A Case-Based Reasoner for Evaluating Crop Rotations in Whole-Farm Planning

by

Prabhakar V.V.Bhogaraju

Thesis submitted to the Faculty of the Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science

in Entomology

Approved:

____

Nicholas D.Stone, Chairman.

________________ ________________

David H.Vaughan Forrest W.Ravlin

July 31, 1996

Blacksburg, Virginia

A Case-Based Reasoner for Evaluating Crop Rotations in Whole-Farm Planning.

by

Prabhakar V.V.Bhogaraju

Nicholas D.Stone, Chairman

Entomology

(ABSTRACT)

I have worked on a Case-Based Reasoning (CBR) system that evaluates crop rotations for their soil erosion and risk of insect pest problems. The purpose of this system is to provide decision support for an automated whole-farm planner (CROPS). CROPS (Buick et al., 1992) generates crop rotation plans that can address some of the environmental, economic and legislative pressures facing natural resource managers. To generate and recommend a crop rotation plan CROPS requires estimates on the soil erosion risks and pesticide pollution potential of the crop rotation. In this research I have designed and prototyped a system that can assist CROPS in the process of whole-farm planning by providing information required for determining the soil erosion risks and the pesticide pollution potential of crop rotations.

Inputs for the system include: a crop rotation, its tillage and residue management practices, and field conditions. Soil erosion risk is quantified using the C-value. Pest risks are likelihood of pest outbreaks that require control in a crop rotation. CBR was the chosen methodology for system implementation. In CBR, solutions to new problem situations are derived from retrieving and adapting solutions to similar problem situations experienced in the past. The system was prototyped using Esteem™, a CBR development shell, and runs on a PC under the MS® Windows™ operating system.

To my mother, Lakshmi Bhogaraju, and the loving memory of my father, KrishnaRao Bhogaraju.

Acknowledgments

I thank my major professor Dr. Nicholas D. Stone profusely for his immense support and patient guidance during the course of this thesis. I thank Drs. Forrest W. Ravlin and David H. Vaughan for their services on my committee and their patience in dealing with my thesis. Special thanks to Dr. Rod Youngman in the department of Entomology and Dr. Jim McKenna in the department of Crop and Soil Environmental Sciences for their time and valuable information that went into this project. Thanks are also due to Drs. Lok T. Kok and Timothy P. Mack in the department of Entomology for their time and encouragement during the early stages of this project. I sincerely appreciate the support extended by my colleagues in the information systems and insect studies lab here at Virginia Tech.

Table Of Contents

List of Figures

List of Tables

Chapter 1.0

Introduction

1.1 General Background

Integrated pest management (IPM) has as its objective, “the development of improved, ecologically oriented pest management systems that optimize, on a long-term basis, costs and benefits of crop protection” (Huffaker and Smith, 1980). Whole-farm planning has a similar and encompassing objective for integrated land use management. It involves the development of long-term sustainable farm management strategies that lower the economic and environmental risks associated with agriculture. IPM is an important constituent of any such planning exercise as the farmer has to deal with pest problems and associated pesticide pollution risks of crop-production practices on the farm. Whole-farm planning integrates the principles of IPM, soil conservation, farm economics, and environmental protection to achieve its goals of sustainable and environmentally safe crop production. To successfully incorporate IPM principles into farm-planning, evaluating the consequences of alternative crop production practices on pest risks and their consequent pesticide pollution potential is essential.

Whole-farm planning is a complex process. Complexity arises because of the large number of factors involved and the combinatorial explosion of their allocation alternatives (e.g., Stone, 1995; Nevo et al., 1994). Because of this complexity involved in achieving an optimal crop plan, computer-based systems have been designed to automate the planning process. CROPS (Buick et al., 1992) is one such system. It utilizes the principles of sustainable agriculture to generate six-year whole-farm management plans. The CROPS planning system considers farmer goals and preferences, soil erosion risks, nutrient and pesticide leaching and run-off risks, and economic assessments when generating management plans. To lower the pesticide pollution risks in a whole-farm plan CROPS tries to take advantage of certain preventive IPM tactics. These include the use of crop rotation and other cultural practices like appropriate tillage and residue management. This is important from economic as well as environmental perspectives because crops and management practices greatly influence the incidence of pest risks in a rotation. As such in CROPS, proper selection of a crop rotation is an important part of generating a farm plan.

To evaluate the effects of practicing a crop rotation in a specified field, the CROPS system requires a heuristic or algorithmic method to estimate the pest problems likely to occur and their economic and environmental significance (Fig. 1). Because CROPS is an automated planning system, an automated tool is required for this purpose. However, up to now, no general solution to this evaluation problem has existed. CROPS has relied on static pesticide hazard ratings that depend only on the crops grown, not on the rotational sequences or field characteristics. The only alternative has been the development of a suite of expert systems (e.g., VICE-Corn: Buick et al., 1992; VICE-Wheat: Warren et al., 1995 Personal communication) for each of the crops included in the system. This approach is expensive and inefficient as it requires the use of several expert systems to evaluate each crop rotation. In this research I have designed and prototyped a novel system that can assist CROPS in the process of whole-farm planning by providing information required for determining the pesticide pollution potential of crop rotations.

1.2 Objectives

The overall goal of this research was to determine the suitability of case-based reasoning (CBR) for evaluating crop rotations for inclusion in farm plans. CBR is an experience-based reasoning methodology and is defined as, "a methodology that

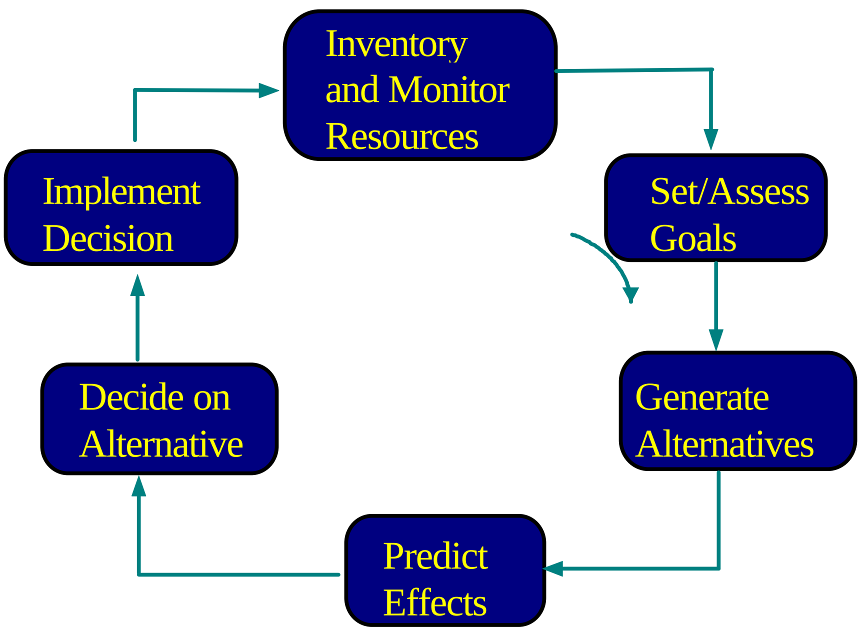


Figure 1.1 Decision cycle as implemented by CROPS in land use management and planning (Re-drawn from Stone, 1995), first at the field level and then the farm-level (inner arrow).

provides solutions to new problems by recalling and adapting solutions from similar situations in the past" (Kolodner, 1993). This technology has been successfully employed in various problem situations and this work was intended to demonstrate CBR in entomological and natural resource management domains.

Towards this objective I explored the concept of CBR, first to test its applicability as a reasoning method to evaluate crop rotations generally, and then to predict pest risks in crop rotations. I had the following specific objectives:

1. Test the CBR methodology on a straightforward problem: develop a system that can generate the soil erosion risk parameter for the USLE soil erosion model based on crop rotation as input. This system was called "***C-Chest***".
2. Develop a CBR system that could assess insect pest risks in any crop rotation for southwestern Virginia. This system was called "***Pandora***".

1.3 Background for Specific Objectives

1.3.1 Objective 1 - Soil Erosion Risk and CBR:

The influence of crop rotations on soil erosion is quantified by the cropping management factor (*C*-value) in the universal soil loss equation (USLE). USLE is a linear model proposed by the USDA, Soil Conservation Service and State experiment stations, to estimate average annual soil loss from crop-land. It is given by the equation:

$A = R \bullet K \bullet L \bullet S \bullet C \bullet P$, where

A = Annual soil loss (kg/m^2),
R = Rainfall erosivity factor,
K = Soil erodability factor,
L = Slope length factor,
S = Slope gradient factor,
C = Cropping management factor,
P = Erosion control practice factor,
(Wischemeier and Smith, 1958).

The cropping management factor is bounded from zero to one and represents the ratio of soil loss from a specific cropping or cover situation to the soil loss from a tilled, continuous fallow condition for the same soil and slope and for the same rainfall (Wischemeier and Smith, 1978). This factor includes the interrelated effects of cover, crop sequence, productivity level, growing season length, cultural practices, residue management and rainfall distribution. Rotations with higher *C*-values result in higher soil erosion.

Several of the parameters involved in determining the *C*-value also influence the pest incidence in a rotation. For instance, the *C*-value is dependent on the crop and its growth characteristics, weather patterns, tillage and residue operations, planting dates, and the sequence of crops in the rotation. These same variables, together with additional information on specific pest management practices, also play a major role in determining pest incidence in crop rotations.

One of the simplest tests of a CBR is to generate a numeric value that can be easily tested using objective methods. A crop rotation's soil erosion risk parameter is such a number. Building a system to estimate *C*-values was a test of the implementation scheme that was later applied to Pandora. Because the *C*-value is a constrained numerical value and because there are alternative objective ways available to calculate or determine *C*-values, the implementation scheme was easy to check for its validity.

1.3.2 Objective 2 - Insect Pest Risks and CBR

This objective is part of a broader study to suggest and evaluate alternative pest management options for a crop rotation. To arrive at these options, I must first identify or predict the pest problems that are likely to occur. Insects, weeds, and diseases form the majority of pests in cropping systems. In this research, I concentrated on insect pest problems associated with crop rotations. In insect pest management, applied agro-ecosystem planning should anticipate pest problems and find ways to avoid or respond to them (Metcalf and Luckmann, 1994). Therefore, this part objective was designed to develop a CBR system that could generate a list of potential insect pests that would likely require control in a particular rotation. Based on this information CROPS could evaluate each management option and determine control options, associated costs, and environmental risks.

1.4 Justification for this Research

This research directly contributes to the incorporation of preventive pest management strategies into whole-farm planning and emphasizes the role of IPM programs in land use management. IPM is the intelligent selection and use of pest control tactics to ensure favorable economic, environmental, and sociological consequences (Rabb, 1972). IPM has as its goals: (i) to minimize the economic impacts of pests, (ii) to produce sustainable solutions to pest problems, and (iii) to maintain environmental quality (Higley and Pedigo, 1991). The tactics employed in IPM programs towards realizing these goals can broadly be classified into two categories: (i) preventive, and (ii) therapeutic. Preventive tactics are those that are used to avoid potential pest problems while therapeutic tactics are used to remedy or ameliorate an existing pest problem (Wintersteen and Higley, 1993).

Many common components of IPM programs, such as sampling methods, developmental models, and economic injury levels (EILs), provide information necessary to implement therapeutic methods (Wintersteen and Higley, 1993). Few therapeutic tactics other than pesticides are available (Funderburk et al., 1993). Although IPM programs have made progress in addressing some of the concerns associated with dependence on pesticides, there remain inherent problems, particularly because of the continued dependence of IPM on therapeutic solutions (Hill, 1990). The alternative approach is to develop preventive solutions (Hill, 1990) and to use them in combination with therapeutic measures.

1.4.1 Preventive IPM

Preventive pest management tactics aim to prohibit establishment, limit growth, and reduce the seriousness of a pest's population (Funderburk et al., 1993). A good example is the use of cultural controls. Cultural controls are modifications of management practices that make the environment less favorable for pest reproduction, dispersal, and/or survival (Flint and Van Den Bosch, 1981). Crop rotation, sanitation, cultivation, time of planting, and residue management are some of the cultural forms of pest management that can be used as preventive measures.

Because preventive tactics are used to avoid potential pest problems (Funderburk et al., 1993), pest population assessment is not necessarily involved in immediate decisions to initiate the activity (Pedigo, 1989). Preventive pest management decisions are primarily focused on risk management. Since prevention involves extended-term action against a key pest or pest complex (Pedigo, 1989), the success of such actions is dependent on long-range planning. Such planning, which forms an important part of integrated crop management, can only be achieved by linking IPM with whole-farm planning to achieve integrated land use management.

1.5 The Problem Reformulated

The choice of crops in a rotation influences not only the pest problems, but also pesticide usage and associated pollution risk of that particular rotation. Prediction of pesticide pollution risks first involves the evaluation of the crop rotation, including its tillage and residue management practices, and field conditions for their pest risks. Subsequently, the management options for the predicted pests can be evaluated for their environmental pollution risks. In the process of planning, many alternate crop rotations and tillage practices must be considered and evaluated including potentially novel or unconventional practices. Therefore, the system developed in this research had to be able to handle both conventional and novel crop rotations and tillage alternatives generated by the planner.

1.6 Scope and Methodology

The direct results of this project are limited to agriculture in southwestern Virginia. The crop rotations represented and evaluated are those practiced or appropriate in this region. Estimates of pest risks are applicable in this region and partly to south-central Virginia.

The system hypothesis is that I can infer the pest risks and erosion properties of a given rotation by comparing it to closely related situations that I understand or have experienced. This is important because, during the process of planning, the plans generated may sometimes include novel or unconventional rotations and management practices. Effects of such rotations and practices, their *C*-values and insect pest risks, must still be evaluated by the planner.

CBR is a methodology that provides solutions to new problems by recalling and adapting solutions from similar situations in the past. It is a relatively new paradigm of artificial intelligence (AI) that has been successfully employed for planning, scheduling and decision support in various domestic (Kolodner et al., 1989), industrial (Hennessy and Hinkle, 1992), medical (Koton, 1989) and financial (Buta, 1994) applications. However, its use in natural resource management applications has been limited. In this research I present two CBR systems that aid in decision support for whole-farm planning, and in the process, make a case for CBR as a suitable methodology for the development of knowledge-based systems in natural resource management domains.

1.7 Report Structure

This thesis is organized into five chapters. Chapter two contains a literature review on whole-farm planning, the CROPS planning system, relevant and related decision support tools, and the suitability of a case-based approach for this project. A literature review on conventional and *ad hoc* methods used for validation and verification of knowledge-based systems, and some issues in the validation of CBR are also presented in this chapter.

Chapter three presents a design for a CBR system to generate insect pest risks and *C*-values for crop rotations. Based on the description of CBR in chapter II, a design is presented with detailed working examples. This chapter also includes discussion on sources for the materials, the processes used for knowledge (case)-acquisition, and a case-based representation of crop rotations, reminding schemes, and the adaptation processes.

Chapter four presents the full implementation of Pandora, a CBR system to generate insect pest risks, and C-Chest, a CBR to determine *C*-values of crop rotations using a commercially available CBR system. This chapter presents an overview of the software used, the materials and methods used in building C-Chest, results of this system and a discussion. This chapter also deals with the changes in design and implementation required by adopting this development environment.

Chapter five discusses validation and verification of the systems developed. Various quantitative and qualitative measures employed in evaluating this work are presented here. This chapter concludes with a discussion on the results and contributions from this research.

Chapter 2.0

Literature Review

Chapter two gives an overview of whole-farm planning, its requirements, a discussion on some of the existing software technologies available for decision support in planning, and their validation and verification procedures. The aim of this chapter is to provide a detailed background for the objectives and a rationale for the selection of CBR for system development.

2.1 Whole-Farm Planning

From a resource management perspective, whole-farm planning (WFP) is a systems approach to agricultural management that involves the development of farm management plans with careful consideration to environmental and economic issues.

It is defined as a holistic approach to conservation planning that deals with air, water, plants, and animals and their interactions either in a natural or managed ecosystem (Bridge, 1993). Planning at the farm level involves: integrating sustainable agriculture practices with conventional methods; reducing soil erosion, chemical/nutrient and pesticide leaching, and surface runoff problems; and considering farmer participation in federal farm programs (Buick et al., 1992).

From an economic perspective, WFP is an outline or summary of the resources available and the type and volume of production to be carried out (Kay and Edwards, 1994). The purpose of a WFP procedure is to provide a means of systematically evaluating alternative farm organizations (Boehlje and Eidman, 1984). WFP contributes to the development of a farm budget with expenses and income detailed on various farm activities. WFP and the resulting budget analyze the combined profitability of all enterprises in the farming operation (Kay and Edwards, 1994). Most planning techniques assume profit maximization as the primary goal, but this is often subject to a number of both personal and societal restrictions including long-term productivity of land, protecting the environment, guarding the health of farm workers and operators, and maintaining financial independence (Kay and Edwards, 1994).

Integrating the two slightly different view points mentioned above, WFP can be said to have three primary objectives: (i) to improve the long-term profitability of the farm enterprise, (ii) to provide environmentally safe management options to farmers, and (iii) to facilitate farmer compliance with legislative and other requirements (Stone, 1995). This can only be achieved when management strategies for the individual components of a farm-plan are designed in accordance with these objectives. Pest management is one important component of a farm-plan and IPM tries to provide a sustainable and ecologically-based solution for that part of the farming activity. WFP, therefore, encompasses IPM.

Formal farm-planning is rapidly becoming a necessity in light of legislative actions in response to public's concern over environmental pollution from farming practices (Stone et al., 1992). However, an optimal cropping plan is difficult to achieve because of the large number of factors involved, the complex interactions between them, and the enormity of allocation alternatives (Nevo et al., 1994). In light of the difficulties in achieving an optimal cropping plan, and the increasing requirements for formal planning (e.g., Chesapeake bay Preservation Act, 1988), computer-based systems are being designed for automating the planning process and for providing effective decision support in agricultural resource management (e.g., Buick et al., 1992; Nevo et al., 1994).

2.1.1 CROPS - An Automated Planning System

CROPS (Crop rotation planning system: Buick et al, 1992) is a computerized planning system that is designed to help farmers implement an economically sound multi-year crop production plan, while also maintaining environmental pollution standards (Buick et al., 1992).

It is a farm-level planner that uses the principles of sustainable agriculture to improve farm's profitability and long-term survival (Stone, et al., 1992). CROPS generates six-year crop rotation and tillage plans for specific crop/livestock operations using constraint-satisfaction methods and heuristic techniques stemming from research in the field of artificial intelligence (AI).

There are two distinct levels at which the CROPS system implements its decision cycle: (i) field level, and (ii) farm-level. CROPS treats its goals as constraints that are to be satisfied. Constraints are any factors that limit or influence the search for a plan (Buick et al., 1992). Field-level constraints are those factors specifically related to individual fields (Buick et al., 1992). In CROPS, these include: soil erosion, pesticide leaching and surface runoff, nitrate leaching, and the crop rotation (Stone, 1992). Acceptable crop rotations are defined by: climate and location of the farm; seasons and the seasonal requirements of the crops the farmer wants to grow; the pest problems in an area and how these can be managed via rotation; the effect of crops on soil fertility, field organic matter and the soil structure (Buick et al., 1992).

According to the decision cycle presented in chapter one (Fig. 1.1), CROPS first determines a set of possible rotation options for each field based on the farmer's economic goals, environmental risks like pesticide leaching, soil erosion and surface run-off of nitrates, and other requirements like base acreage requirements for federal and state programs. The key step in this process is the evaluation of each rotation (alternative), together with its tillage and management practices for various effects. In CROPS, this process is carried out by several ‘evaluator modules’, each of which evaluates the generated crop rotations for a specific effect.

CROPS evaluates soil erosion risks of a field using the universal soil loss equation (USLE - Wishchmeier and Smith, 1958) (Buick et al., 1992). Pesticide leaching and surface runoff risks are determined using a soil type and pesticides database (Stone et al., 1992). However, there is no comprehensive system available to assess pest risks and the associated pesticide usage/pollution potential of crop rotations at the field level. Currently CROPS relies on general and static hazard ratings of various pesticides that are normally used for various crops (Stone, 1995). These ratings are developed based on enterprise budgets developed by the cooperative extension service (VCE publication 446-0447, 1993).

2.2 Available Alternatives

There are several techniques that can be used for the development of automated tools employed in decision support (Table 2.1). There is considerable amount of discussion in literature on these various techniques employed for decision support in agriculture (e.g.: Coulson and Saunders, 1987; Stone, 1989; Engel et al., 1992). Databases, including geographic information systems (GIS) and management information systems (MIS), simulation models, operations research (OR) techniques, and knowledge-based systems (KBS) are some of the more prominent classes of techniques. The objectives for this project require the representation and reasoning about crop rotations. In the following sections these various technologies are discussed, along with their merits and limitations, from this perspective.

2.2.1 DBMS

Database management systems (DBMS) are one of the more commonly employed tools for manipulating databases. DBMS were developed in response to the need to integrate treatment of data (Coulson and Saunders, 1987). A typical example for a database application would be arriving at the crop yield based on location, soil type and other field features from a pre-defined table of values.

Table 2.1. Comparison of automated approaches for decision support (based on: Coulson and Saunders, 1987; Plant and Stone, 1991; Engel et al., 1992).

Approach	**Advantages**	**Limitations**	**Applications**
Simulation Models	Can describe complex biological and physical processes, including extensive numerical analyses.	Inability to select appropriate input values, and operate under incomplete information.	Stand-alone systems that can model optimum harvest dates for crops, insect threshold levels etc.,
Databases	Can efficiently store, manage and retrieve data.	No reasoning capability of the associated data.	Information delivery on pesticides, biological control agents etc.,
Operations research techniques	Can find optimum solutions to complex problems.	Inability to solve many non-linear problems, and inability to utilize qualitative data.	Economic analyses of pest management systems.
Knowledge-Based Systems	Can handle incomplete/ uncertain information, reduce combinatorial explosion by using heuristics, simple to develop and maintain	Validation of expert systems is difficult. Large systems tend to get unwieldy for maintenance and upgrading.	Pest identification, prediction, crop management and various forms of decision support.

Databases have no capacity for reasoning or analysis of the associated information. Their weaknesses lie in their inability to supply information (they supply data) or to reason with or about the data they contain (Engel et al. 1992).

