 also shows data from CERES being directed back to the strategic planning module. As this indicates, CERES Wheat will also be used in plan construction. Currently the Fertilizer/Water Regime Specialist uses compiled knowledge of normal fertilizer and water usage in a region to base its recommendations. However, our objective is to allow the Fertilizer/Water Regime Specialist to also interact with CERES Wheat. The predictive capabilities of CERES Wheat will allow us to simulate the growth of the wheat crop assuming unlimited resources of water and fertilizer. By examining the amount of each resource used by the plant daily, the irrigation and fertilization needs of the plant can be predicted with greater accuracy. Actual amounts and timings are then scheduled based on these amounts. Adjustment are made to take into account not only the availability of irrigation and fertilization resources, but also processes which affect the resources when they cannot be applied at the same rate as plant uptake (e.g., evaporation and leaching).

The weather data upon which the simulations are based is predicted from past weather data. Since this prediction can never be completely accurate, the schedule for irrigation/fertilization may need modification during the season to compensate for irregularities in the weather. However, by bringing as much knowledge as possible to the task of Irrigation/Fertilization planning, we can make the best prediction possible.

7. MODES OF OPERATION

When utilizing Neper Wheat, the user is presented with several modes of operating the system. The most common mode of operation is to invoke Neper Wheat in “single design” mode. In this mode, the system will generate a single plan based on the farmer’s circumstances and preferences. The second mode of operation is to invoke Neper Wheat in “multiple design” mode. In multiple design mode, Neper Wheat generates a list of possible plans that are found to be suitable. These plans can then be ranked based upon other criteria, such as cost. Finally, the last mode allows each individual module to act as a stand alone expert in its own area (e.g., a tillage expert or an irrigation/fertilization expert). Thus, if the farmer prefers, he/she may invoke a single module to explore only a portion of the wheat crop management problem.

8. CONCLUSION

The work discussed above has proven several predicted advantages of following the Generic Task approach for the development of complex systems. The GT approach provides structure and direction throughout the development process. With multi-task problems, such as wheat crop management, an approach that provides flexibility in the tools available to address the problem is imperative, as was evident in this project.

The process of task decomposition is essential to managing the complexity of large scale problems. Through task decomposition, the GT approach allows the knowledge engineer to focus on each subtask and develop the problem solver for the subtask as a stand-alone problem solver. Integration is performed by setting up the communication channels though which input and output are passed to and from the problem solvers. The development of portions of our problem solver in this manner offered significant leverage throughout the development stages. The design of the problem solver was considerably simpler since it was possible to focus on a small portion of the overall task. Knowledge acquisition was more efficient and effective since both the knowledge engineer and the expert could dedicate their attention to a well defined subtask. Finally, testing proceeded more effectively throughout the development process since individual modules could be tested separately before being brought together into the overall problem solver.

Investigation into the KLA leads to a distinction between three levels of description of knowledge-based systems:

1. **The individual problem solvers:** A functioning system is an example at this level. The systems are defined in terms of objects specific to the domain. For example, for the NEPER system, the terms would include *planting date, variety, soil type*, etc.

2. **The problem solving types:** Since problem solving types span numerous domains, the systems are described at this level using problem solving specific, but domain independent terms. For example, classification is used in a number of different domains. Terms such as *classification hierarchy, established nodes*, etc. are used to describe the problem solver at this level.

3. **The Knowledge Level Architecture:** This is the highest level of description. The vocabulary used at this level is in terms of *agents, communication channels and message protocols.*

By describing a KBS in terms of these levels, a better understanding of the actual problem solving being performed is possible.

It is important to emphasize that the problem solving types from level 2 which are integrated into a KLA need not be from the Generic Task tool set. As the wheat crop management system shows, other problems solvers such as simulation models can also be incorporated easily into our system. Other Task Specific Architectures (TSAs) can be integrated as well. These problem solving agents can interact with the GTs along the communication channels to perform cooperative problem solving.

Our work also stems many future research directions related to wheat management. Neper Wheat can be considered prototypical for the management of other crops in Egypt. Thus, the next step is the development of a management system for other major crops. Furthermore, considering these crops in rotation is essential to the development of a robust crop management system. Although the present research focuses on agro-management in Egypt, applicability to other countries is also anticipated. For example, problems of water utilization and management directly affect sectors of US agriculture. In areas such as the Central Valley of California, water has always been a limiting factor. We observe agricultural interests increasingly being asked to justify water allotments. Thus, systematic, effective, and easily documented water management methods will become increasingly important.

9. ACKNOWLEDGMENTS

The Egyptian/American team which is performing this work consists of the two computer science groups led by A. Rafea in Cairo and Jon Sticklen at Michigan State University, and three agricultural groups: a simulation group at Michigan State led by Joe Ritchie, a wheat group at Michigan State led by R. Ward, and a wheat group at the Agricultural Research Center in Cairo led by A. Shafi. Without smooth integration between the agricultural groups and the computer science groups, this project would not be possible.

This research is supported by a joint USDA (OICD) / NARP funding. The Intelligent Systems Laboratory, MSU, receives substantial equipment support from Apple Computer.

10. BIBLIOGRAPHY

Brown, D. C., and B. Chandrasekaran (1986). Knowledge and Control for a Mechanical Design Expert System. IEEE Expert, 3(July), 92-100.

Bylander, T., & S. Mittal (1986). CSRL: A Language for Classificatory Problem Solving and Uncertainty Handling. AI Magazine, 7(2), 66-77.

Chandrasekaran, B. (1986). Generic Tasks in Knowledge-Based Reasoning: High-Level Building Blocks for Expert System Design. IEEE Expert, 1(3), 23-30.

Chandrasekaran, B. (1987). Towards a Functional Architecture for Intelligence Based on Generic Information Processing Tasks. In IJCAI-87, Milan.

Chandrasekaran, B., J. Josephson, A. Keuneke, and D. Herman (1989). An Approach to Routine Planning. International Journal of Man Machine Studies, 377-398.

Gomez, F., & B. Chandrasekaran (1981). Knowledge Organization and Distribution for Medical Diagnosis. IEEE Transactions on Systems, Man, and Cybernetics, SMC-11(1), 34-42.

Mittal, S. (1980) Design of a Distributed Medical Diagnosis and Data Base System. Computer and Information Science Department, Ohio State University.

Ritchie, J. T., D. C. Godwin and S. Otter-Nacke (Ed.). (1985). CERES Wheat. A Simulation Model of Wheat Growth and Development. College Station, Texas: Texas A&M University Press.

Sticklen, J. (1989). Problem Solving Architectures at the Knowledge Level. Journal of Experimental and Theoretical Artificial Intelligence, 1, 1-52.