Database systems are not considered to be problem-solvers, they merely allow access to a variety of information that the user might require to solve the problem (Plant and Stone, 1991). A separate inferencing mechanism is often necessary to analyze the data presented in a DBMS. For instance, a database on insect pest risks recorded on various crop rotations in Virginia can be used to provide information pertaining to a particular rotation or a pest on the crop rotations incorporated in the database. However, to apply the same information to new or unique rotations, databases will be inadequate. A system that can utilize the collected data and reason about their contents to assist in arriving at a decision for the new situation is required, and DBMS cannot offer such a capability.

2.2.2 Simulation Models

A model is any abstraction or representation of a system or process (Starfield and Bleloch, 1986). Simulation models permit the study of a real system without actual modification of that system in any way (McKinion, 1992). They are helpful in developing a quantitative understanding of the components of a system and in identifying components which require additional study and data collection (Shoemaker and Onstad, 1983).

Models have their strengths in defining a process and in making useful predictions. Because they can incorporate a large number of dynamically changing variables, simulation models are very useful in describing the dynamics of an agro-ecosystem (Shoemaker and Onstad, 1983). Engel et al., (1992) mention the ability of models to dynamically describe complex biological and physical processes, that include extensive numerical analysis, as one of their main strengths. In IPM, the simulation modeling approach provided a convenient and efficient way of abstracting complex systems in biology and economics (Coulson and Saunders, 1987).

However, models, like databases, do not solve problems. They are often components of larger systems that solve problems (Plant and Stone, 1991). To solve problems models must be linked as components to a decision support system (DSS). Simulation models are generally numeric and do not deal with qualitative knowledge. Handling uncertain and incomplete information, inherent in most agricultural systems, is yet another major limitation of this approach. The emphasis in the development of simulation models is on capturing the key processes in a system. This precludes models from acting as synthesizers of all our knowledge, and therefore makes them inappropriate as organizing paradigms for IPM (Stone, 1989). Further, IPM is a management system. In addressing management questions simulation models have some disadvantages, because a management strategy must be specified before the simulation is model is calculated (Shoemaker and Onstad, 1983).

To recommend a crop rotation plan, a fair number of alternative plans or rotations have to be evaluated, each for their pest risks. If a number of alternative management strategies are being considered, a simulation model must be re-computed for each possible alternative (Shoemaker and Onstad, 1983). This might be a computationally expensive approach. Further in the process of crop rotation planning, an integration of information on the effects of tillage, residue management, planting dates and the sequence of crops is required. Often such information is qualitative and either incomplete or uncertain. Simulation models might not perform well under such conditions.

2.2.3 Operations Research Techniques

Operations Research (OR) Techniques include a wide variety of techniques for representing problems mathematically and determining the optimal solution (Plant and Stone, 1991). Optimization techniques can, for example, choose the most economical policies without exhaustive simulation of each possible alternative (Shoemaker and Onstad, 1983). Linear programming is one of the most popular OR techniques, and is frequently used in economic analysis of systems. In this technique, the objective of a system is defined in a linear equation and the system is described by a set of linear inequalities or constraints. This is a problem solving technique as opposed to simulation models and databases, which serve more as information delivery tools.

Although the optimization methods are capable of analyzing a large number of management alternatives, optimization models are not well suited to analyzing a model with a large number of variables (Shoemaker and Onstad, 1983). According to Plant and Stone (1991), “Numerical methods have failed in crop systems because so much of our understanding about those systems is qualitative, based on experience; it does not lend itself well to mathematical representation”. Extending the same observation to whole-farm planning and IPM, where decision making often involves reasoning about qualitative data, suitability of OR approaches as decision support tools is limited.

2.2.4 Knowledge-Based Systems

Knowledge-Based Systems (KBS) are computer programs that use AI reasoning methods to solve complex problems within defined domains (Stone, 1989).

KBS are characterized by their unique problem solving methods that mimic human reasoning processes and rely on logic, belief, rules of thumb, opinion and experience (Plant and Stone, 1991). The KBS approach is one of the many outcomes of AI research. Rule-based expert systems are probably the best known and most widely used form of a knowledge-based system (Stone et al., 1986). These systems can handle incomplete or uncertain information, can reason with quantitative and qualitative data, and significantly reduce the computing complexity in a given problem by using heuristics or "rules of thumb" for reasoning. A brief discussion on some aspects of rule-based systems follows.

A rule-based system is a KBS where the knowledge base is represented in the form of a set (or sets) of *rules* (Hopgood, 1993). A rule is a conditional IF... THEN... statement (Fig. 2.1). Each rule represents a small chunk of knowledge relating to the given domain of expertise (Patterson, 1990). Rules in the knowledge-base are chained through recursively to reach a conclusion (or failure). This process could either be *forward chaining* or *backward chaining* or both depending on the system. Forward chaining is the name given to a data-driven strategy where rules are selected and applied in response to the current fact base (Hopgood, 1993). Backward chaining, on the other hand, assumes the existence of a goal that needs to be established or refuted (Hopgood, 1993) and goes on to look for facts that can substantiate its goal. There are numerous expert systems developed in the domain of pest management. A review on some of the systems can be found in Engel et al.(1992), and in Carrascal and Pau (1992).

Rule-based expert systems (RBES) have several strengths. Knowledge in these systems is separated from the reasoning procedures which makes updating of the knowledge base simple (Flemming, 1990). With their easily updated and maintained dynamic knowledge-base, reasoning facilities, and user interface, expert systems provide a way of delivering both qualitative and quantitative information to the user (Carrascal and Pau, 1992). They are transparent as they can explain how they have reached a conclusion by way of rules presented in natural language (Flemming, 1990). Though the rule-based approach is simple and powerful, a large number of rules is difficult to manage (Saarenmaa, 1992).

Evaluation of the pest scenario in a crop rotation is a problem involving several crops and several pests. To encode all the required knowledge in the form of rules would make the rule-base large and unwieldy. KBS reason from first principles (rules), with no basic understanding of the underlying model, for every session that is run. Their knowledge-base is relatively static in that they do not learn well. Further, the process of acquiring knowledge for the development of rule-based systems has been identified clearly as the bottleneck in developing these systems (Buta, 1994). To acquire knowledge on the effects of various crop and management practices would

1. IF tillage OF field IS minimum
AND
(previous crop cover OF field IS small grain
OR
previous crop cover OF field IS pasture left)
THEN
ArmyWorm risk OF field := High.

2. IF corn practice OF field IS continuous
AND
location OF field IS South Eastern
Virginia
AND
(soil Organic Matter OF field IS high
OR
soil drainage OF field IS poor)
THEN
BillBug risk OF field := High.

Figure 2.1 Rules in a RBES. Adapted from VICE-Corn (Buick et al., 1992).

mean repeated knowledge acquisition endeavors, and this has been proven to be a difficult task. Also, as RBES are narrow in their scope, the problem of evaluating a crop rotation would require the use of a suite of expert systems.

2.3 Case-Based Reasoning

Case-Based Reasoning is a method of developing knowledge-based expert systems that draws on examples of past experiences in the expert domain that can be applied to new problems (Buta, 1994). It is defined as, “a computerized method that attempts to study solutions that were used to solve problems in the past to solve current problems by analogy or association” (Katler, 1993). CBR has been proposed as a more psychologically plausible model of the reasoning of an expert than rule-based systems, and is claimed to be the essence of human reasoning (Riesbeck and Schank, 1989). CBR is an experience based reasoning technology and therefore performs well in domains with documented or available expertise.

2.3.1 History of CBR

Research in cognitive science and on the nature of human memory at Yale University and the works of Roger Schank on dynamic memory (Schank, 1982) form

Table 2.2. Some examples of CBR systems.

Name	Developers/Authors	Domain
CYRUS	Kolodner, J., 1983	Representation in memory and understanding.
MEDIATOR	Simpson, R. L., 1985	Dispute mediation.
JUDGE	Bain, W. M., 1986	Subjective assessment
CHEF	Hammond, K. J., 1989	Recipe planning.
CASEY	Koton, P. ,1989	Heart failure diagnostics.
CLAVIER	Mark,1989; Hennessey et al., 1992;	Autoclave loading designs.

the roots of CBR. One of the earliest CBR systems to be developed was CYRUS (Kolodner, 1983) which was based on the memory organization packet (MOP)'s concept put forward by Schank (1982). CYRUS was developed with a focus on how memory is used to answer questions of understanding, especially those related to Cyrus Vance's (former US. Secretary of State) diplomatic travels (Riesbeck and Schank, 1989). Several CBR systems have been developed since then in a wide range of domains including meal planning, medical diagnoses, legal analysis, and, design and configuration. A few examples of case-based reasoning applications are given in Table 2.2.

2.3.2 Terminology and Processes in CBR

The basic tasks in a CBR are: "input a problem, find a relevant old solution, adapt it" (Riesbeck and Schank, 1989). Its major processes involve remembering and adaptation (Chi and Kiang, 1991). Remembering is the process of retrieving a case or a set of cases from memory, and adaptation is the process of fixing an old solution to fit a new situation (Kolodner, 1994). The operation of a CBR is given in Fig. 2.2.

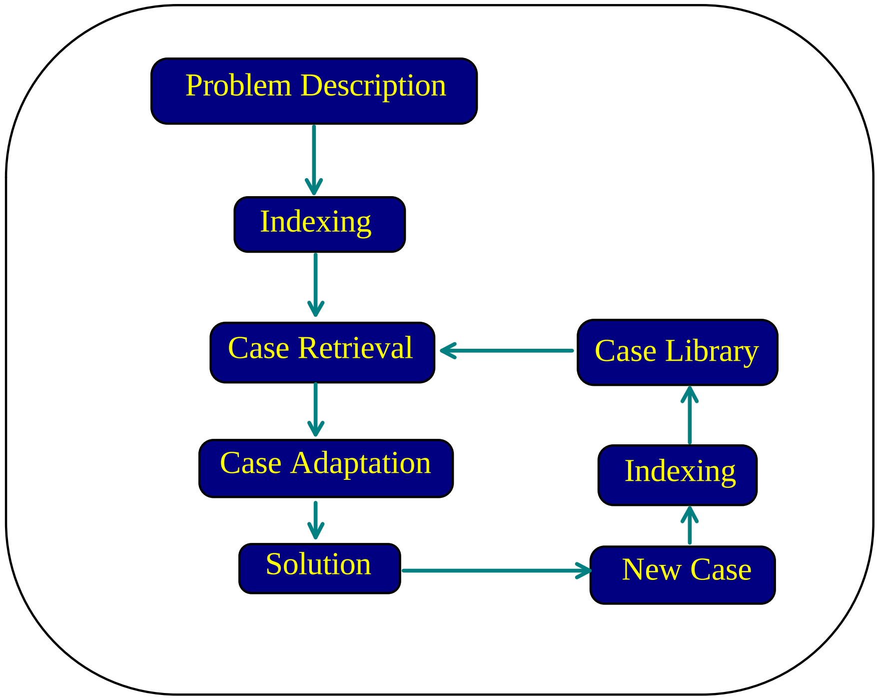


Figure 2.2 General flowchart of a case-based reasoning system (Redrawn from Plant and Stone, 1991).

2.3.3 Case Library

A case library, or case-base is merely a set of cases. A case usually denotes a problem situation or experience (Aamodt and Plaza, 1994) and its solution. It is defined as "a conceptualized piece of knowledge representing an experience that teaches a lesson fundamental to achieving the goals of the reasoner" (Kolodner, 1993). Cases have features that describe the contents of a case. Feature of a case is an *attribute-value* pair used in the description of a case (Kolodner, 1994). These features allow a uniform vocabulary in the descriptions of cases and facilitate comparison among cases for determining similarity in their values. Crop rotations, for example, can be represented as cases, with features including tillage operations, cropping sequence, residue management practices, *C*-values, insect pest risks observed, etc. Cases in a case library are organized according to their features. There are several models of case organization, but Schank's (1982) dynamic memory model, is one of the more commonly used. In Schank's scheme, cases are organized according to *indices*. Indices are combinations of a case's important descriptors, the ones that distinguish it from other cases (Kolodner, 1993).

2.3.4 Indexing

The process of assigning indexes to cases is called *Indexing* and is a fundamental step in the development of a CBR. Indexing involves assigning labels to cases designating under what conditions each case can be used to make useful inferences (Kolodner, 1993). They are assigned to cases to allow them to be recalled easily when an appropriately similar case arises (Plant and Stone, 1991). Indexing determines what extra, non-obvious, non-input features are needed for a particular domain (Riesbeck and Schank, 1989), and it allows a CBR to organize relevant cases together. Relevance is usually determined not by the obvious features of the input problem, but by abstract relationships between features and absence of features (Riesbeck and Schank, 1989).

For instance, consider a case library with two cases, one describing a two-year crop rotation of corn grain followed by rye as winter cover, and the other a one-year rotation of sorghum grain followed by oats as winter cover. The two cases have different 'crop sequences'. However, these cases are *similar* in that both of them involve a large grain grown in summer and a small grain raised as a winter cover. Thus, the underlying similarity is that they both have common 'crop descriptors' in their sequences and therefore behave similarly (in terms of *C*-values, for instance). This crop descriptor feature forms an abstract feature for these cases (and is derived from the crop sequence feature). Such abstract features often form the indexes for these cases. Schank's dynamic memory model proposes a structure for the organization of such important features and thereby allows efficient storage and retrieval of cases with similar indices.

2.3.4.1 The Dynamic Memory Model

In the dynamic memory model, memory is a flexible, open-ended system (Schank 1982; Riesbeck and Schank, 1989; Kolodner, 1993), and indexing is the key to using experience in understanding (Kolodner, 1993). The basic unit in dynamic memory is called the memory organization packet (MOP) and is used to represent knowledge about classes of events (Schank, 1982). MOPs have two functions: (i) they hold general knowledge, and (ii) they organize, in a complex hierarchy, specific experiences or instances of that general knowledge (Kolodner, 1993). A MOP contains a set of *norms* which represent the basic features of the MOP, e.g., what events occur, what goals are accomplished, what actors are involved, etc. (Schank, 1982). For example, in a crop rotation MOP, crops that are raised are the actors, tillage and residue management practices are the events that occur, and yield and economic returns from the crops are the goals that are accomplished. These are the norms of a crop rotation MOP.

MOPs are arranged in a network that includes hierarchical links with inheritance, such that searching the case library can go from the general to the specific (Plant and Stone, 1991), or vice-versa. Links in a MOP network may indicate *abstractions, specializations, instances,* or *scenes* for the MOPs (Fig. 2.3). Abstractions and specializations can be visualized as a network of MOPs arranged in a parent-child hierarchy. Abstraction links point from the specific to the more abstract or general knowledge, while index links point from the general to the more specific MOPs (Riesbeck and Schank, 1989). MOPs that are specializations of other MOPs inherit all the norms of the parent MOP.

Instances represent the highest degree of specialization in the MOPs structure and they can have no specializations. Instances can be visualized as examples of the MOPs and form the cases in a case-based system. A *scene* is a sub-event of a MOP and denotes the occurrence of an unique event associated with a given MOP or instance. For example, a MOP representing conventional tillage would have as a scene, a deep plow during the pre-season.

These various links and the MOPs network are illustrated in Figure 2.4. Here, the MOP *Corn* is a *specialization* under a *Large Grain* MOP, while the *Large Grain* MOP is an *abstraction* of the *Corn* MOP. A specific crop rotation of corn (denoted by a box) is an *instance* under the *Corn* MOP. The operation *No-till Planting* is a *scene* (denoted by an oval) in the *Sorghum* MOP as no-till is the common tillage practice in sorghum cultivation. Likewise *Hay harvest* would be a *scene* under an *Alfalfa* MOP or a *Crops for Hay* MOP.

Index links play a key role in determining the branch of the network to search for in a MOPs hierarchy. Abstract entities like *Grass Rotations* are linked to specializations like *Silage Rotations* by means of index links. An index link is labeled with an attribute-value pair (Riesbeck and Schank, 1989). In the above mentioned example, the attribute *Purpose* has a value *Silage* leading the grass rotation MOP to the silage rotation MOP.

The last category of links are the *failure* links. These involve an expectation failure (Riesbeck and Schank, 1989) in that they associate specific MOPs or instances to other MOPs whose norms they fail to meet. In Figure 2.4, the rotation involving barley, wheat and rye has a failure link pointing to the legume rotations MOP. This legume rotation MOP has as its norm the inclusion of at least one leguminaceous crop in a rotation. The rotation given in the example fails to meet this norm and therefore has a failure link pointing towards the legume MOP.

2.3.5 Case Retrieval

Case retrieval is a primary process in CBR (Kolodner, 1993). Case retrieval in a CBR system is supposedly analogous to *reminding* in human memory (e.g., Schank, 1982; Kolodner, 1989; Riesbeck and Schank, 1989). Case retrieval involves (i) The recollection of previous cases, and (ii) selection of the best subset in those recalled cases (Kolodner, 1993).

For reminding to take place, the key features in the new problem must be identified, and values must be assigned to these features so that an attempt can be made to match the new case with other cases (Katler, 1993). This is done by the *Indexing* process. During the process of reminding, the indices assigned to the key variables of a case are used to navigate through the MOP network. This matching is done to (i) reject cases that are too different from the input situation, and (ii) determine which of the remaining cases is most similar to the input (Riesbeck and Schank, 1989).

When a new case description is given and the best match is searched for, the input case is *pushed down* the network structure, starting at the root node (Aamodt and Plaza, 1994). Matching starts at the most abstract level in the hierarchy, and based on

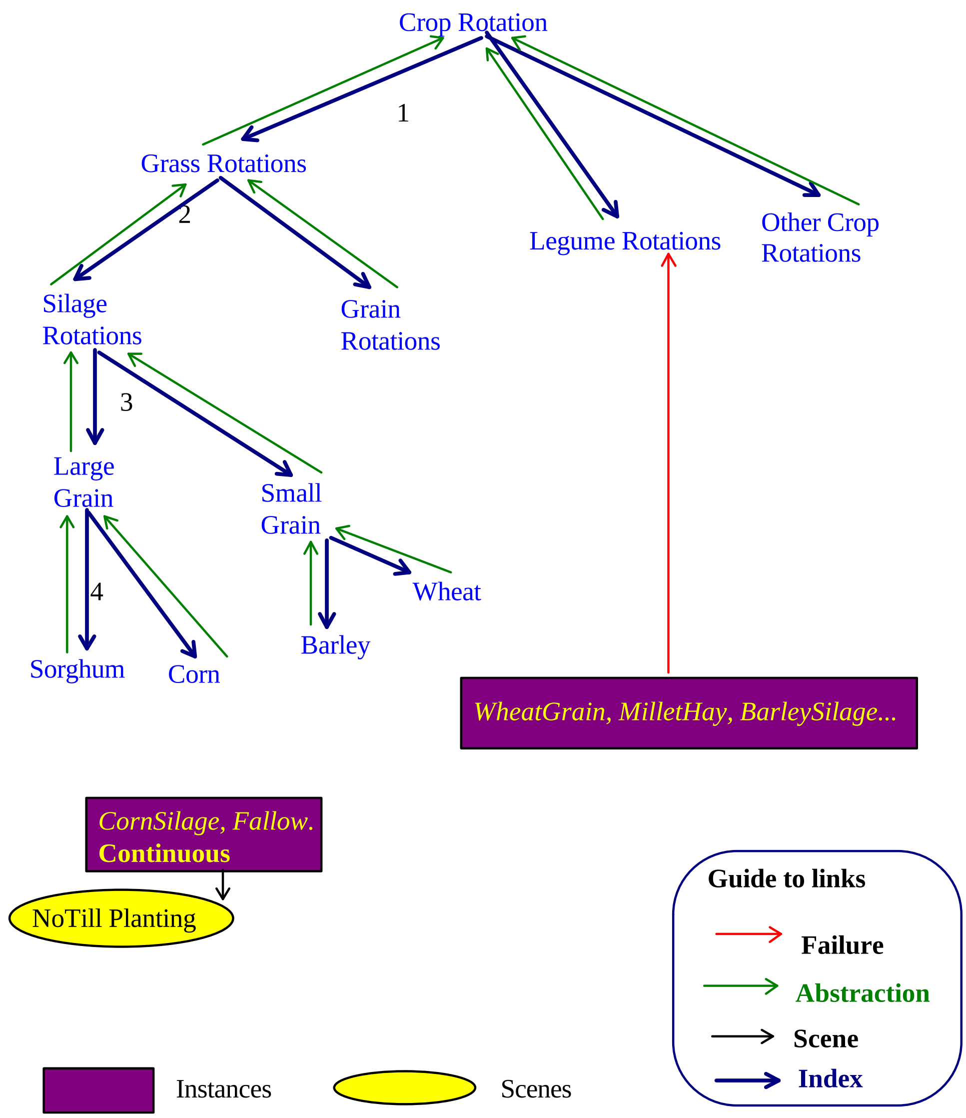


Figure 2.3 Dynamic memory organization. Numbers on the index links denote attribute-value pairs described in Table 2.3.

Table 2.3. Attribute-value pairs for the MOPs hierarchy presented in Figure 2.3.

Index no.	Attribute	Value	Other possible values
1	Crop family	Graminae	Leguminae
2	Purpose of crop	Silage	Grain
3	Type of crop silage	Large grain	Small grain
4	Type of large grain	Corn	Sorghum

the indices of the input case, the case is referenced to more specific nodes in the network until the most specific instance (case) is reached. If the input case does not match any of the specific instances of a particular node, the reminding stops at that point, bringing forward all the instances under that node as similar cases. The case(s) or instance(s) with most MOPs in common with the problem description is the most similar case and usually forms the basis for adaptation. CYRUS (Kolodner, 1983) and CASEY(Koton, 1989) are examples of CBR systems built on this theory of memory organization.

In the crop rotation example presented in Figure 2.3, when a input case involving grain corn and grain sorghum is encountered, first the indexing process determines the indices for this rotation. Values for the various attributes used in the hierarchy are sought in assigning indices to this case. In this example the relevant indices determined are given in the table below (Table 2.4).

Table 2.4 Indexing the input crop rotation.

Attribute	Value	Assigned index
Crop family	Graminae	Index1
Purpose of crop	Grain	Index2
Type of grain crop	Large grain	Index3
Type of large grain	Corn, Sorghum	Index4, Index5

the indices[1] generated for the input case Index1 leads the matching process to the *Grass Rotations* MOP. The input rotation satisfies the norms of this MOP as well, in having at least one graminaceous member in its crop sequence. Index2 then leads the system to the *Grain* MOP. Thus the system progressively steps down the network to the *large grain* MOP and then to the *Corn* and *Sorghum* MOPs. At this node, a more specific comparison of the input case takes place with the instances (cases) under these two MOPs to see if there is a case that is common to both the MOPS (i.e., with the same description as the input case). If there is an exact match, then that instance/case will be retrieved as the most similar case. If there is no exact match all the instances under *mixture* rotations will be retrieved and then are used for adaptation.

2.3.6 Adaptation

Adaptation is the process of modifying an old solution to meet the demands of the new situation (Kolodner, 1994). It involves reconciling the differences between the retrieved cases and the specific problem (Plant and Stone, 1991). The adaptation process looks for salient differences between the retrieved case and the input case and then applies rules that take those differences into account (Riesbeck and Schank,

[1]Indices are referred to by numbers purely for convenience in illustration. In practice indices can take on more meaningful names.

1989). This is perhaps the least structured and most difficult part of the CBR solution (Dutta and Bonissone, 1992).

During case retrieval, several instances (or cases) that are similar to the input case are brought forward. These cases are often ranked by some scale of similarity based on the number and extent of matches in their corresponding features. If there is an exact match with any one instance then the CBR process ends there suggesting the retrieved case as a solution to the input problem. This process is somewhat analogous to data retrieval in a DBMS. However, if there is no exact match in the case-library then the retrieved cases need to be adapted to suit the current situation. This is done by the adaptation process. There are two basic methods of adaptation: structural or transformational and derivational (Aamodt and Plaza, 1994).

2.3.6.1 Transformational (Structural) Adaptation

This is the standard method of adaptation in CBR systems. A retrieved solution is modified based on rules or critiques that compare the input case to the retrieved case. In the structural form of adaptation, rules apply directly to the solution stored in a case (Riesbeck and Schank, 1989). The past case solution is not directly a solution for the new case, but, there exists some knowledge in the form of transformational operators (stored in a rule-base) which, when applied to the old solution they transform it into a solution for the new case (Aamodt and Plaza, 1994). This form of adaptation is also referred to as the ‘substitution and transformation’ method.

For example consider as an input case, a one-year rotation of sorghum silage followed by oats as winter cover. Let the most similar case be a one-year rotation of corn silage followed by rye as winter cover. The transformational scheme first proposes all the insect pests in the retrieved case (corn and rye rotation) to the input case (sorghum and oats rotation). Then the CBR checks to see if any part of the solution (proposed insect pest risks) can be omitted or modified to accommodate sorghum in place of corn.

This is carried out by a set of adaptation rules that act on the selected (particular) case. However, the same set of rules apply to all the retrieved cases. The CBR system, therefore, has a general body of adaptation rules that would include information on how to adapt corn situations for sorghum. These rules transform the retrieved case solution to suit the input case. CHEF (Hammond, 1989) and JUDGE (Bain, 1986) are examples of systems with such an adaptation methodology. CHEF is a meal planner, while JUDGE specializes in subjective assessment.

Transformational adaptation works well when each problem parameter (input feature) is associated with one or more solution parameters (target feature) (Riesbeck and Schank, 1989). For instance, in the JUDGE (Bain, 1986) system, the solution (sentence) is determined by features like seriousness of the crime, criminal's motivation, degree of violence shown, etc. When a similar case is retrieved from the case-base, JUDGE first assigns the sentence from the retrieved case to the input case and then modifies that sentence. It does so by comparing the values of their features and identifying the difference. Increase (or aggravation) in the values of any of these features leads to an increase in the duration of the sentence and vice-versa.

2.3.6.2 Derivational Adaptation

In the derivational adaptation scheme, the solution for each case is stored as a set of rules or methods that are applied to inputs to get the solution. The retrieved case holds information about the method used for solving the retrieved problem including a justification of the operators used, sub-goals considered, alternatives generated, failed search paths, etc. (Aamodt and Plaza, 1994). During adaptation, the solution procedure of the retrieved case is rerun to produce the solution to the new case (Dutta and Bonissone, 1992). This method of adaptation is also referred to as derivational replay or derivational reinstantiation (Aamodt and Plaza, 1994).

Reinstantiation is used when the frameworks of an old and new problem are obviously the same but roles in the new case are filled differently than the roles in the old one (Kolodner, 1993). An example of a CBR system with such an adaptation scheme is MEDIATOR (Simpson, 1985). MEDIATOR is a system that specializes in dispute mediation.

When MEDIATOR is given a new problem where two children are fighting over an orange and is asked to solve the dispute, it reminds itself of a case where two sisters were fighting over a candy bar. It then associates the new case with this case based on the similarity in their frameworks (dispute over a common object). The solution in the candy bar case was to give the candy to one sister and the wrapper to the other (as that is what they wanted in the first place). MEDIATOR tries to apply the same plan to this solution and proposes peeling the orange and giving the peel to one child while the other has the rest.

The obvious advantage to this scheme of adaptation is that fewer rules (per case) are required to adapt the retrieved solution (Riesbeck and Schank, 1989). Further, such a scheme allows for adaptation across multiple domains if there is sufficient knowledge to derive the frameworks.

2.4 Learning in CBR Systems

One of the hallmarks of a case-based reasoner is its ability to learn from its experiences (Kolodner, 1993). Learning in CBR is accomplished primarily in two ways: (i) through the accumulation of new cases, and (ii) through the assignment of abstractions and/or indexes (Riesbeck and Schank, 1989; Kolodner, 1993). By retaining cases posed to the system, a CBR will have more instances in its case-library and therefore increase its potential of providing a quicker and correct solution. According to Kolodner (1993), "a reasoner whose cases cover more of the domain will be a better reasoner than one whose cases cover less of the domain". Some of the earlier CBR systems like CYRUS or IPP employ the case retention method of learning.

A more significant form of learning occurs when new indices (and MOPs) are created as a result of accumulating new cases. New MOPs are formed when a number of cases are discovered to share some common set of features (Riesbeck and Schank, 1989). As several similar experiences are encountered, their similarities are extracted to create new general memory structures (MOPs) (Kolodner, 1993). The common features of these similar cases are used to create the norms of the new MOPs, while the unshared features are used as indexes to the original MOPs (Riesbeck and Schank, 1989). This method of learning is also referred to as similarity-based generalization (Reisbeck and Schank, 1989).

For example, consider a "small grain-legume" rotation MOP which has as its instances: (i) crop rotations of soybeans double croppped with winter wheat, and (ii) soybeans double-cropped with barley. If an input case of soybeans double-cropped with barley is posed to the system the reminding scheme has to consider all the instances under this "small grain-legume" MOP for similarity assessment. This will include rotations of soybeans double-cropped with wheat as well as those double-cropped with barley.

In systems with learning mechanisms, if the CBR encounters more than one instance of soybeans double-cropped with barley rotation then it creates a new MOP called "barley-soybeans" MOP which will be a specialization under the "small grain-legume" MOP. All the rotations of soybeans and barley will be indexed as instances under this MOP. When a new case of soybeans double-cropped with barley is encountered the reminding scheme leads to this newly created MOP. Instances under this MOP will be considered for reminding as opposed to all the instances of small grain-legume rotations in the earlier case. This process reduces the number of instances to be compared during similarity assessment and increases the speed of the reasoner.

2.5 Advantages of CBR systems

By far, the most frequently cited advantage of the CBR technology is its power to solve problems in poorly defined domains. Ashley and Rissland (1988) mention that CBR is used to capture expertise in domains where the rules are ill-defined, incomplete or inconsistent. Dutta and Bonissone (1994) claim that CBR is useful when domain rules are expensive and hard to formulate. Thus, CBR expands the expert system concept to areas with incomplete or contradicting information (Kolodner, 1991). Also, many real world domains are so complex that it is either impossible or impractical to specify fully all the rules involved. On the other hand cases (i.e., solutions for problems, can always be given) (Riesbeck and Schank, 1989).

Other advantages of CBR include the speed of development of the application and a faster reasoning process. Creation of a case-base is usually more rapid than the creation of a rule-base (Katler, 1983). This is attributed to the nature of human memory which is supposed to be more of a library of experiences than a set of rules or principles (Riesbeck and Schank, 1989). Therefore, the bottleneck of knowledge acquisition in most knowledge-based systems is overcome in this approach. In CBR, the problem-solver makes its inferences based directly on previous cases rather than by the more traditional approach of using general knowledge (Kolodner, 1989). Thus, solution generation by a CBR tends to be faster since I adapt a prior solution as opposed to reasoning from first principles.

Further, Case-based systems are maintained and expanded easily by the addition of cases (Buta, 1994). The ability to retain cases and form new generalizations is one of the main strengths of CBR systems. This method of incremental learning results in increased efficiency in familiar situations and allows a CBR to cope with problematic situations (Kolodner, 1993).

2.6 Limitations of CBR systems

In general, a CBR system will be able to generate a better recommendation if it has a larger rather than a smaller case base (e.g., Ruby & Kibler, 1988; Goodman, 1989; Gaines, 1991). Therefore, a large collection of cases might be necessary for developing an effective CBR system. A CBR system might be tempted to use old cases blindly, relying on previous experience without validating it in a new situation and it might allow cases to bias it too much in solving a new problem (Kolodner, 1993). Such situations require stringent indexing and similarity assessment. Such similarity assessment is not structured in the pest management domain as there are no established objective measures of similarity between crops or pests.

In the process of learning, a CBR might try to retain all the cases it encounters and increase its case-base quickly. This might reduce the efficiency of the reminding schemes if the cases are not significantly different from one another. Further, in dynamic memory, formation of spurious or silly generalizations is a potential problem (Riesbeck and Schank, 1989). There are several control measures that can be used to overcome these limitations. By employing control measures like setting a threshold number of instances required for forming a generalization the problem of spurious generalizations can be overcome. Alternative learning methods like explanation-based generalization (Mitchell et al., 1986; DeJong and Mooney, 1986) can also be used to control the creation of generalizations.

Validation and verification of CBR systems is a difficult process. As CBR is suitable for domains where rules are ill-defined and/or hard to formulate, it may be difficult to establish expectations or standards of behavior in those systems (O'Leary, 1993). Further, most CBR systems add solved cases to their case-base. Such learning mechanisms can impact a CBR's performance, and thus, validation results (O'Leary, 1993). However, some of the conventional methods of validation and verification used in KBS development can be extended to CBR systems.

2.6 Evaluation of CBR systems

The term *evaluation* is used to refer to all procedures applied to ensure that a model is appropriate for its intended purpose (Harrison, 1991). In general, *validation* and *verification* comprise the complete evaluation of a model. Verification tests are aimed at building the system right (correctly), and validation tests are aimed at building the right system (O'Leary, 1993). Virtually all the research in verification and validation has been focused on rule-based systems rather than other knowledge representations, such as case-based systems (Gupta, 1991, in O’ Leary, 1993).

2.7.1 Verification

Verification is defined as, "the demonstration of the consistency, completeness and correctness of the system" (Adrion et al, 1982). The basis of comparisons for verification is the knowledge representation and the knowledge stored in those representations. For verification of a case-based system, O’ Leary (1993) proposes a structural approach to domain independent verification of cases . This scheme follows a four-step procedure to verify the functioning of the system.

- Check for consistency: Consistency check in case-based systems is primarily a test of the vocabulary used in the description of cases. Misspellings, different names for the same object, etc. are some of the causes for errors in consistency. Such inconsistencies might lead to improper similarity assessment.

- Eliminate redundancy: Redundancy occurs if the case-base stores the same case more than once. Repeated instances can lead to errors in maintenance and also lower the efficiency of the search process.

- Assure completeness: Completeness in the verification of cases requires that the cases that are planned to be included in the system are included, and that for each of the cases a complete case specification is given. It is concerned with the possibility that knowledge or cases are omitted.

- Ensure correctness: Correctness refers to determination of whether or not there are any ascertainable errors in the knowledge. For example, *C*-value of greater than 1 is an incorrect representation.

The chief advantage of O'Leary's scheme is that it is domain independent. Any CBR system can be subjected to the above mentioned tests for verification. Further, these tests are simple and can be easily carried out during CBR development.

2.7.2 Validation

Validation is the determination of the correctness of the final program or software produced from a development project with respect to the needs and requirements of the end user (Adrion, 1982). Validation of KBS is a difficult problem. Objective validation, using statistical tests or linear regression analysis of model output, will rarely be appropriate for validating such systems because they deliver qualitative advice rather than quantitative results (Jones, 1992). Harrison (1991) suggests several subjective tests:

- Ascertain face validity: In this test whether or not the output seems reasonable is determined. The establishment of face validity in effect means a subjective evaluation of the reasonableness of output from an expert system in its own right.

- Compare against high quality performance: For similar data, the system output is compared to that of a panel of experts. This test is used to compare outputs of the expert system over a number of test cases with a reference level of expertise.

- Target standard performance: In this test, the system must achieve a minimum standard of performance. An absolute or relative measure of the system's success is set as a threshold for satisfactory performance in this procedure.

- Use the Turing test, a double-blind test of the system and the experts: In this procedure the output from the system and from a panel of experts are presented to a second panel of experts for distinction. If there is no distinguishable difference in the recommendations generated by the system and that by the experts the system is then deemed to have passed the 'Turing' test.

Some of these validation tests require multiple experts for comparison on qualitative aspects. This is a potential limitation in the development of knowledge-based systems. However, validation tests like face validity and comparison against high quality performance can be implemented with one or few experts.

2.8 Summary for chapter two

CBR meets the requirements for this project better than the other modeling paradigms mentioned in this chapter. The qualitative and often incomplete nature of the information to be processed makes it difficult to use either simulation modeling paradigms or optimization procedures. Because the project requires inferencing capabilities built on a database (case-base), the database management system paradigm alone would not be suitable. A knowledge-based approach is the most appropriate of the techniques due to its ability to handle qualitative and incomplete information. However, the most common method of KBS development, the rule-based approach, would be inadequate for the purpose of this project owing to the bottlenecks in knowledge acquisition and the volume of rules that are required for reasoning. Use of CBR in this project is further supported by the availability of case-bases (e.g., NRCS data) for system development, and tools (e.g., RUSLE, VICE-Corn etc.,) to evaluate the developed CBR systems.

Chapter 3.0

System Design

3.1 Overview

Case-based reasoning (CBR) is the primary representation and reasoning methodology in this project. In this chapter a design for developing a CBR system that evaluates crop rotations is presented, including sections on developing a case-base, indexing and retrieval, and adaptation. Chapter four describes the implementation of this design using a commercial CBR development tool.

The overall goal of this research was to determine the suitability of CBR for application in the whole-farm planning process. I intended to accomplish this first by developing a design based on some of the models suggested in the literature, and then implement a CBR system for my objectives. The purpose of this design was to identify some of the issues and challenges involved in the representation and reasoning about complex crop rotations, and to provide a framework for the development of other similar CBR systems. In conventional software development, the design process dovetails with the implementation phase, with few modifications to the design once coding begins. In this case, however, the design and implementation processes differed significantly because of certain restrictions imposed by the selected software and are therefore presented as two separate topics in this report.

3.2 Case Library

The basic components of a CBR are: (a) a case library, (b) a reminding scheme, and (c) an adaptation procedure. A case library forms the knowledge-base of a CBR system where knowledge is stored in the form of cases arranged in hierarchies of MOPs (Riesbeck and Schank, 1989). To design a case library I first had to identify the features that are required for representing a case and then index the cases according to the relationships between them.

3.2.1 Case Representation

Since each case describes a crop rotation, the various components of that crop rotation can be used as features for the cases. Key factors influencing soil erosion (i.e., factors that influence the "cropping factor", *C*, in the USLE soil loss model) and insect pest risks for crop rotations are some of the features I identified in this process. Selection of these features was based on literature reviews and on interaction with experts (Drs. Youngman and Kok, Department of Entomology, Virginia Tech).

C-values for crop rotations at a given location depend on the crops being grown, their growth patterns, tillage, and residue management practices (Brady, 1990). Therefore, these formed the key features in a case for determining the *C*-values. In addition to the factors mentioned above, insect pest risks are also influenced by field conditions such as the soil organic matter content, prior pest history, field drainage, and weed populations. Cases, therefore, require these extra parameters as features (Table 3.1).

Since the *C*-value is dependent on crop growth parameters, crops that are closely related taxonomically or that have similar growth patterns tend to have similar *C*-values. For example, *C*-values for wheat, rye and barley for the same set of management practices are almost the same (± 0.01) (Tables 3.2a, 3.2b). Likewise, certain insect pests tend to show affinities along taxonomic characters or management practices. For example, white grubs are more common when corn is preceded by a leguminous crop the previous summer, and wireworms are more common when corn is sown into dense sod (VCE Publication 456-016 for field crops, 1993). Accordingly, several features indicating taxonomic affinities, seasonal requirements, and management practices were added to the cases. These features include: crop descriptors (e.g., describing the crop as a large grain, small grain or a legume etc.,), crop seasons (indicating the crop growth period), crop families (indicating taxonomic affinities of the crops), and tillage descriptor (describing the nature of tillage operation as either a conventional tillage or conservation tillage practice).

Weather parameters play an important role in determining the risks associated with insect pests. However, these parameters are not explicitly included in this design. Because the primary focus of this project was to demonstrate the suitability of CBR, seasonal variability of weather was not taken into consideration in order to reduce the complexity in design. Further, the planner in CROPS deals with the average or the expected case and does not necessarily generate plans from a risk management point of view. Therefore, the CBR proposed in this design is based on the average case or reasonable expectation.

Table 3.1. Key attributes in defining a Crop Rotation as a Case.

Attribute / Feature	Description
Rotation Sequence	List of crop names and their sequence in the rotation.
Tillage Sequence	Tillage type for crops in the rotation.
Residue Management	Residue management practices for various crops in the rotation
Rotation Length	Duration of the crop rotation
Field Organic Matter	Organic Matter content of the field (Low, Medium or High)
Field Soil Type	Soil type (for determining drainage)
Prior Pest History	List of insect and other pests that were a problem in that field.

Table 3.2a Output from RUSLE showing *C*-values for a soybeans and rye rotation.

RUSLE Version: ARS30.1

- Inputs for C-Factor -

city code: 33001 ASHEVILLE, NC
adjust for soil moisture depletion: NO
percent surface covered by rock fragments: 0
surface cover function; B-value code: (1) normal conditions

crop	start date	end date	percentEI	C-factor
soybeans	6/26/1	10/25/1	58.5	0.028
rye cover crop	10/25/1	6/26/2	41.5	0.017
Rotation C Factor = 0.045				

crop	start date	end date	percentEI	C-factor
soybeans	6/26/1	10/25/1	58.5	0.024
winter wheat	10/25/1	6/26/2	41.5	0.014
Rotation C Factor = 0.039				

Table 3.2b. Output from RUSLE showing *C*-values for a soybeans and wheat (double-crop) rotation.

3.2.2 Case Indexing

Once the features for case representation are identified, the next step in the design of a CBR is indexing. Based on the dynamic memory model (described in section 2. 3. 4. 1), I develpoed a hierarchy of crop rotation MOPs for this CBR system. This hierarchy, called the rotation hierarchy (Figure 3. 1), forms the basis for the indexing scheme, in which cases are indexed as instances of its various MOPs.

3.2.2.1 MOPs and their norms

Norms for each of the MOPs in the rotation hierarchy (Table 3.3) are used to classify the crop rotations. For instance, the norm for the top-level crop-rotation MOP is that it has a sequence of crops raised over a number of years. Its specializations, e.g., the rotations-with-grasses MOP and the rotations-with-legumes MOP, inherit norms from the crop-rotation MOP. In addittion, these MOPs have other norms that differentiate them from one another. For instance, the norm for the rotations-with-grasses MOP is that the rotation must include a crop belonging to the family Graminaceae. Therefore, the rotations-with-grasses MOP must have two norms: (i) a sequence of crops raised over a number of years, and (ii) contain at least one graminaceous crop.

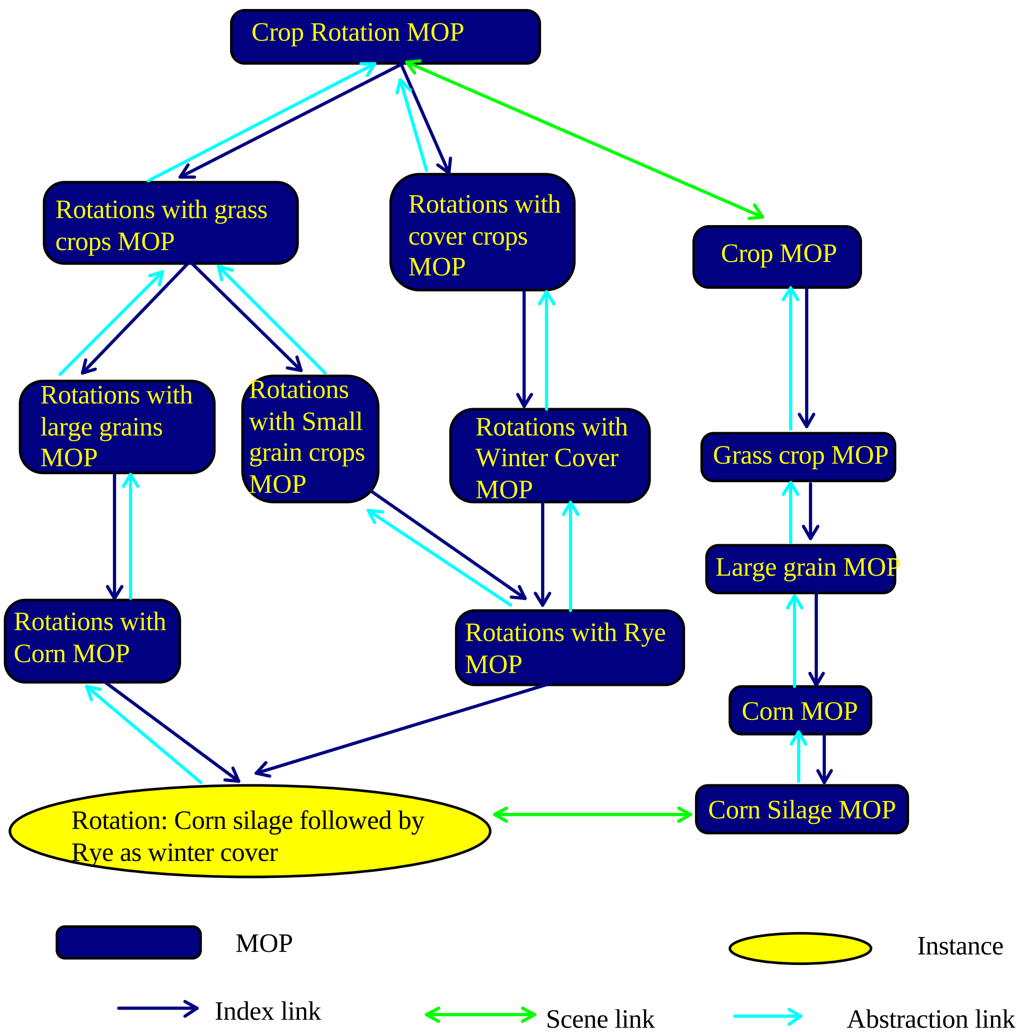


Figure 3.1. **Rotation hierarchy.** Schematic arrangement of crop rotations as MOPs in dynamic memory depicting a shared MOP and various links.

Table 3.3. Norms for some of the higher level MOPs in the crop rotation hierarchy.

Name of the MOP	Norms	Immediate abstraction
Crop Rotation	Sequence of crops and practices.	None. This is the ROOT MOP.
Grain Rotations	Rotations with crops raised for Grain purposes only.	Crop rotation MOP.
Silage Rotations	Rotations with crops raised for Silage purposes only.	Crop rotation MOP.
Legume rotations	Rotations with legume crops.	Crop rotation MOP
Rotations with Meadows	Rotations with some years under a meadow.	Crop rotation MOP.
Rotations with hay	Rotations with at least one hay crop	Crop rotation MOP
Rotations with cover	Rotations with cover crops.	Crop rotation MOP.
Large grain rotations	Rotation with Corn and/or Sorghum	Grain Rotations MOP.
Small grain rotations	Rotations with Wheat, Barley, Rye, Oats, and Millets.	Grain Rotations MOP.
Mixed grain rotations	Rotations with combinations of large and small grains.	Grain Rotations MOP.
Large grain silage rotations	Rotation with Corn and/or Sorghum raised for silage.	Silage Rotations MOP.
Small grain silage rotations	Rotations with Wheat, Barley, Rye, Oats, and Millets raised for silage.	Silage Rotations MOP.

Each case entered in the case-base is indexed by declaring it to be an instance of one or more MOPs. If a rotation satisfies the norms of a given MOP then it is indexed as an instance of that particular MOP.

For example, a crop rotation of corn and rye would be indexed under two MOPs: (i) the rotations-with-large grains MOP, and (ii) the rotations-with-small grains MOP because it satisfies the norms of both these MOPs. During the process of indexing, a case is first tested to see if it meets the norms of the top-level MOP, the crop-rotation MOP. Once it satisfies the norms of that MOP, the norms of its specializations are tested. In the example rotation, the norms of the top-level MOP (the rotations-with-grasses MOP) were satisfied, so the norms of the specializations of this MOP (the rotations-with-large grains MOP and the rotations-with-small grains MOP) were then tested.

This process of testing norms continues successively until the case fails to meet the norms of a certain MOP or if there are no further MOPs in that particular branch of the hierarchy. At this point the case is indexed as an instance of the deepest possible MOP in the hierarchy. Because cases inherit all the abstraction links from the MOP they are indexed under, they are essentially instances of their ancestral MOPs as well.

3.2.2.2 Nomenclature for the MOPs

Nodes in the rotation hierarchy are MOPs with a set of norms, specializations, abstractions and scenes (described in chapter two). This hierarchy starts with a root MOP (having no abstractions, only specializations) and is given a unique name. Specializations append their names to the names of their parent MOP. For example, in Figure 3.1, the indices *Grasses.Large grain* refer first to the rotations-with-grasses MOP in rotation hierarchy, and then the first sub-node, the rotations-with-large grains MOP. This index intuitively suggests the 'distance' between MOPs based on the extent of similarity in their names. For instance, *Grasses.Large grain.Corn* (rotations-with-Corn MOP) is closer to *Grasses.Large grain.Sorghum*, (rotations-with-Sorghum MOP) than *Legumes.Soybeans*, which is the rotations-with-Soybeans MOP.

3.2.2.3 Scenes

While a crop rotation is represented as a MOP, its component crops and their tillage and residue management practices (Table 3.4) in a crop rotation are represented as scenes. These scenes are essentially MOPs arranged in a different hierarchy called the scene hierarchy (Figure 3.2). Each MOP in the rotation sequence hierarchy has scene links to various MOPs under this hierarchy (i.e., each crop rotation has scene links to its component crops). The scene hierarchy was developed based on the classification of crops and agronomic practices. The taxonomic and crop descriptor criteria are based on botanical classification and morphological descriptions of crop plants, while the plowing-time and residue management criteria are based on the natural resources conservation service (NRCS) conventions.

Tillage practices are broadly classified as “conservation practices” if the tillage systems maintain at least 30 percent of the crop residues on the soil surface (Brady, 1990). Within the conservation tillage category, “No-till” systems permit direct planting in the residue of the previous crop and use localized tillage necessary to plant the seed, while “minimum-till” covers all other systems that keep at least 30 percent of residues on the surface (Brady, 1990). Other categories were developed based on the common names used for crop plants and the normal practices and conventions used in dealing with crops. Given this set of MOPs, cases can be assigned to MOPs in the rotation hierarchy and linked to their scenes in the scene hierarchy. Indices and scene links are stored along with the cases. For example, the corn and rye rotation mentioned above has an index link pointing to the ‘corn and cover’ MOP in the rotation hierarchy, while it has scene links pointing to the ‘corn’ MOP and the ‘rye’ MOP in the scene hierarchy

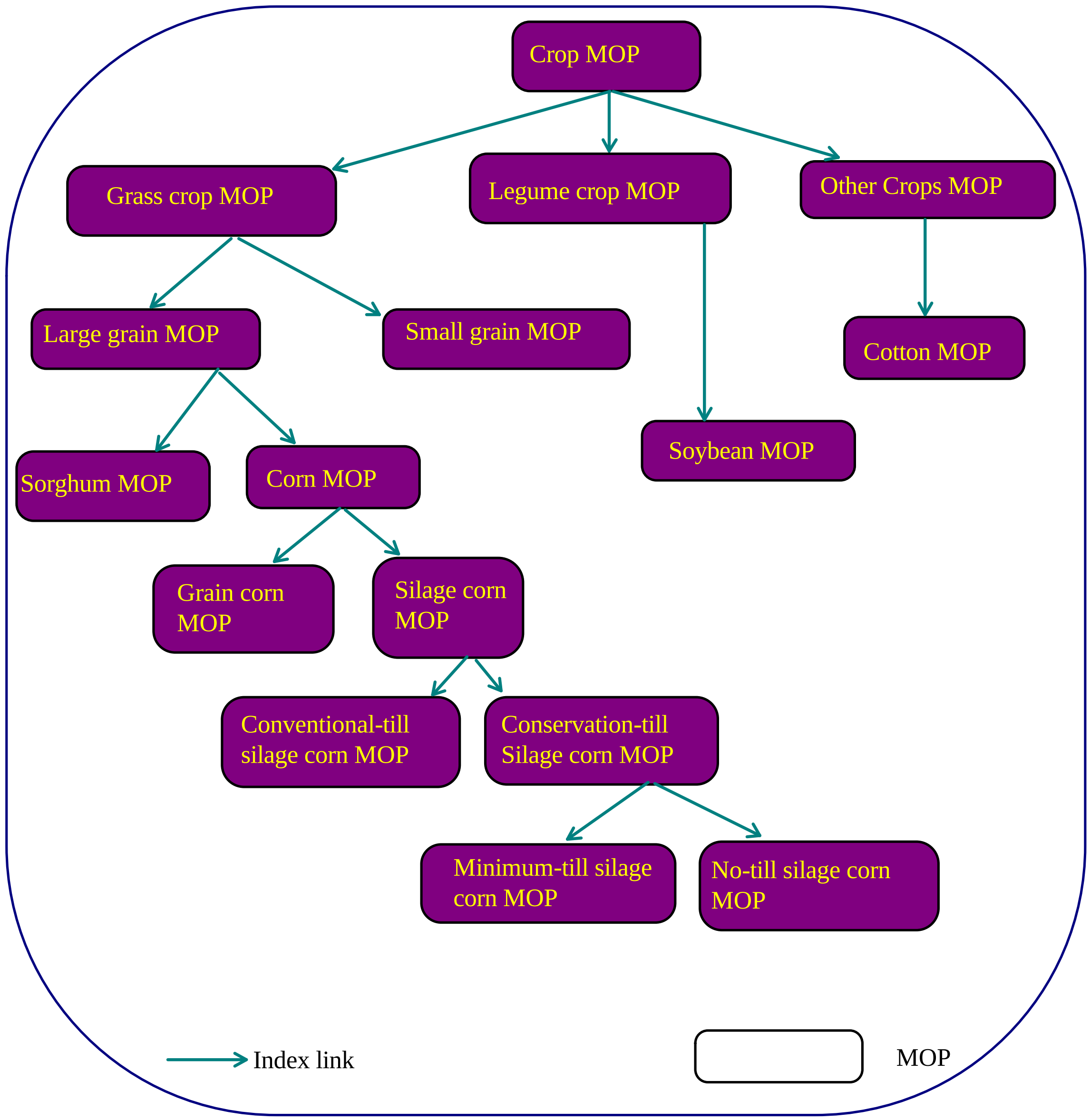


Figure 3.2 **Scene hierarchy.** Schematic representation of scenes as MOPs in a hierarchical fashion. Each of these MOPs is referenced by a scene link from the rotation hierarchy (Figure 3.1).

Table 3.4. Criteria used to represent and classify scenes at successive layers in the crop scene hierarchy.

Criteria	Description	Example
Taxonomic	Crops are classified based on the membership in taxonomic groups.	Grasses, Legumes etc.,
Crop Descriptors	This hierarchy allows for the classification of crops based on their descriptors.	Large Grains, Small Grains, Legumes etc.,
Crop Purposes	Crops are classified based on the purposes of crop included in the rotation.	Silage, Grain, Hay, Beans, Meadow etc.,
Tillage	Crops are classified based on their tillage practices.	Conventional tillage, no-till etc.,
Plow-Time	This hierarchy allows for the classification of crops by the number and season of plowing operations in the rotation.	Spring plow, Fall Plow etc.,
Residue Management	Crops are classified based on their residue management practices.	Residue left, Residue removed.

Enabling a CBR system to reason about crop sequences was a significant challenge. Two different cases could share the same set of indices and scenes if they had the same crops and practices but a different crop sequence. For example, the indices and scenes for the rotation, corn-rye-soybeans-wheat-millet-rye, would be the same as for the rotation, millet-wheat-soybeans-rye-corn-rye (assuming the management practices for the individual crops remain the same). The two rotations, however, have different *C*-values and insect pest risks due to differences in their sequences. Therefore, the arrangement of individual scenes in a rotation MOP is an important factor in determining their similarity.

Kolodner (1993) describes MOPs as entities that package scenes, by referring to their sequence as part of the MOP's descriptive information. Therefore, the sequence problem can be overcome by making the sequence a norm of the MOP. For example a "small grain-legume" MOP has as its norm a small grain crop followed by a legume crop. If the rotation is a legume followed by a small grain crop, then it would not be indexed as an instance under this "small grain-legume" MOP. This method of arranging scenes is enforced by the MOPs in CBR systems like JUDGE (Bain, 1986; Riesbeck and Schank, 1989) and CYRUS (Kolodner, 1984).

However, crop rotations are cyclical and often open ended, (i.e., the number of crops in a rotation is not fixed). So the enforcement of a set sequence as a norm is not an efficient method of representation. I propose an alternative *ad hoc* method for handling this anomaly in the following sections.

3.3 Reminding

Reminding occurs when a new case is posed to the system and cases deemed similar to the input case are identified or 'reminded'. Reminding in a MOP-based system is an operation on the MOP hierarchy under which the new case is indexed and close matches are returned (Figure 3.3). Two cases are identical if all their indices and scenes match, (i.e., their component crops, their tillage and residue management practices must all be the same). Because the reminding process in a MOPs system is based on indices assigned to the cases, the first issue in the development of a reminding scheme for this CBR design is the problem of indexing the input case.

3.3.1 Indexing the Input-Case

Since this case-based reasoners developed in this project (C-Chest and Pandora) are intended to evaluate the crop rotation alternatives generated by CROPS, I assume that

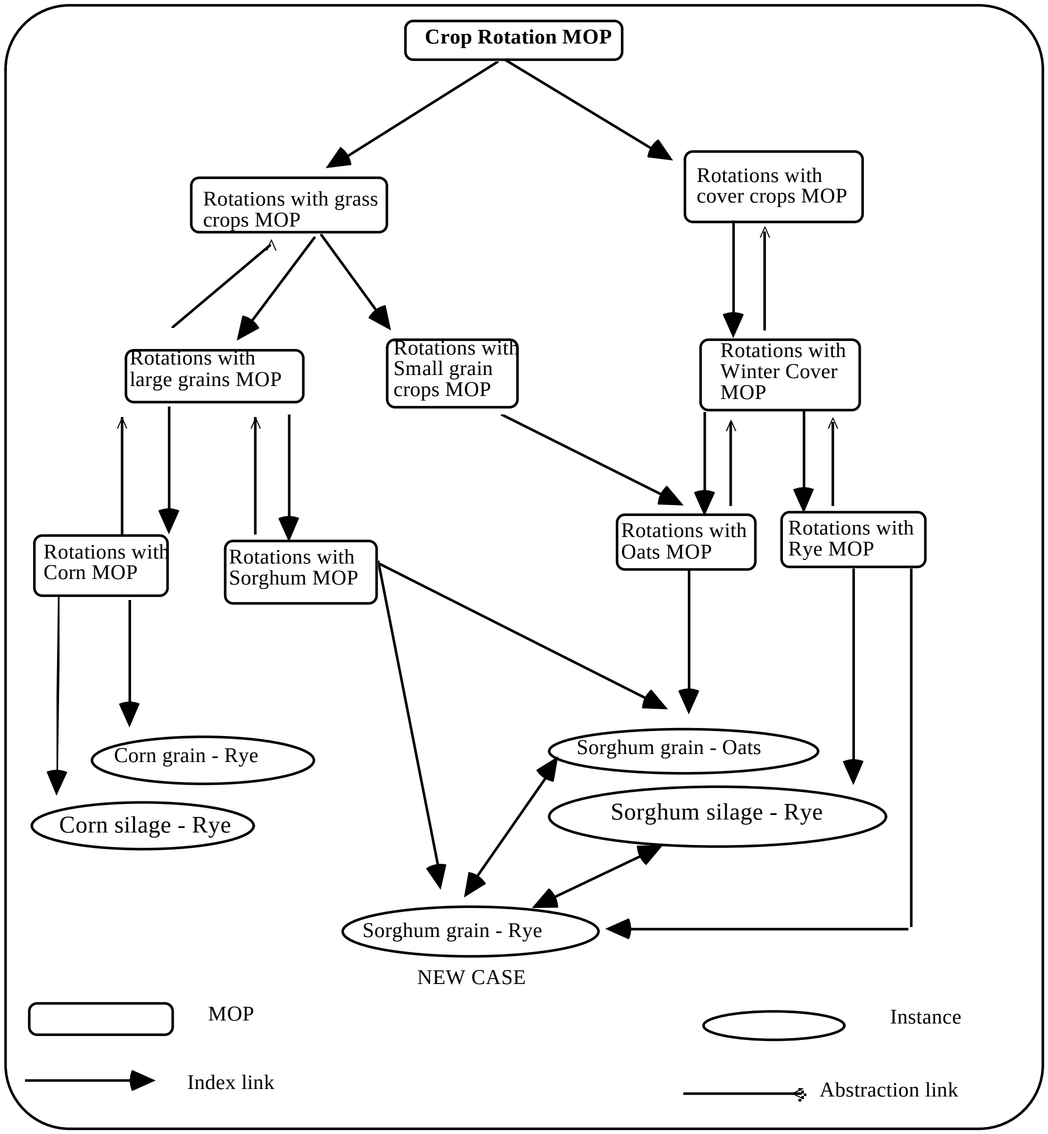


Figure 3.3. Schematic representation of reminding in a MOP hierarchy. The new case is indexed under two MOPs in this diagram and the instances under those MOPs, indicated by double-headed arrows, are the reminded cases.

each new case upon entry will have a complete description of its key features. These include: cropping sequence, tillage practices, residue management practices, purposes of the crops, and duration of the rotation. From these features the values of indices must be derived, including: crop descriptors, crop families, crop seasons, tillage descriptors, and meadow type. This process of deriving indices can be accomplished by using rules as in the JUDGE system (Bain, 1986; Reisbeck and Schank, 1989).

In the rule-based scheme of generating indices, rules are simple IF...THEN... statements that take individual crops from the cropping sequence and assign a corresponding value to the crop families, seasons, etc. For example, for an input case of a corn silage and rye rotation (Table 3.5), the generated indices include values for the Crop Descriptor, Tillage Descriptor, and Crop Purpose attributes of the input rotation. Based on these features the crop rotation is then indexed as an instance under one or more MOPs in the rotation hierarchy with scene links to its component crops and practices.

3.3.1.1 Working example

Consider an example input rotation of a 3-year crop rotation, Graincorn-winterwheat-soybeans-rye. Rye is used as the winter cover crop in the third year. Corn is raised for grain and planted no-till. Wheat is grown under the conventional tillage practices with a fall plowing after corn is harvested. Corn, soybeans and rye are raised using "no-till" conservation tillage. The rotation sequence and the corresponding tillage and residue management practices are summarized below.

Rotation sequence	: CornG-Rye-CornG-WheatG-SoybeansDC-Rye.
Corresponding Tillage practices	: Notill-Notill-Notill-FallConventional-Notill-Notill.
Crop residue management	: Left, Left, Left, Removed, Left, Left.

Index links are characterized by an attribute-value pair. i.e., for the attribute Crop, for example, for the attribute Crop, if the value is Large grain, then the index link points to a Large grain MOP (in this case the rotations-with-large grains MOP). To determine the index links for this input rotation, a separate rule-based inference is employed which determines the values for the various indexing attributes in the hierarchy. Indexing starts from the root MOP and progresses successively downward in a hierarchy. The indices for the example input rotation are given in Table 3.6.

From these indices the MOPs that are deepest in each of the branches from the Root MOP are identified. These are rotations with: DC Soybeans, Wheat, Corn, Rye, No-till, Conventional-Till. Once these MOPs are identified the input case is then compared to their instances to determine similarity. The first step in this process is to

Table 3.5. Rule-based determination of indices for the input rotation.

Antecedent	Conclusion	Indexed MOP
Input	Sequence of crops	Crop Rotation MOP (Root)
Cover crop yes/No?	Rotation has a cover crop.	Rotations with CoverMOP
Grass crops yes/no?	Rotation has grass crops in it.	Rotations with GrassesMOP
Legume crops yes/no?	Rotation includes a legume crop.	Legume rotations MOP.
Rotation has conservation tillage?	Five crops are given conservation till	Cons. Till rotations MOP
Rotation has conventional tillage?	Rotation has wheat raised under Conv. till.	Conv-Till rotations MOP
Rotation has a meadow?	Rotation does not have a meadow.	None
Is the cover in Winter?	Rotation has a winter cover	Rotations with winter cover MOP.
Is the cover crop Rye?	Rotation has Rye as winter cover.	Rotations with Rye MOP
Does the rot have a Large grain?	Rotation includes a large Grain	Rotations with Lg.Grain MOP
Is the Large grain Corn?	Rotation has Corn	Rotations with Corn MOP
Does the rot have a Small grain?	Rotation has two small grain crops.	Rotations with SmGr. MOP
Is the Small grain Wheat?	Rotation has wheat	Rotations with Wheat MOP.
Does the rotation have Soybeans?	Rotation has soybeans	Rotation with Soybeans MOP
Are the beans raised Double-crop?	Rotation has a double-crop soybeans situation.	Rotation with Double-Crop Beans MOP.

find a rotation that is common to all these MOPs. If there is no such case then the case with the most scenes in common will be selected for similarity assessment.

3.3.2 Similarity Assessment

On indexing the input case, the CBR has to assess the similarity between the input case and the instances of the various MOPs the input case points to. Instances of MOPs at the deepest node in the hierarchy are used in this process.

Similarity assessment can be qualitative, where the CBR retrieves one or more instances of the deepest MOP the case points to based on a qualitative criterion; or can be numeric, where the CBR assigns a similarity score to each of the instances under the relevant MOPs. The need for a similarity score arises when a numerical measure of similarity is required in a CBR.

For instance, in C-Chest, a weighted average of the retrieved cases' *C*-value was desired to generate the *C*-value of the input case. Therefore, a numerical similarity measure is required in C-Chest. In Pandora, however, the retrieved cases need not be assigned a similarity score as it is of no relevance in the adaptation process. As such, similarity assessment is limited to evaluating the extent homogeneity in the scenes and facilitating retrieval of the reminded cases. No numeric score is assigned in this case. For illustration purposes I present a simple example of similarity assessment in the following sections.

3.3.2.1 Working example

Continuing the example presented above, I have now determined the indices for the input case and have identified the instances under each of the deepest MOPs they point to. Let us assume that one of the instances is a 3-year rotation with soybeans, sorghum and corn in the three summers with wheat preceding corn. Rye is the winter cover and tillage practices are no-till for all the crops except Wheat, which is planted conventional till.

The features of this rotation are summarized as follows:

Rotation sequence	SoybeansDC-Rye-SorghumS-Rye-CornG-Wheat.
Corresponding Tillage practices	Notill-Notill-Notill-Notill-Notill-FallConventional.
Crop residue management	Left-Left-Removed-Left-Left-Left.

To compare this rotation (stored rotation) with the input rotation and determine their similarity, the CBR has to evaluate the extent of homogeneity in their component scenes. The scenes in the input and the stored rotation are given in Table 3.6. Similarity is then determined between the scenes in these two crop rotations on a scene by scene basis. However, the comparison is not strictly between Scene1 of the input case and Scene1 of the stored case. It can be between any two scenes in the two rotations as long as the order is not broken. Scene1 of the input rotation can be compared to any of the six scenes in the stored rotation as long as scene two of the input rotations is compared to the immediate succeeding scene in the stored rotation). To illustrate this three possible comparisons are given in the sets below (**bold** letters indicate the scenes in the input rotation) :

Input case rotation sequence : **CornG-Rye-CornG-WheatG-SoybeansDC-Rye.**

Stored case rotation sequence : SoybeansDC-Rye-SorghumS-Rye-CornG-Wheat.

(**Corn**, Soybeans), (**Rye**, Rye), (**Corn**, Sorghum), (**Wheat**, Rye), (**Soybeans**, Corn), (**Rye**, Wheat)

=> Comparison one.

(**Corn**, Sorghum), (**Rye**, Rye), (**Corn**, Corn), (**Wheat**, Wheat), (**Soybeans**, Soybeans), (**Rye**, Rye)

=> Comparison two.

(**Corn**, Wheat), (**Rye**, Soybeans), (**Corn**, Rye), (**Wheat**, Sorghum), (**Soybeans**, Rye), (**Rye**, Corn)

=> Comparison three.

Table 3.6. Identification of the component scenes in the input and the stored cases.

Scenes in the Input Rotation	Scenes in the Stored Rotation
No-till grain corn	No-till double crop beans
No-till rye	No-till rye
No-till grain corn (repeats)	No-till silage sorghum
Conventional till wheat grain	No-till rye
No-till double crop beans	No-till grain corn
No-till rye	Conventional till wheat G

It is obvious that these two rotations have a greater similarity in comparison two than the rest. The rationale behind such comparisons is that a rotation is deemed cyclical and as such there is no ‘first’ and ‘last’ scene in a rotation. The only arrangement of consequence is the logical sequence of crops that occur in the rotation. These comparison procedures are particularly important to determine the similarity between two rotations that were essentially the same but presented in two different sequences.

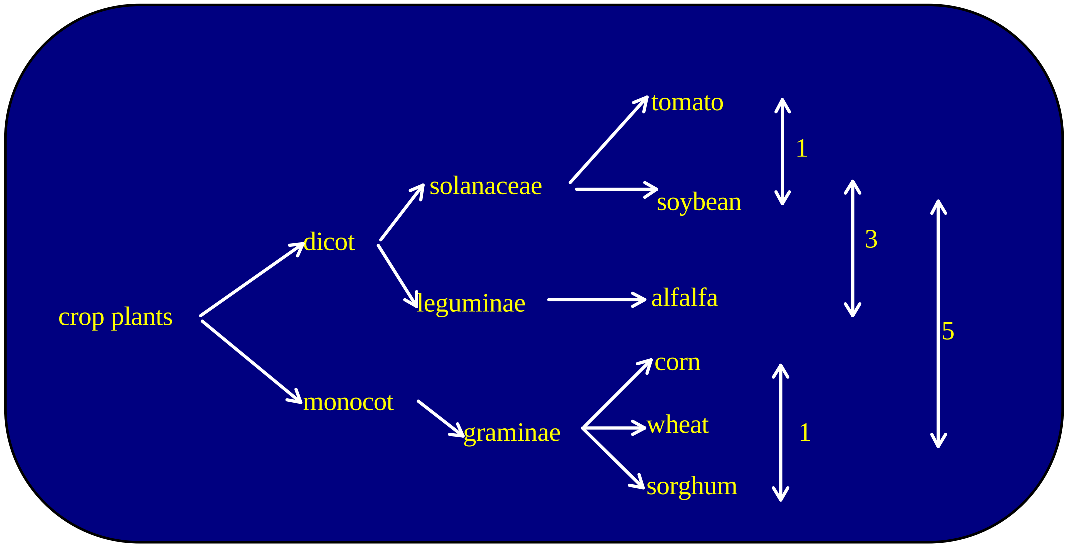
For example consider two two-year rotations, A and B, involving corn, rye, and wheat. Let the sequence for rotation A be corn, wheat, corn, and rye; while the sequence for rotation B is corn, rye, corn, and wheat. Assume that the management practices are the same. These rotations then have the same scenes but not in the same order. As such, when comparing these two rotations, it is important to recognize the anomaly in their sequences and assess similarity accordingly. There are several ways to assess similarity between cases (and their component scenes). I present examples of two methods based on numerical taxonomy in this discussion.

Numerical taxonomy deals with the grouping by numerical methods of observational taxonomic units (OTUs) into taxa on the basis of their character states (Sneath and Sokal, 1973). Numerical methods also help to determine similarity (or dissimilarity) between OTUs based on their character states. In this design each MOP was treated as an OTU and its norms formed the character states for the OTUs.

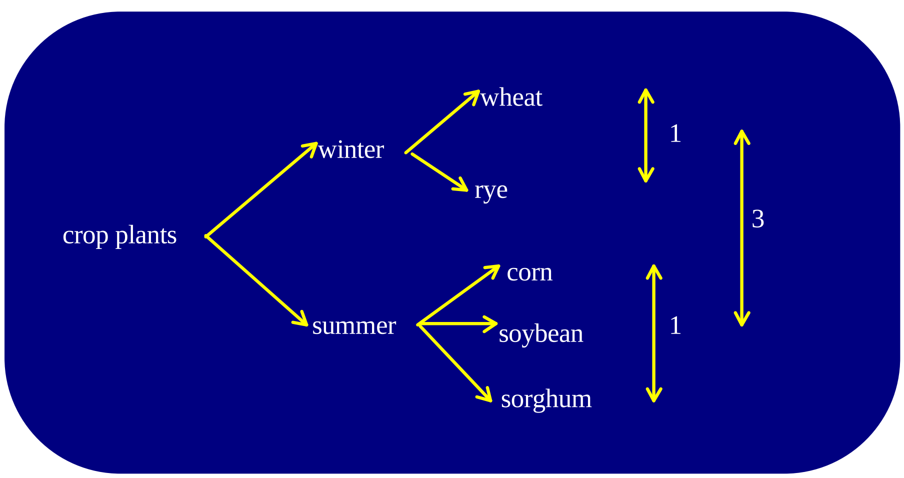
For assessing similarity between MOPs at various hierarchies, the hierarchies are first represented as dendrograms. Dendrograms are a branching network of relationships devised using morphological or evolutionary criteria. Every node leading to one or more branches in a dendrogram is called a furcation. When dendrograms are developed using morphological or phenetic criteria they are referred to as Phenograms, while dendrograms based on evolutionary or ancestor-descendant relationships are called Cladograms (Sneath and Sokal, 1973). The cladistic distance method (Farris, 1969) is a convenient method to estimate the distances between OTUs in a cladogram. This plan involved counting the number of furcations one would have to go through from one OTU to another in a cladogram (Sneath and Sokal, 1973). I used this method to determine similarities between MOPs in a hierarchy based on various phenetic and cladistic relationships[2]. For example, in Figure 3. 3a, corn is more closely related to wheat than to soybean or alfalfa. This is based on the taxonomic relationship between corn, wheat and soybeans. Figure 3.3b is based on "crop growth season", a phenetic criteria. In this dendrogram soybean is more closely related to corn than wheat. Thus, distances depend

[2]Since the hierarchies were based on a combination of both evolutionary and phenetic criteria, I generically refer to them as dendrograms.

a). Dendrogram based on taxonomic relationships.



b). Dendrogram based on seasonal relationships



Figures 3.4 a, b. Dendrograms of MOPs. Numbers by the arrows indicate cladistic distances.

upon the classification criteria used in developing the dendrogram. This system of determining distances was simple and fairly straightforward. However, this plan assumes an objective dendrogram of OTUs and develops distances based on the nodes and divisions. The hierarchies presented in this design were not developed using such objective numerical techniques.

The second method tested was the measurement of similarity using a simple matching coefficient introduced by Sokal and Michener, 1958. It is defined as: “the ratio of the total number of matches to the total number of characters” (Dunn and Everitt, 1982). This method is based on the similarity in the number of character states. While comparing any two OTUs, *i* and *j*, data for all of the character states recorded (assuming that they are all binary characters) can be summarized in a 2 x 2 table given in Table 3.7 (Dunn and Everitt, 1982). Character states are coded as one (1) for the presence, and zero (0) for the absence of a given character (Table 3.6). Mathematically, the simple matching coefficient (s_{ij}) is then given by:

$$s_{ij} = (a + d) / p$$

where, p is the total number of binary characters studied, a is the number of characters where both OTUs are coded 1, and d is the number of characters where both the OTU’s are coded 0.

The character states and values matrix were generated using hypothetical data presented in Table 3.8. Similarity values are given in Tables 3.9 and 3.10. For this example, using values from these tables, I could derive the similarity between a grain MOP and silage MOP to be about 67 percent (3/5), and the similarity between a grain MOP and beans MOP to be 80 percent (4/5). Using this method similarity scores can be calculated for all the instances of the MOPs the input and stored cases refer to. Ultimately, while comparing two cases, the CBR can use these distance metrics to arrive at an overall similarity between the scenes in each of the two cases. Thus, numerical methods provide convenient objective means to estimate similarities between crops in a rotation represented in a MOPs based hierarchy.

One of the advantages of using a numerical measure of similarity is that the developer can control the retrieval process using a numerical threshold (i.e., a threshold for reminding could be set to retrieve only desirable matches). Most of the commercially available CBR development environments have this threshold function as a standard feature. For example, if a threshold of 70 percent is set, then the system will retrieve only cases that match the input case in more than 70 percent of its features. This allows for efficient retrieval as the system can stop similarity assessment for cases that are not close matches to the input case and move on to the next case.

Table 3.7. A 2 x 2 matrix of data collected on character values (redrawn form Dunn and Everitt, 1982).

		OTU *i*		
		1	0	
OTU *j*	1	a	b	a + b
	0	c	d	c + d
		a + c	b + d	p = a + b + c + d

Table 3.8. Character scoring and values for the crop purpose hierarchy. Numbers in parenthesis indicate values assigned to the corresponding character state.

OTUs	**Primary plant part of interest**	**Usual re-growth**	**Residue**	**Duration**	**Typical crops**
Grain	Seed (0)	No (0)	Left (0)	Annual (0)	Grasses (0)
Silage	Vegetative(1)	No (0)	Removed (1)	Annual (0)	Grasses (0)
Beans	Seed (0)	No (0)	Left (0)	Annual (0)	Legumes (1)
Hay	Vegetative(1)	Yes (1)	Removed (1)	Perennial (1)	Legumes (1)
Meadow	Vegetative(1)	Yes (1)	Left (0)	Perennial (1)	Grasses (0)

Table 3.9. Data for the character states recorded between OTUs grain and beans.

		OTU *grain*		
		1	0	
OTU	1	0	1	1
beans				
	0	0	4	4
		0	5	5

Table 3.10. Data for the character states recorded between OTUs grain and silage.

		OTU *grain*		
		1	0	
OTU	1	0	2	2 (a+b)
silage				
	0	0	3	3 (c+d)
		0 (a+c)	5 (b+d)	5 (a + b + c + d)

3.4 Adaptation

Once similarity is assessed between the input case and the cases in the case-library, the cases that matched above the threshold are selected as the 'reminded' cases. These cases are stored in a list, ordered by their degree of match with the input case. These retrieved cases form the basis for adaptation. The adaptation design proposed here is an *ad hoc* scheme. For convenience, it is described in two parts, one for each specific objective. The first one is named Pandora and deals with the problem of generating insect pest risks in crop rotations. The other one, C-Chest, generates *C*-values for crop rotations. As the scope of this chapter is the development of a design for the proposed CBR system, actual implementation details are not elaborated here. Chapter four deals with these issues in greater detail.

C-Chest and Pandora differ fundamentally at the adaptation stage because of the nature of their outputs. In C-Chest, adaptation is required to generate a quantitative value, so a quantitative method is suggested for adaptation. In Pandora, the desired output is qualitative, and therefore a different adaptation procedure is proposed. Both systems, however, make use of the retrieved cases.

3.4.1 Adaptation in Pandora

A modification of the derivational replay scheme discussed in chapter two (section 2.3.6) is proposed for adaptation in Pandora. In this scheme, each of the retrieved cases provides information on how the pest risks were generated in that case. Each of the component crops in the rotation has a scene link to its major pests. Each pest scene, in turn, exists in three levels: low, medium and high risk. For instance, if the retrieved case is a continuous corn rotation and the rotation points to a high risk from rootworms scene, the CBR checks the input rotation for an index link to continuous corn in its crop sequence. If there exists any such index then the system automatically suggests that rootworms would be a high risk in the input case. Likewise, scenes from several retrieved cases are used for generating a list of insect pest risks.

In the event of conflicting operators (i.e., if one retrieved case suggests a high risk for a given pest while another suggest medium risk) the similarity assessed in the reminding process is used as an evidence measure for prioritizing the use of the retrieved operators. Thus, simple rule-like operators derived from each of the retrieved cases tailor an insect pest list along with their predicted risks for the input case.

3.4.2 Adaptation in C-Chest

Assigning similarity scores to the retrieved cases plays a prominent role in the adaptation of C-Chest. The ranking in the list of retrieved cases is converted into a percentage based on the overall similarity assigned to each of the retrieved cases. Each of the retrieved cases has a *C*-value as a feature. For generating the *C*-value of the input case, a weighted average of the retrieved cases' *C* -values is taken. This is an *ad hoc* scheme and as such is subject to refinement based on performance.

3.5 Learning

Recall from my discussion in chapter two (section 2.4) that CBR allows for automated learning. New cases posed to the system are retained by adding them to the case library. Further, learning also occurs by the incremental extensions to the MOP hierarchy where new MOPs and indices are created as a result of case accumulation. In this design I propose a combination of case retention and creation of new generalizations to facilitate learning. The following section provides an overview of the learning process. Actual details of implementing a learning scheme are beyond the scope of this chapter.

Case retention is facilitated by storing the solved case as an instance in the case library. This case would have the appropriate indices determined during the reminding process. However, not every case needs to be stored. Only those cases that are significantly different are desirable for retention. This significant difference is determined using a combination of quantitative and qualitative methods. Numerically, a threshold is set for the similarity score returned by the similarity assessment scheme to determine significant difference. For example, in C-Chest a threshold of 75 percent is set for case retention and any new case that does not have a match of greater than 75 percent is retained by the system. Case retention can also be guided by using heuristcs derived from interacting with experts. Creation of new MOPs follows the case retention process. Simple rule like operators are used for guiding the CBR in forming new MOPs based on similarities among instances stored under any given MOP.

3.6 Summary for Chapter 3.0

In CBR, the system cycle primarily consists of a knowledge repository or a case-base, an indexing and reminding scheme or retrieval process, and an adaptation procedure (Dutta and Bonnissone, 1992). The case-base in this system consists of cases encoding information pertaining to individual crop rotations along with their *C*-values and information on pest risks in that rotation. Case representation is in memory organization packet (MOP)s-based hierarchies and networks, introduced in chapter two. Cases are indexed by several derived features and reminding takes place based on similarities among these derived features. Adaptation is carried out using a simple weighted average scheme for generating *C*-values, and a derivational scheme using rules for generating the insect pest risks.

Chapter 4.0

System Implementation

This chapter provides details on the implementation process. It describes how the design presented in chapter three was modified and implemented using a commercial CBR software package, and reports results from the systems developed.

4.1 Software Selection

Based on the primary functions of a CBR described in literature (Riesbeck and Schank, 1989; Kolodner, 1994), I determined the basic features required of a CBR shell to be: (i) availability of a case-base creation and storing mechanism; (ii) indexing, similarity assessment, and case retrieval algorithms; and (iii) availability of adaptation measures. With these as the minimum standards for software performance, I sought product information from several vendors to see if they meet the requirements in this project.

Those CBR development environments available and with at least the minimal set of features were *CBR Express* from Inference Corporation, *ReMind* from Cognitive Solutions, and *Esteem*™ from Esteem software incorporated. CBR Express and ReMind were more comprehensive than Esteem™, but each of them had the basic elements required for the development of a CBR application.

Since both CBR Express and ReMind were priced significantly higher than Esteem™, I selected Esteem™ as the development environment for this project. Esteem™ provides tools, in the form of editors, for representing cases, defining indexing and similarity schemes, and building rules for similarity assessment and case adaptation. A summary of Esteem's features, in comparison with ReMind and CBRExpress, is given in Table 4.1.

All these three CBR shells have, as standard features, some automated weight generation schemes for similarity assessment. These schemes have their origin in the machine learning paradigm of artificial intelligence. This approach to reminding differs significantly from the classical notion of CBR as proposed by Schank et al., (1989) in

Table 4.1. Comparison of some CBR software

FEATURE	CBR Express[3]	ReMind[4]	Esteem™[5]
Platforms and Operating Systems	Built on ART-IM, runs on mainframes, IBM-PCs with Windows 3.1.	Written in C++, runs on Macintosh, Windows 3.0 or OS/2-PM.	Built on Kappa-PC (from Intellicorp), runs on Windows 3.1.
Indexing Techniques	Case-representation and indexing is in Frames and Schemas.	Inductive indexing and nearest neighbor indexing.	Nearest neighbor matching; Nested Case and Case-Base indices.
Reminding Schemes	Feature counting with weights by default, induction or user-definition.	Inductive retrieval, Nearest neighbor retrieval, and template matching are allowed.	Feature counting, Weighted feature (with Inductive methods), and Inferred computation.
Case Adaptation	Allows for refinement of the input case based on the retrieved case(s).	Uses user-defined formulas to adjust output.	Mostly by rules, Kappa functions and external function calls.
User Interface	Interactive, graphical user interface with natural language handling capabilities.	A fairly comprehensive interface that can be customized for specific applications.	Functional with limited functions. Allows customizing based on features.
Remarks	A fairly comprehensive high-end development tool, offers client/server DBMS and multimedia support.	Allows representation of case relationships in a hierarchy; Case generalization, and inductive explanation.	Allows storage and mapping of cases from dBase, Lotus or ASCII formats.
Price (5/94)	>$5000.00	>$3,500.00	$395.00

[3] Based on CBR Express documentation, Inference Corp.,1994, and Adam, Johan D (1991).
[4] Based on ReMind documentation provided by Cognitive Software Inc., 1994.
[5] Based on Esteem documentation provided by Esteem software inc., 1994.

which the system's reminding is guided by the domain knowledge as opposed to machine generated weight vectors. In this context, CBR applications developed using weight generation schemes are not 'true' CBR systems. However, most of these shells also allow the development and use of domain specific rules that can be used during the process of similarity assessment. Thus, most applications developed using a CBR shell are a combination of machine learning and CBR paradigms. This project also employs the combinatorial approach.

4.2 Objectives for the CBR

I had two specific objectives for this project. One was to develop and test a reminding scheme, while the other was to generate a list of insect pest risks in a crop rotation. In accordance with these objectives, the CBR applications developed in this project were intended to:

1. Develop and test a reminding scheme, *C-Chest*, that can provide C-values for new and unique crop rotations, and
2. Develop a CBR application, *Pandora*, to predict insect pest risks in a crop rotation.

4.3 Case Sources

The case library comprised a set of 207 crop rotations involving corn, soybeans, tobacco, and small grains in southwestern Virginia under different management regimes. Data on *C*-values from the tables published by the soil conservation service (SCS, now called the Natural Resources Conservation Service - NRCS) during 1988-90 for mountains and valleys region of Virginia (Area 4) were compiled into 140 cases. In addition, 65 cases of multi-crop rotations from the CROPS system (Buick et al, 1992) were included in the case base. These rotations were generated using the crop succession network presented in Figure 4.1. Additional information on crop rotations practiced in Virginia came from Dr. James McKenna, Associate Professor, Crop and Soil Environmental Sciences Department, Virginia Tech. The *C*-values for these rotations were derived from the three crop network model used to determine the *C*-value in the CROPS system (Stone et al, 1992). Crop names, tillage, and residue management practices were classified according to the NRCS conventions reflected in the data tables obtained from their Montgomery county field office.

Information on pest risks in crop rotations was obtained from interviews with IPM experts and from literature (VCE Publication 456-016, 1993). Three formal interview sessions were conducted with Dr. R. R. Youngman, associate professor and extension entomologist, Department of Entomology, Virginia Tech, during 1994-95.

Additional information on relationships between insect pests and hosts came from discussions with Dr. Loke T. Kok, professor, Department of Entomology, Virginia Tech and from literature (VCE Publication 456-016, 1993). A brief description of the cases is given in Table 4.2.

4.4 Case Representation

Case representation in Esteem™ starts with the 'case-base definition' (Fig. 4.2). Esteem™ allows six types of features: Boolean (true or false), Text, Numeric, One-of-a-list, Case, and Multimedia. Text type features can be strings of characters or can be lists in which the individual elements are separated by a comma or a space. Numeric types can be integer or real number values; they have a maximum and a minimum value. In this application, each case describes a crop rotation by providing values for the various features. Features involving sequences like crop rotation, tillage, and residue were represented as text type features while features like rotation length and *C*-value were represented as numeric features. Feature names and types used for describing crop rotations as cases in Esteem™ are given in Table 4.3. Cases were developed using the case-base editor (Fig. 4. 3) provided by Esteem™. Cases were assigned unique names.

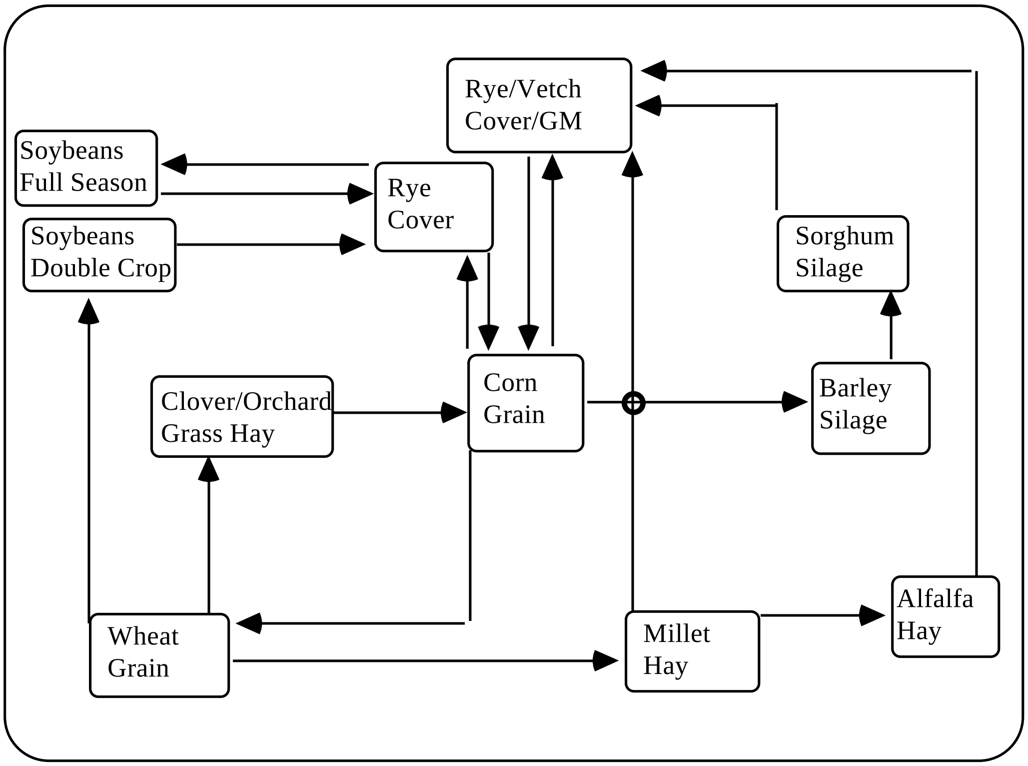


Figure 4.1. Succession of crops in a rotation (Redrawn from Buick et al., 1992).

Table 4.2. Description of cases in the case-base.

Rotation Type	Rotation Descriptions	Number
Tobacco Rotations	Rotations of tobacco, tobacco with cover, and tobacco with meadow.	20
Corn Rotations I	Corn grain rotations including corn followed by fallow, and corn followed by cover rotations.	40
Corn Rotations II	Corn silage rotations including corn silage followed by fallow, corn silage followed by cover and corn silage followed by meadow rotations.	40
Mixed Crop Rotations	Rotation including corn, wheat and soybean double crop rotations.	90
Soybean rotations	Rotations of soybeans with a cover crop (full season soybeans).	10

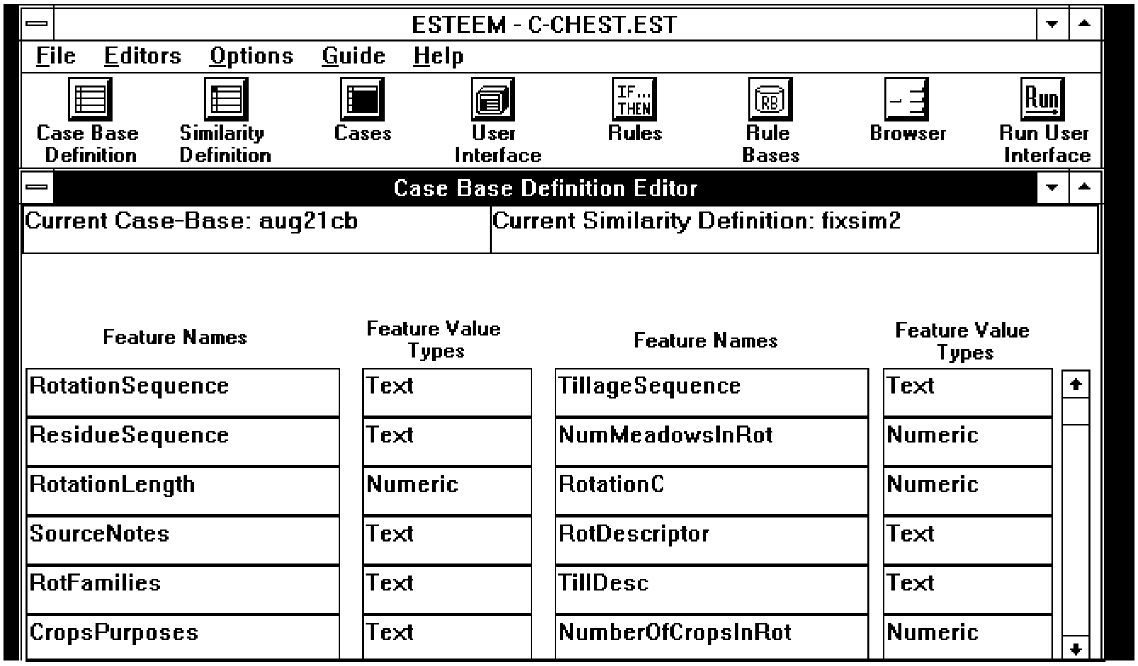


Figure 4.2 Screen capture from Esteem showing the case-base definition editor. The features used for representing a case and the type of their values are shown in this picture.

Table 4.3. Feature names and types in the case base. Numeric values are real numbers.

Feature Name	**Feature Type [Maximum, Minimum]**	**Feature Name**	**Feature Type [Maximum, Minimum]**
Rotation Sequence	Text	Meadow Duration	Numeric (0, 10)
Rotation Length	Numeric (0, 20)	Number Of Meadows	Numeric (0, 2)
Number Of Crops	Numeric (0, 20)	Meadow Notes	One-of-a-list
Tillage Sequence	Text	Field Organic Matter	One-of-a-list
Residue Management	Text	Field Soil Type	One-of-a-list
Crop Purpose	Text	Insect Pest History	Text
Crop Descriptor	Text	Rotation Notes	Text
Crop Families	Text	*C*-Value Source	Text
Tillage Descriptor	Text	Meadow Type	One-of-a-list

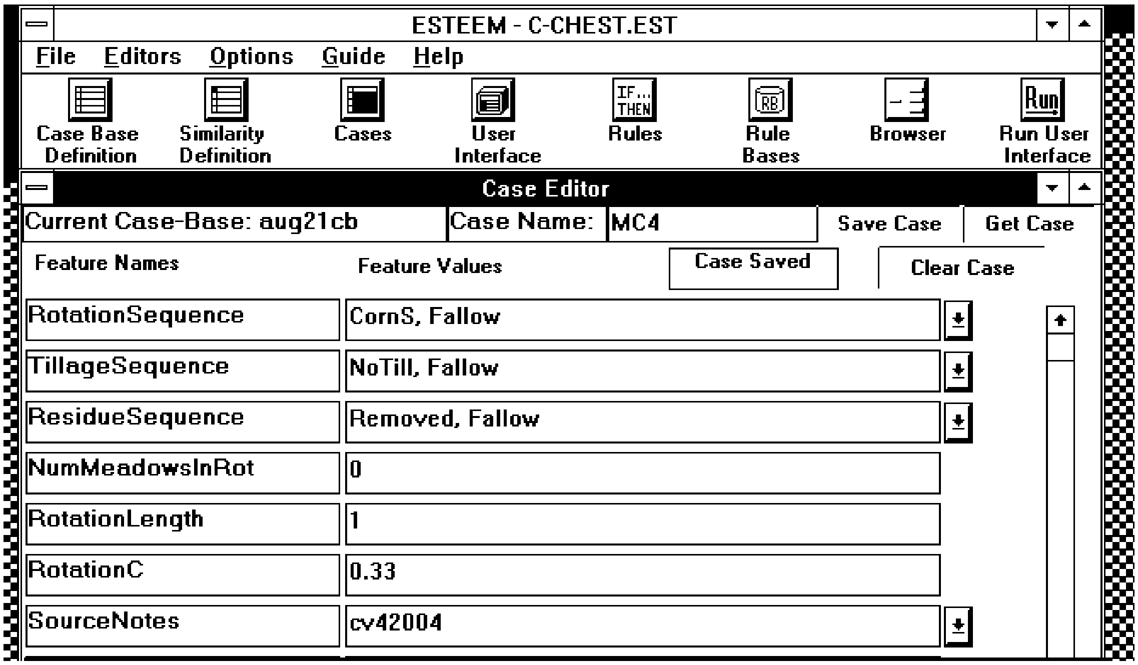


Figure 4.3 Screen capture from Esteem showing the case editor.

4.5 Indexing

Indexing a case-base allows the CBR application to organize cases based on one or more features as defined in the Indexing editor. In Esteem™, cases can be organized across a single feature, or across multiple features. However, only those features that are used for indexing are counted during similarity assessment in Esteem™. Therefore, all the features selected for determining similarity have to be used in building an index. On indexing, Esteem™ organizes cases with similar indices into groups by assigning them with an unique numeric notation. During reminding, the CBR system looks for cases with the same, or similar indices and computes their similarity for retrieval.

4.5.1 Design modifications in the Case-Base

To develop the system in Esteem™, several modifications and compromises were made to the design. The case-base description calls for a hierarchical representation of features and cases. Such a representation was not feasible in Esteem™. Nested case-bases are allowed in Esteem™, permitting a fair degree of hierarchical representation. However, the use of nested cases reduced the efficiency of the retrieval and similarity algorithms considerably. Therefore, a flat case structure was used in the system development. To include hierarchical relationships in a flat case representation, I used the concept of ‘derived’ features. These features were generated using indexing rules that operate on the input values.

For example, if the input rotation is a corn and rye rotation in which corn is raised for grain and rye is the winter cover, the only obvious user input is the rotation sequence. This forms a ‘surface’ feature. Derived features are abstract values that are inferred from the surface features. In the example presented above, based on the rotation sequence, the indexing rules generate the values for the “Crop Descriptor” feature as a rotation involving “a large grain and a small grain”. Likewise, the “Crop Purpose” feature has the value of “a grain and winter cover” rotation. Such derived features implicitly enforce a hierarchy of relationships between cases and therefore formed the indices for this case. This technique allowed the representation and reasoning of crop rotations at a more general level than the crop names feature.

4.6 Reminding and Retrieval

Reminding in Esteem™ is primarily numerical and is carried out in two steps. First, the system assesses the similarity between the input case and the stored cases and second, retrieves the set of cases that are above a certain similarity threshold set by the developer. For instance, reminding can be limited to cases that have a similarity score of 50 or greater (i.e., those cases that are at least 50 percent similar to the input case or better). This is termed as the 'Similarity Threshold' in Esteem™ and is defined by the developer in the Similarity definition editor (Fig. 4.5).

4.6.1 Similarity Assessment

Similarity assessment consists of a feature matching algorithm in which individual features of the input case are iteratively compared with the same features in the stored cases and a numerical similarity score (expressed in terms of a percentage) is computed. The names of the features to be compared and the weights and type of matching desired are described in the similarity definition editor (Fig. 4.4). After defining the features, their weights and the type of matches and similarity, Esteem™ uses the following formula (Eq. 4.1) for computing the similarity between cases.

$$\frac{\sum_{i=1}^{n} W_i \text{ X } sim\left(f_i^1, f_i^R\right)}{\sum_{i=1}^{n} W_i} \qquad 4.1$$

where: Wi is the importance or weight of the feature i; sim is the similarity function for its values; and f_i^1 and f_i^R are the values for feature f_i in the input and retrieved cases respectively (similarity score computation in ReMind, based on Kolodner, 1994).

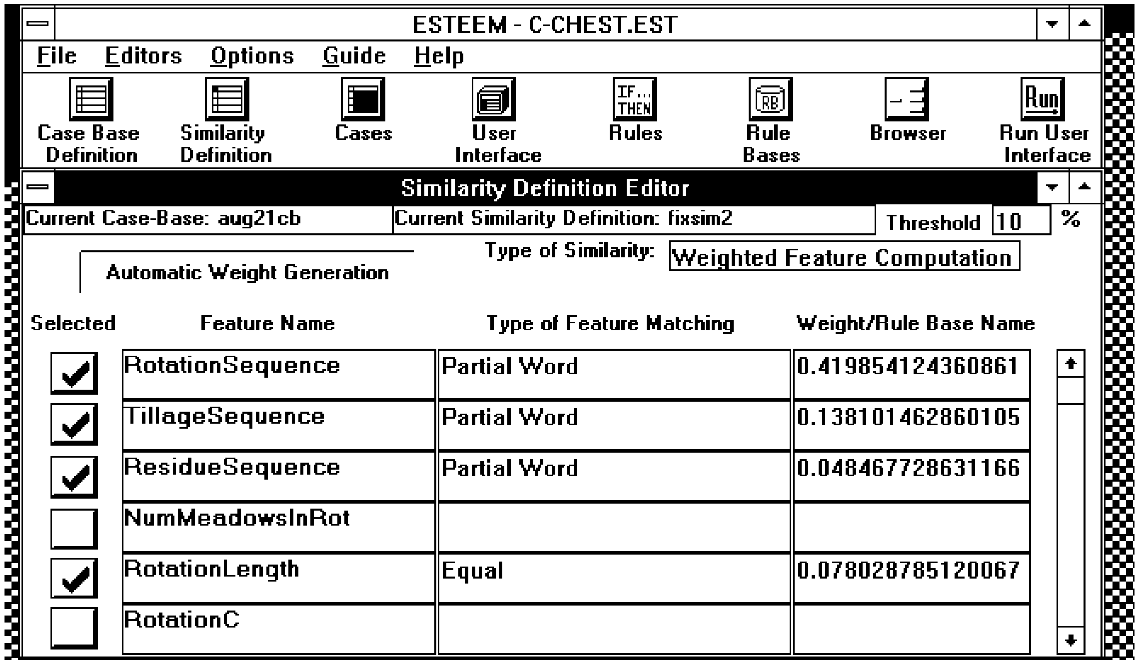


Figure 4.4 Screen capture from Esteem showing the similarity definition editor. The type of similarity between feature values and their weights are defined here.

4.6.2 Feature Matching

Esteem™ offers several kinds of matching options depending on the feature type. In addition to its pre-defined matching options, the developer can also use inferred matching, where specific rules that define similarity between features are allowed. Some of the feature types and the matches allowed are presented in Table 4.4. As most of the features constituted lists of crops and their descriptions, I chose to employ partial matches of strings to facilitate similarity assessment between crop rotations. Exact matches were not used because they do not allow for typographical errors during user input and feature matching. Feature matching for the various features is summarized in Table 4.5.

4.6.3 Feature Weights

The similarity definition also allows the developer to assign relative weights to the selected features. The default weight assigned to any selected feature is one. However, features considered important can be given more weight than the others. For instance, if the feature ‘Rotation Sequence’ is of paramount importance in a rotation then

Table 4.4. Feature Matching in Esteem™.

Feature Type	Matching Allowed
Text	Exact, Partial, Partial Word, Inferred.
Numeric	Equal, Range, Fuzzy Range, Inferred, and Absolute Fuzzy Range.
Case	Exact, Partial, Partial Word, Inferred, and Recursive
Multimedia	Partial, Partial Word, and Inferred.
Boolean	Exact, Inferred.

Table 4.5. Feature matching for the features in the case-base. These features were also used for indexing the case-base.

Feature Name	Matching	Remarks
Rotation Sequence	Partial	Enables the retrieval of cases with similar cropping sequences.
Tillage Sequence	Partial Word	Enables the retrieval of cases with similar tillage practices.
Residue Sequence	Partial	Enables the retrieval of cases with similar residue management.
Meadow Notes	Exact	Rotations with the same meadow characteristics will be considered.
Meadow Duration	Range (50percent)	Rotations with similar meadow lengths will be considered with a tolerance of 50percent.
Rotation Length	Exact	Rotations with the same duration will be treated as similar.
Number Of Crops	Exact	Cases with the same number of crops will be considered similar.
Crops Purposes	PartialWord	Enables the retrieval of cases with crops raised for similar purposes.
Crops Families	PartialWord	Enables the retrieval of cases with crops of the same family (ies).
Crops Descriptors	Partial Word	Enables the retrieval of cases with similar crops.
Tillage Descriptor	Exact	Enables the retrieval of cases with similar tillage practices.
Field Organic Matter	Exact	Enables the retrieval of cases with similar field conditions.
Field Soil Type	Exact	Enables the retrieval of cases with similar field conditions

this feature can be weighted higher by assigning a relative weight of greater than one. Two different weighting schemes were implemented in this project: (i) an *ad hoc* scheme and (ii) an automated weight generation scheme.

4.6.3.1 The *ad hoc* scheme of assigning weights

In the *ad hoc* scheme, features were assigned integer-valued weights indicative of their relative importance in determining similarity between cases. Importance of the features used in the similarity assessment was determined from interaction with experts (Drs. Youngman and Kok, Department of Entomology, Virginia Tech) and study of the NRCS data sheets. However, there were no heuristics for assigning numerical weights to the features in a case. Therefore, I developed the weights empirically using an iterative approach. I tested the similarity between cases for different relative weights of the features (ranging from 0 to 5) in the case-base and compared their outcomes. This comparison was done by looking at the *C*-values of the retrieved cases and the similarity score assessed by the CBR application.

4.6.3.2 Automatic Weight Generation

Esteem™ supports two forms of automatic weight generation schemes. One is the ID3 algorithm (Quinlan, 1986), and the other uses a gradient descent algorithm (GDA).

ID3 was developed from the Concept Learner Systems (CLS) proposed by Hunt et al., (1966). In ID3, ten or fewer cases are randomly selected from the case-base. Using these as training examples, a preliminary discrimination tree is constructed (Patterson, 1992) (i. e., using the features, a hierarchy of similar cases is constructed). For example, a node in the tree is a legume rotation node which has crop rotations that involve legumes in their cropping sequence as its branches. This tree is then tested by scanning all the cases in the case-base to see if there are any exceptions to the tree (Patterson, 1992). Exceptions form new branches or nodes in the tree. This process continues till no further branches arise during testing. Esteem™ then uses this tree to calculate the weights for the features that were used in the tree.

The GDA is an optimization procedure (Hopgood, 1993) that maximizes the similarity (minimizes the distance) between a set of randomly selected cases from the case-base by assigning weights to their features based on a target feature. For instance, in C-Chest, the target feature is the *C*-value of a crop rotation. Esteem™ generates weights for each of the features in the case that determine the *C*-value of that case. The weights so generated are tested on several random groups of cases across the case-base. The weights of the features are then incremented or decremented based on how well the matching cases' source feature and the target feature values match (Esteem™ 1.4 Documentation, 1994). After examining several random cases, the resulting "weight updates" vector is normalized, scaled by a factor delta and added to the current source weight vector. This process continues until delta reaches its minimum value or the user asks Esteem™ to stop (Esteem™ 1.4 Documentation, 1994).

Because the ID3 algorithm is restrictive in its matching capabilities (only exact matches are allowed), I used the GDA to generate weights for the various features used in the case representation. The weights generated using both the *ad hoc* and the automatic weight generation schemes are presented in Table 4.6.

4.6.4 Similarity Score

There are three ways in which similarity scores are computed by Esteem™. One is a feature-counting method that returns the percentage of the number of features selected to the number of features matched. This scheme assumes a default weight of one to all features and just indicates the number of features matched to the total number of features compared.

Second is the inferred feature counting where rules are used to determine the degree of matches of different features. Third is the weighted-average method in which the weights assigned to the features are used to arrive at the similarity score. A score of zero to 100 is generated based on the type of feature match selected (Table 4.4), the weight assigned to that feature (Table 4.6) and the similarity computation method. A similarity score computation example is described below.

Let the input case be a two-year rotation of corn silage and full-season soybeans grown in spring/summer with rye grown as winter cover. Let there be a stored case of a three-year sorghum silage, corn grain and soybean rotation with rye as winter cover. To assess the similarity between this case and the input case (Table 4.7), the CBR system uses the type of feature matches defined, the weights assigned to the features and the method of generating similarity scores. Let the comparison be between the rotation, tillage, and residue sequences of the two rotations as well as their rotation length. The type of matching desired for each of the sequences is ‘partial word’ (i.e., a percentage of the number of strings that match in either features is returned). The weights assigned to these four features are 3.0, 2.0, 1.0, and 1.0 respectively (Table 4.8).

Table 4.6. Similarity metric with normalized feature weights.

Feature Name	**WEIGHTS**	
	Automatic	***ad hoc***
Rotation Sequence	0.22	0.073
Tillage Sequence	0.073	0.054
Residue Sequence	0.025	0.036
Meadow Notes	0.026	0.18
Meadow Number	0	0.073
Rotation Length	0.041	0.145
Number Of Crops	0.4	0.073
Crops Purposes	0.053	0.073
Crops Families	0.031	0.073
Crops Descriptors	0.053	0.073
Tillage Descriptor	0.051	0.145
Field Organic Matter[6]	0.053	0
Field Condition	0.0053	0
Field Soil Type	0.0053	0
Insect Pest History	0.0053	0

[6] Features used for reminding in Pandora only.

Table 4.7. Example Cases.

Feature Name	**Stored Case1**	**Input Case**
Rotation Sequence	CornS, Rye, SoybeansFS, Rye	SorghumS, Rye, CornG, Rye, SoybeansFS, Rye
Tillage Sequence	NoTill, MinimumTill, NoTill, MinimumTill	NoTill, MinimumTill, NoTill, MinimumTill, NoTill, MinimumTill
Residue Sequence	Removed, Left, Left, Left.	Removed, Left, Left, Left, Left, Removed
Rotation Length	2 yrs.	3 yrs.

Based on the partial word matches there is a 66 percent match between the two cases at their rotation, tillage and residue sequences. There is a 100 percent match between the rotation lengths of the two cases as the type of match desired was to allow a range of 50 percent for similarity in this feature. In the feature counting scheme, Esteemô takes these similarity values and computes the overall similarity between these two cases by averaging the similarity of all the features. In the weighted feature computation, the system takes the assigned weights into consideration (Table 4.8) and then generates the similarity scores using a weighted-average. Table 4.9 presents the similarity calculated using the two methods.

4.6.5 Design modifications in the reminding scheme

In the design, reminding was to take place using the indices assigned to the input crop rotation under various hierarchies. However, Esteem™ limits the possibility of hierarchical case representation and thus an explicit knowledge-based indexing scheme, with abstractions, specializations, and instantiations, could not be implemented. However, the indexing scheme that is implemented in this system (using Esteem™) provides for ordered traversal of the data in the case-base based on the features for which the indices were built. These features indirectly impose a hierarchy of cases and case features in the case-base. Nevertheless, the construction of

Table 4.8. Similarity Description.

Feature Name	Feature Matches	Weight
Rotation Sequence	Partial Word	3
Tillage Sequence	Partial Word	2
Residue Sequence	Partial Word	1
Rotation Length	Range (50percent)	1

Table 4.9. Similarity Scores

Feature Name	Feature Counting	Weighted Computation
Rotation Sequence	0.66	1.98
Tillage Sequence	0.66	1.32
Residue Sequence	0.66	0.66
Rotation Length	1.0	1.0
SIMILARITY	**74.5percent**	**70.8percent**

index trees and traversal between cases during similarity assessment is not necessarily knowledge driven as designed in the MOPs-based system in chapter three.

Similarities in MOP-based systems can be explained in terms of domain knowledge and by the presence or formation of various links (see 2.3.1.1) from the case to the objects represented in the hierarchy. In CBR systems using automatic weight generation schemes, similarities between features are assessed not necessarily based on domain knowledge, but are more directly based on the relationship between the input feature(s) and the output feature(s) with conversion of either entities into some numeric from. As such similarity assessment and case retrieval remained largely numeric in this CBR application.

4.7 Adaptation

The systems *C-Chest* and *Pandora* differed in their adaptation procedures because the nature of their outputs was fundamentally different. The output for *C-Chest* was the *C*-value, which is a real number, while the output for *Pandora* was a list of insect pests and their likelihood of requiring control in that crop rotation. Adaptation in both the systems was implemented externally, using the ranking of the retrieved cases.

4.7.1 Adaptation in *C-Chest*

A weighted-average scheme is used for adapting the retrieved case solutions (*C*-value) to the input case. The similarity metric ranks and retrieves cases in the order of their closeness to the current situation and assigns a similarity score (Figure 4.5). This score is used and as a weight for each case's *C*-value, and a weighted-average of the top five retrieved case *C*-values is used as the *C*-value for the input rotation.

For example if the input case is a one-year rotation of corn silage and rye, *C-Chest* retrieves a set of cases similar to this input rotation. Each of these cases has a case-id and a similarity score assigned by the similarity metric. The *C*-value of each of the top five retrieved cases is multiplied by the respective similarity score of the case. This number is summed over five cases and is divided by the total of the five similarity scores. This is the *C*-value predicted by C-Chest for this particular input rotation. Table 4.10 shows some of the retrieved cases and their *C*-values for this input rotation.

4.7.2 Adaptation in *Pandora*

Every case (rotation) includes a set of pests associated with it. Each of

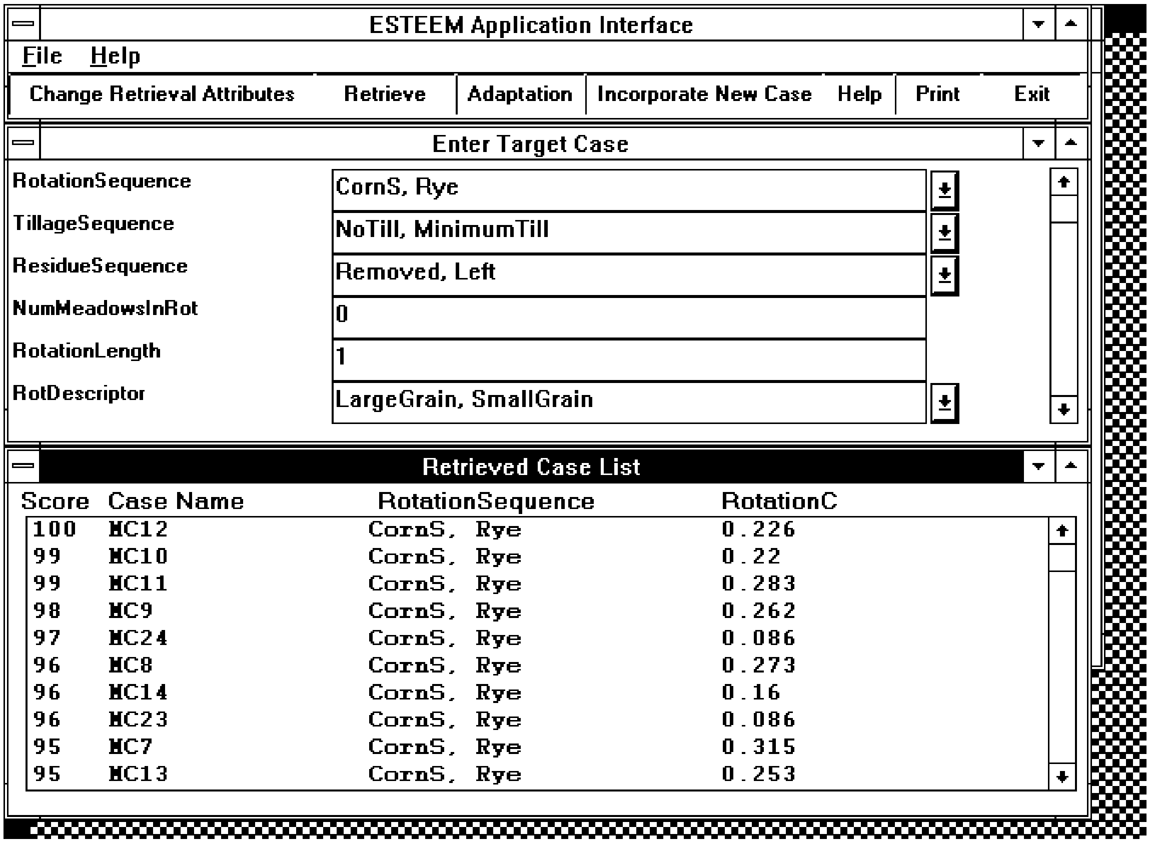


Figure 4.5 Screen capture from Esteem showing retrieved cases (bottom window) in C-Chest. The top window shows the input case.

Table 4.10. Example case adaptation in *C-Chest.*

Similarity Score	**Case Name**	***C*-Value**
99	MC25	0.086
97	MC11	0.283
96	MC9	0.262
96	MC12	0.226
96	MC21	0.137
Weighted Avg.	**(95.97/484)**	**0.198**

Input Case:

Rotation: CornS, Rye; Tillage: Notill, MinimumTill; Residue: Removed, Removed.

For this rotation the *C*-value projected by the system is 0.198 and is derived as shown in the table above.

these pests can exist in one of three levels: high, medium or low risk of requiring control in that rotation. Further, a particular pest might be insignificant in that rotation and therefore might not appear at all. When the system retrieves cases similar to a current input rotation, each of these cases retrieves a list of insect pests associated with it. This information is first converted into a comprehensive list of insect pests that appear in all these retrieved cases. Then a weighted-average of the proportion of cases where each of the pests appears as atleast a low, medium or high risk is taken. This is treated as the risk level of that particular pest in that given rotation.

For example, if the input case is a one-year crop rotation of corn silage followed by rye, Pandora first retrieves a set of cases that are similar to this input rotation based on the feature matches and weights (Figure 4.6. Table 4.11 shows some of the retrieved cases for this input rotation. Each of the retrieved cases includes a list of insect pest risks and a similarity score assigned by Pandora. A weighted-average of each of the high, medium and low risks of each of the insect pests in the top five retrieved cases is taken and normalized. In the example given in Table 4.11 rootworms were a high risk in all the five retrieved cases, so the input rotation is assigned a risk of 1 in the high category of rootworms. A risk of 1 in the high category also implies a risk of 1 in the medium and low categories as well. Likewise a risk of 1 in the medium category implies a risk of one in the low category as well (but not vice-versa).

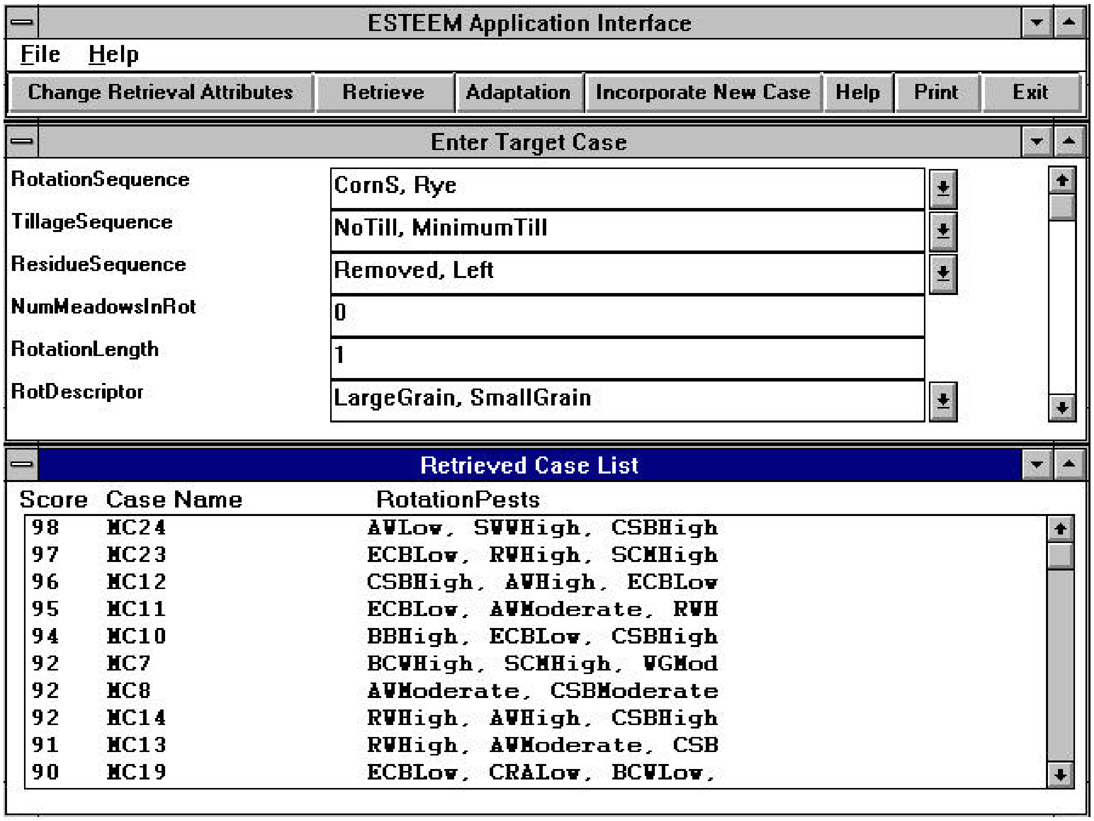


Figure 4.6 Screen capture from Esteem showing retrieved cases (bottom window) in Pandora. The top window shows the input case.

Table 4.11. Example Case retrieval in *Pandora.*

Similarity	**Case**	**Insect Pest Risks**
98	MC14	RW***High***, AW***High***, CSB***High***, BB***High***, ECB***Low.***
96	MC11	ECB***Low***, AW***Medium***, RW***High***, CSB***Medium.***
95	MC12	CSB***High***, AW***High***, ECB***Low***, RW***High.***
94	MC10	BB***High***, ECB***Low***, CSB***High***, RW***High***, AW***High.***
94	MC23	ECB***Low***, RW***High***, SCM***High***, WW***High***, CSB***Low***, AW***Low.***

Input Case:

Rotation: CornS, Rye; Tillage: Notill, MinimumTill; Residue: Removed, Removed.

For this rotation the retrieved cases are shown in Table 4.11. and the insect pest risks projected by the system are presented in Table 4.12.

Armyworms, on the other hand, were a high risk in only three out of five retrieved cases. So a weighted-average of the similarity scores of cases with a high risk of armyworms is taken and assigned to the high category of armyworms. Similarly a weighted-average of the similarity cases with at least a medium risk of armyworms is calculated and assigned to the medium category. Finally, a weighted-average of the similarity cases with at least a low risk of armyworms is calculated and assigned to the low category. Table 4.12 shows the normalized scores for this example.

These risks were then plotted in a area/histogram chart (Fig. 4.7). A cut-off line or action threshold was used to convert this quantitative information back into a qualitative comment. This was done in order to determine what level and/or combination of low, medium and high risk values output by the system constitute actual high risk of requiring control for a given insect pest. If the plot for a given pest transgresses or exceeds this line, then that pest would require control.

The action threshold was developed based on numerical analyses of the normalized scores determined for each of the predicted pests in 50 test cases. It was determined using an algorithm that first to generates lines at different intercepts and different slopes, and then generates error measures for each line. An error occurs whenever Pandora and the known result disagree (i.e., if Pandora predicts a high risk to an insect pest while the known value does not, or vice-versa). This error term was

Table 4.12. Normalized scores for Insect pest risks generated from the retrieved cases.

Insect Pest	**Low**	**Medium**	**High**
Rootworm (RW) [*Diabrotica virgifera virgifera*]	1	1	1
Armyworm (AW) [*Pseudaletia unipuncta]*	1	0.8	0.6
Common Stalkborer (CSB) [*Papaipema nebris*]	1	0.8	0.6
Billbugs (BB) [*Sphenophorus callosus*]	0.19	0.19	0.19
European Corn borer (ECB) [*Ostrinia nubilalis*]	1	0	0
Seed Corn maggot (SCM) [*Hylemya platura*]	0.19	0.19	0.19

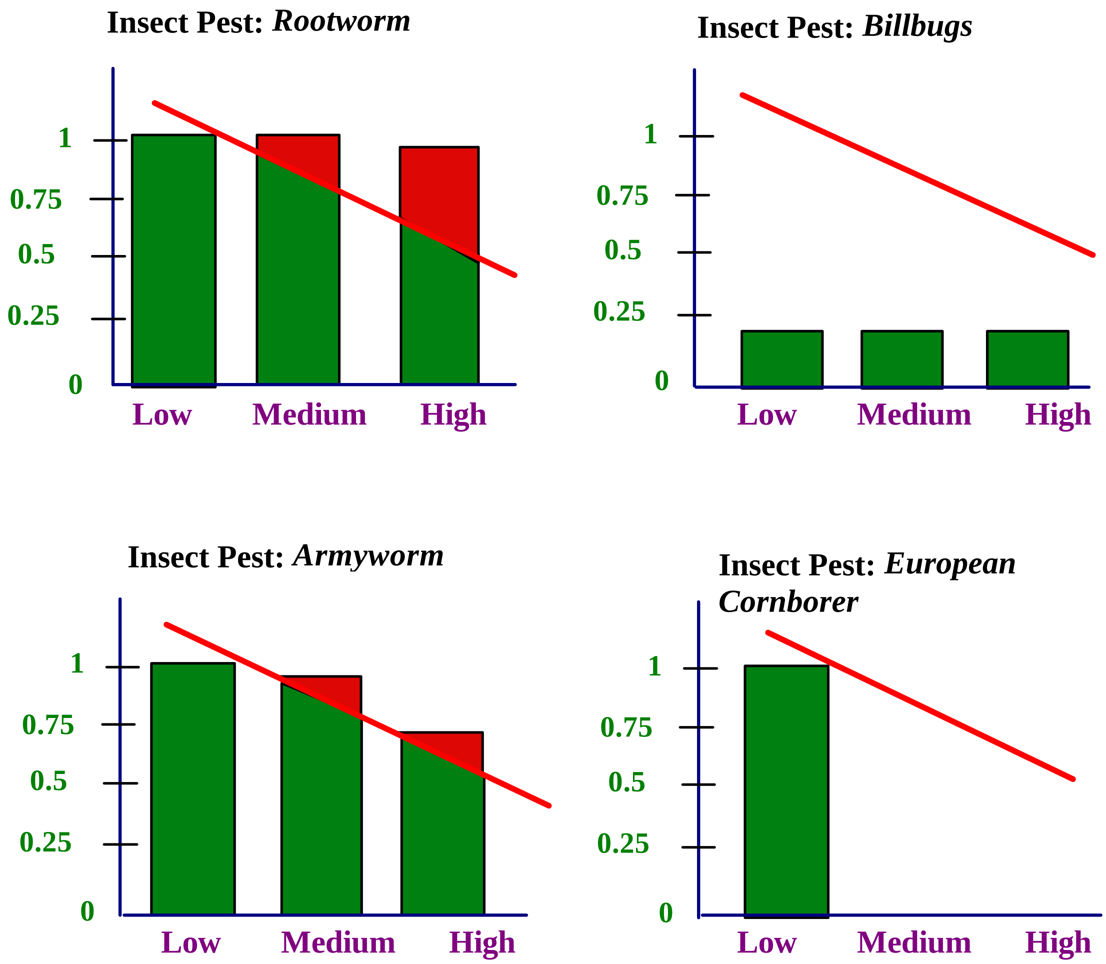


Figure 4.7 Visual representation of insect pest risks. Each of the bars represent the likelihood of the insect pest being a low, medium and high risk respectively. The line shown in the Figure is the “Threshold” which is used as a cut-off line for determining insect pests with high risks.

summed over all the insects over a case and then over all the cases in the CBR. Errors were then represented as a surface, over various slopes and intercepts, to determine the best possible combination of slope and intercept defined by the deepest point on the error surface. Towards this end, I generated surface plots at different resolutions in the range of 0 to 5 for the intercept and +3 to –5 for the slope. Higher resolutions had intervals of one while the best resolution had an interval of 0.1. Figure 4.8 shows the error over a wide range of slopes and intercepts. Based on these results several feasible lines were identified as potential threshold lines. These thresholds were further evaluated by using the remaining cases as tests. However, this error term did not differentiate between the over-predicted and under-predicted cases. It only pointed out the number of cases where the system and the known value disagree. To determine the error in over-prediction and error in under-prediction I further split the error term into an under-predicted error, where the system does not foresee a high risk for a particular insect pest while the known result does; and an over-predicted error, where the system predicts a particular pest to require control while the known result does not.

These error terms were again summed over each case and then over the entire set of cases to determine over-prediction and under-prediction errors. Invariably these two error terms add up to the total error. The best ten lines with their error terms, slopes and intercepts are given in the Table 4.14.

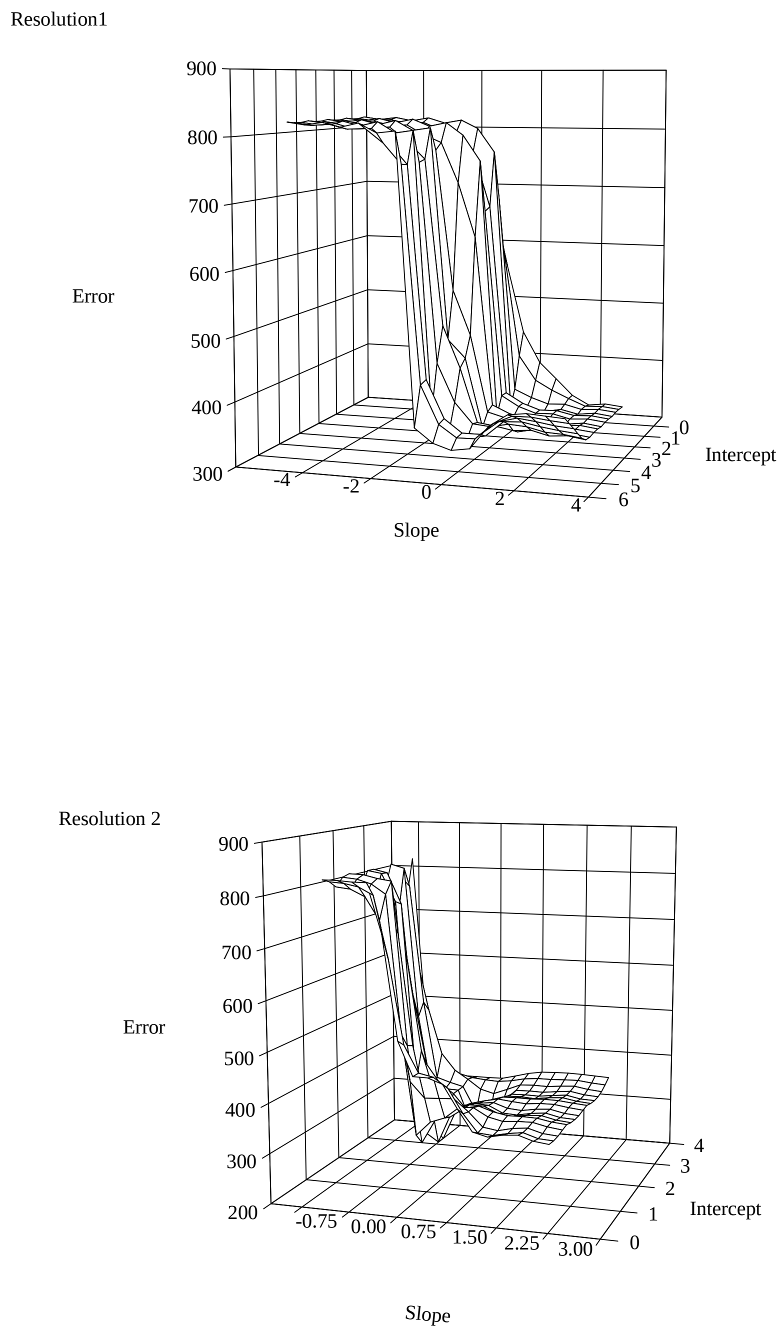
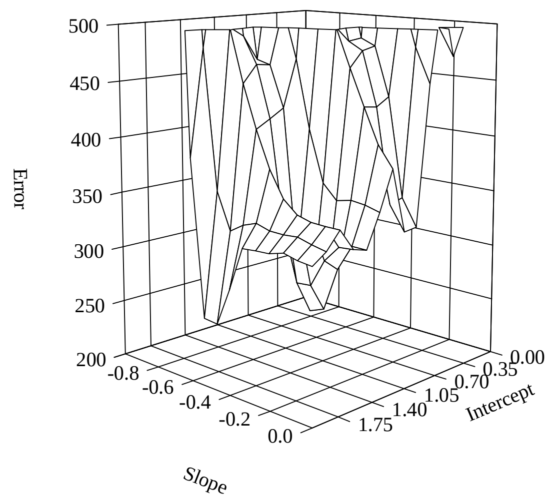


Figure 4.8 Error surface generated in the development of the action threshold.

Resolution 3



Resolution 4

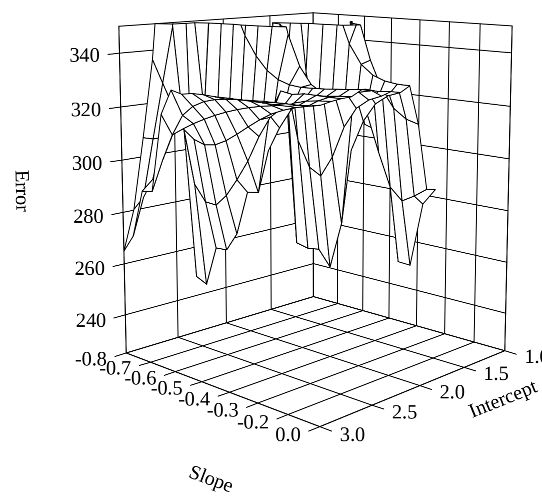


Figure 4.9 Error surface generated (at higher resolutions) in the development of the action threshold.

Table 4.13. Resolution intervals for the various slopes and intercepts used in generating the error surface plots shown in Figure 4.4 a - d.

Resolution	Intercept range	Intercept Interval	Slope range	Slope interval
1	0.0 to 5.0	1.0	+3.0 to - 5.0	-1
2	0.0 to 3.75	0.75	0.0 to -1.5	-0.75
3	0.0 to 2.0	0.5	0.0 to -1.0	-0.25
4	1.0 to 3.0	0.5	0.0 to -1.0	-0.1

Table 4.14. Various threshold lines (intercepts and slopes) that minimize the error term (S). Sover is the over-predicted error and Sunder is the under-predicted error.

Slope	Intercept	S	Sunder	Sover
-0.7	2.5	249	112	137
-0.3	1.5	258	146	112
-0.4	2.0	260	229	31
-0.5	2.0	264	126	138
-0.8	3.0	265	191	74
-0.6	2.5	266	210	56
-0.7	3.0	290	279	11
-0.5	2.5	290	279	11
-0.2	3.0	290	279	11
-0.4	2.5	309	279	11

Once I identified the error terms associated with various lines, I tested the remaining 50 cases to see how well the line performs in determining HIGH risk of insect pests. One of the best lines determined using the algorithm described above also had the best performance in these tests. This line had a slope of -0.3 with a Y-intercept of 1.5. The threshold value in the low category was 1.2, in the medium category was 0.9 and in the high category was 0.6.

4.7.3 Design modifications for Adaptation

Derivational adaptation was the choice method for Pandora. In this scheme, rules associated with individual (retrieved) cases would be tried on the input case to generate a solution. However, there is no provision in Esteem™ that allows storage of rules along with individual cases. Esteem™ allows a more general rule-based case adaptation, in which a set of rules is defined in a rule-base that is used for adapting the solution from one retrieved case. This feature differs significantly from the design that required adaptation to be carried out based on derivational replay of the retrieved case solutions. Further, Esteem™ rule bases do not allow more than 38 rules to be listed for adaptation. This severely limits the extent of adaptation possible between cases based on rules.

Table 4.15 Sample results of the *C-Chest* system. The first column indicates the input case id, the second column gives the actual (known) values of C for each of these rotations, columns under Autosim and AdhocSim present C-values predicted by the automatic and *ad hoc* weight generation schemes respectively. Suffixes 5 and 10 indicate the number of retrieved cases used in computing the C-value for each of the input rotations.

Input Case	**Actual**[7]	**AutoSim5**	**AutoSim10**	**AdhocSim5**	**AdhocSim10**
Recase1	0.107	0.111	0.097	0.088	0.0865
Recase2	0.1067	0.0926	0.078	0.1054	0.1047
Recase3	0.0935	0.0727	0.0766	0.0756	0.0785
Recase4	0.0716	0.077	0.079	0.077	0.073
Recase5	0.08175	0.074	0.074	0.076	0.071
Recase6	0.0652	0.074	0.08	0.068	0.073
Recase7	0.06833	0.0657	0.0719	0.0651	0.069
Recase8	0.0591	0.0637	0.0728	0.068	0.069
Recase9	0.0533	0.0625	0.0728	0.0629	0.0678
Recase10	0.0935	0.0727	0.0766	0.0756	0.0785

[7]Values obtained from the NRCS data tables, the RUSLE model or from the Three-Crop Network model.

4.8 Results

4.8.1 *C-Chest*

C-Chest was run for 204 test cases and two different reminding schemes were evaluated. The first reminding scheme uses *ad hoc* assignment of feature weights and in the second scheme automatically generated weights were used for similarity computations (Table 4.6). In each of these schemes, I tested *C*-values derived from 5 best matches and compared them with the solutions derived form 10 best matches among the retrieved cases. Table 4.15 presents results from these four different sets of *C*-value computations.

The results indicate a close correlation between the known *C*-values and those generated by C-Chest. Among the four different methods of *C*-value computation tested in C-Chest, the methods AutoSim5 and AdhocSim5 were consistently in agreement with the test value than the others. Based on these results I concluded that adaptation from five best matches is adequate and that both the automatic weight generation scheme AutoSim and the *ad hoc* scheme of assigning weights provide satisfactory results.

4.8.2 *Pandora*

One hundred and five test cases were run with the *Pandora* system and their outputs recorded, processed, and plotted in charts. Based on the results of C-Chest, only one similarity scheme was used for case retrieval and similarity computations. AutoSim was the chosen similarity scheme as it was one of the more reliable schemes tested in C-Chest. Further, based on the results of C-Chest, I limited the number of cases required for adaptation to five best matches.

Of the 1148 insect pests predicted by the CBR system (over 100 cases), 258 (22.4 percent) of them were incorrect and with 112 (9.7 percent) of them were over-predicted and 146 (12 percent) were under-predicted. This amounts to an accuracy of about 78 percent in determining high risks of insect pests based on the scores generated by the CBR.

4.9 Summary for Chapter 4

Development of this CBR system included the development of a case-base, an indexing method, a similarity assessment procedure and a retrieval process. Various modifications had to be made to the system design due to several reasons. Esteem™ has restricted in some ways, the development of a system based on domain indices and MOPs-based hierarchies. The flat structure that had to be used posed several challenges in the development of the indexing, similarity and retrieval process. Though efforts were made to incorporate domain knowledge into the system by way of additional features in each of the cases, the reminding and similarity schemes remained primarily numeric. Adaptation was implemented externally due to the lack of suitable procedures in Esteem™. The derivational scheme required the use of case specific rules for adaptation while Esteem™ provided only a generic rule base for adaptation. Transformational adaptation was not feasible due to the large number of case features and limitations of the rule module in Esteem™.

Chapter 5.0

Evaluation and Discussion

This chapter discusses evaluation of the system with details on verification and validation procedures. Literature review on various issues in validation and verification of knowledge-based and case-based systems is presented in chapter two. Contributions from this research, limitations and future directions are also presented here.

5.1 Evaluation

Evaluation of the systems developed in this project was conducted using the validation and verification procedures described in O'Leary (1993). The systems developed were first tested for their syntactic accuracy to make sure that the components of the system were encoded as required for proper functioning. Tests for contents of the case-base, the similarity schemes and adaptation, were then implemented to validate the operation and outputs of the system. The validation tests implemented vary slightly between C-Chest and Pandora due to the nature of their outputs.

5.2 System Verification

Verification measures implemented are common to both C-Chest and Pandora. There are no general algorithms available for objective verification of a case-based system; instead it is left for the developer to discover errors in the application. However, there are several methods that can be used as guidelines for subjective assessment. I used one such guideline mentioned in O'Leary (1993) for verification of my CBR application.

O'Leary's is a four-step verification procedure mentioned in chapter two. I used this procedure for both C-Chest and Pandora. This procedure suggests the verification of a CBR system by testing its consistency in performance, redundancy in representation, completeness of the case-base, and correctness of the reasoning procedures (O'Leary, 1993).

5.2.1 Consistency

To ensure consistency, a strict nomenclature was used to represent objects (crops, tillage operations, etc.). This was done to make sure that the same crop, tillage or residue practice did not get represented under different names or spellings in different cases. Consistency was further tested in the process of cross testing, in which all the cases in the case-base were posed to the system, iteratively, as input cases. If the system were consistent, a case with 100 percent match would be observed for each of the input cases. This test proved useful in identifying a few inconsistencies in the vocabulary used in the cases. Such instances were then corrected.

5.2.2 Redundancy

Redundancy in the system was eliminated by formulating a list of cases before the case-base was developed and by ensuring that the same cases did not get represented in the case-base more than once. For the C-Chest system, data came from the NRCS offices in tables (hard copy and electronic format) which had already been arranged with unique file numbers for each crop rotation and C value. A manual scanning of the case base to identify repeated file numbers was performed to check for redundant cases.

5.2.3 Completeness

Test for completeness involves answering two fundamental questions: (i) is the domain covered completely, and (ii) do all the cases have complete descriptions. To ensure that the domain was covered completely, I developed cases covering all the crop rotations that are commonly practiced in southwestern Virginia. Data to this effect was obtained from the Montgomery county field office of the NRCS in the form of tables. For the C-Chest system I incorporated all the 150 cases provided by the NRCS. Another 65 cases were generated using rotation sequences developed from the CROPS system.

For the Pandora system, 65 cases were generated using rotation sequences developed from the CROPS system and on interaction with experts. Further, 50 of the cases provided by the NRCS with *C*-value data were extended to this system. To ensure that all the cases had complete case descriptions (this is important for the automatic weights generation schemes), I manually scanned the case-base and made sure that data for all the features according to the design was entered. All the cases had complete feature descriptions as defined in chapter three (section 3.5).

5.2.4 Correctness

To ensure correctness, the features in the case-base and the rules that generate derived features were all subject to scrutiny for logic errors of circular references, wrong indexing, placing rotations under wrong hierarchies, etc. Cases were thoroughly compared with the original source(s) to check for incorrect representation of information. All the indexing rules were checked for conflicts and circularity. Further, incorrect case representations were clearly identified and corrected during cross-testing.

5.3 Validation

Validation measures used were different for C-Chest and Pandora. In both cases, however, comparisons with existing sources of knowledge were made to rate the system performance. In the case of C-Chest, comparisons of test cases were made with the data from the NRCS tables and the outputs of the RUSLE model (Renard et al., 1981). In Pandora, I used a combination of procedures for validating the system. As the output of C-Chest is quantitative, I implemented several statistical analyses to test the accuracy of the system and to compare it with the said sources. The validation methods implemented for Pandora, however, were subjective tests.

5.3.1 C-Chest

For those rotations for which the NRCS data were available, comparisons were made between C-Chest's predictions and published NRCS data. For rotations that could be evaluated using the RUSLE model, comparisons were made between the RUSLE output and C-Chest's results. For all other rotations the *C*-values of the test rotations were compared with the three-crop network model as implemented in the CROPS system. Testing of all the cases was not possible with RUSLE as it could not be used for rotations involving barley or millet, or for rotations with more than ten crops in them. Likewise, all the test cases could not be compared with NRCS data as the *C*-values tabulated by NRCS cover only a limited range and type of crop rotations.

In this comparison test, I plotted the observed or actual "C"-values against the predicted "C"-values for all the 205 test cases. An ideal agreement between the output from C-Chest and the actual values would result in all the points falling on a line with a slope of one and zero intercept. To determine how closely the model matched the data I used the residual sum of squares from a linear regression procedure with the line forced through zero and a slope of one for each of the four different reminding schemes.

This test was not intended to make any statistical inferences about the model. Instead, it was designed to help identify the best reminding scheme of the four that were developed. Reminding schemes with lower residual errors were considered better. Results from this test are given in Table 5.1 and in Fig. 5.1.

Of the four different schemes tested ‘AutoSim5’ and ‘AdhocSim5’ resulted in the lowest residual errors. However, on looking at the actual weights assigned to features (Table 4.5) instead of a numerical similarity between cases, the ‘AutoSim5’ scheme has a higher importance and weight placed on the rotation sequence and crops in the rotation than the ‘AdhocSim5’ similarity metric. Thus, in addition to performing consistently, this similarity scheme was more intuitive than ‘AdhocSim5’. Therefore, I concluded that ‘AutoSim5’ was the most reasonable reminding and adaptation scheme of the four different schemes tested in this application. This was the primary focus of the first specific objective.

5.3.2 Pandora

Based on the results of C-Chest, the reminding scheme ‘AutoSim5’ was selected for the insect pest prediction problem. 100 test runs of the system were performed

Table 5.1. Results from validation tests of C-Chest. *C*-values from the four different similarity schemes were tested against the observed values and the residual sum of squares (for a line with a slope of one and an intercept of zero) were tabulated using linear regression procedures implemented in SAS (SAS institute, 1983).

Scheme	**Residual Sum of Squares**
AutoSim5	0.434
AutoSim10	0.469
AdhocSim5	0.376
AdhocSim10	0.498

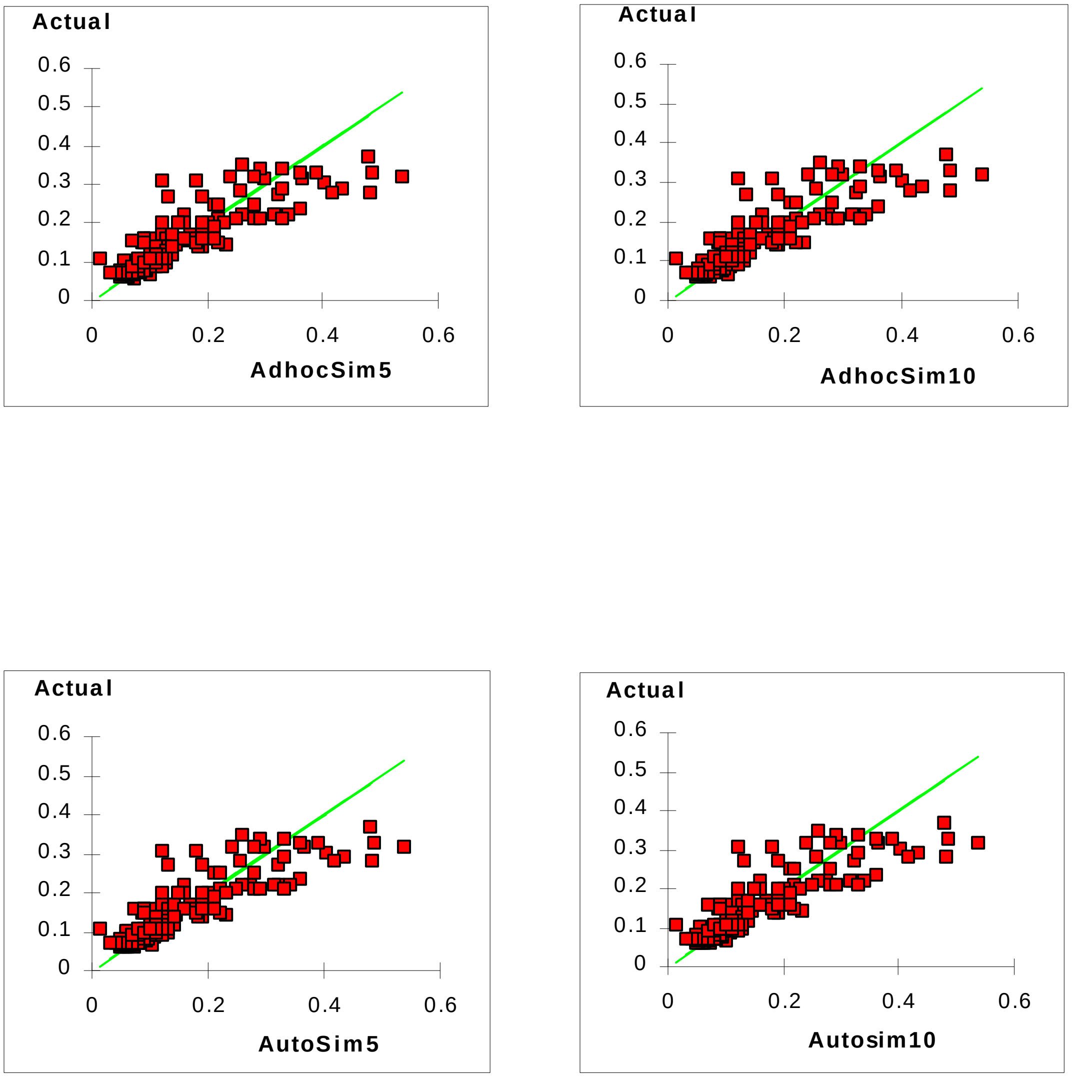


Figure 5.1. Scatter plots of the C-values from the four different reminding schemes plotted against the actual values. The line with slope of one and intercept of zero was used to calculate the residual sum of squares.

during the validation process. Two different tests were carried out for validation of Pandora. The first one was the 'comparison against a target standard', in which I compared the output of Pandora to that of an expert system; and the second one was 'cross-testing', in which the cases from Pandora's case-base were tested iteratively against one another. Comparison against a target standard was implemented as proposed by Harrison (1991) (section 2.7). These were subjective tests conventionally used in the validation of most knowledge-based systems.

5.3.2.1 Comparison Against a Standard Target

Pandora had 30 cases of exclusively corn and small grain crop rotations. This is the most common crop rotation practiced in southwestern Virginia. The available expert system for comparison was VICE-Corn (Buick et al., 1992). VICE-Corn is a rule-based expert system for insect pest management on corn in Virginia. In this test, results of 30 test cases from Pandora were compared with the output from VICE-Corn. The number of correct, over and under-predicted results were counted. Sample results from this test are presented in Table 5.2. Of the 293 insect pest predictions made by Pandora (over 30 cases) 236 (80.5 percent) predictions were in agreement with the output from VICE-Corn. There were 39 (13.3 percent) over-predictions and 18 (6.2 percent) under-predictions made by Pandora.

Table 5.2. Results from validation tests using the comparison against a target standard (results of 12 comparisons out of 30 are presented here).

Test Case id	No. Insect Pests	No. Correct	No. Over-predicted	No. Under-predicted	Percent correct
MC105	13	8	5	0	62
MC2	11	10	0	1	91
MC5	10	9	1	0	90
MC7	8	7	1	0	88
MC8	8	6	0	2	75
MC10	7	6	0	1	86
MC11	8	7	1	0	88
MC12	7	6	1	0	86
MC13	5	3	2	0	60
MC18	9	7	2	0	78
MC15	11	10	1	0	91
MC17	9	6	2	1	67

5.3.2.3 Cross-Testing

All the 100 cases in Pandora's case-base were tested using the cross-testing procedure. This test involved removing cases iteratively from the case-base and posing them to the system as input cases. The output of the system was then compared with the actual insect pest list in the removed case. The number of correct, over-predicted, and under-predicted insect pests for each rotation was then calculated.

Pandora had an accuracy of 78 percent in determining the high risk of insect pests in the cross-testing procedure. There were 10 percent over-predictions and 12 percent under-predictions made by Pandora. However, the action threshold developed in chapter four provides the end-user with an option to lower either the under-prediction error or the over-prediction error and thereby increase the overall accuracy. Results from cross-testing are given in Table 5.3.

Evaluation of prototypes is important for further development of a CBR. However, there is no set threshold for considering a prototype to be satisfactory. Talebzadeh et al., (1995) report a system accuracy of 70 percent while alpha testing their prototype KBS and found it acceptable to proceed with further enhancements and fine tuning. PROTOS (Porter et al., 1990) had an accuracy of 90 percent over its entire case-base (Kolodner, 1993) in classifying cases before it was enhanced to develop a 100 percent accuracy rate.

In general, the acceptable degree of accuracy is dependent on the domain and the intended use of the system. One of the major advantages of CBR is that it is possible to build and field a system with a small library of cases and allow the knowledge-base to be expanded and refined over time (Hinkle and Toomey, 1995). Therefore, the performance of a CBR can potentially improve over time and use. Based on these observations and because Pandora was intended to be a prototype CBR system, I concluded that its performance at 78-80 percent accuracy was satisfactory.

5.4 Limitations

Evaluation was not comprehensive due to several limitations. Availability of experts and/or other expert systems was a significant constraint during this process. Test cases were evaluated for only one crop, corn, in the comparison test and found to be 80 percent accurate. Insect pests risks on other crops in the rotations could not be verified using this or other validation procedures like establishing face validity and the Turing test.

Table 5.3. Sample results from cross-testing

Test Case id	Number Predicted	Number Correct	Number Over-Pred.	Number Under-Pred.
RECASE1	11.0	9.0	1	1
RECASE2	12.0	9.0	0	3
RECASE3	14.0	9.0	0	5
RECASE4	14.0	10.0	0	4
RECASE6	13.0	9.0	3	1
RECASE7	12.0	11.0	0	1
RECASE8	11.0	8.0	0	3
RECASE9	11.0	8.0	0	3
RECASE12	13.0	8.0	4	1
RECASE13	12.0	9.0	0	3
RECASE14	14.0	11.0	3	0
RECASE15	14.0	11.0	3	0

However, given that Pandora has been found satisfactory in predicting the insect pest risks in one crop in a rotation, and that all formal consistency checks were employed during the development process, the system should perform as well in predicting insect pests risks in other crops in the rotation. Pandora was developed to look at representing and reasoning about the complex crop rotation process, and hence the tests were focused on the suitability of this approach rather than the exact numbers or pest lists.

5.5 Summary and Conclusions

Incorporation of IPM strategies into whole-farm planning is an essential step in achieving the goals of both IPM and WFP. The automated planning system, CROPS, requires an adequate methodology to assess the impacts of cropping plans on pests and pesticide risks. Case-based reasoning can be used to represent and reason about complex domains. This research explored this methodology and provided a framework to develop a robust CBR system that will allow CROPS to generate pest risks for a given crop rotation. Specific contributions from this work include:

- Development of a knowledge base on the effects of rotations on pest populations that can be used in determining the environmental risks associated with crop rotations.

- Demonstration of the CBR technology for successfully utilizing existing expertise to solve the complex problem of incorporating preventive IPM strategies into WFP.

- Initial development of a region-specific knowledge base on the contribution of crop rotations to soil conservation in the form of *C* factors. This system also allows efficient regeneration of *C* factors for different cropping situations that can be used by the planner in determining the soil erosion risks associated with a crop rotation.

5.5.1 Recommendations for Future Research

The software used in this project was useful for rapid proto-typing and technology demonstration. The design process has helped us in creating a framework for further system evolution and development. Development of a practical extension to the CROPS system would require robust indexing and similarity assessment procedures not available in most of the commercial CBR tools. The CBR so developed can be implemented as an automated learning system as well. CBR systems can be programmed to accept new cases and dynamically update their case-bases. This feature facilitates the improvement of the system over time. Effective domain-specific algorithms are required to be able to implement a CBR system to its fullest potential.

References

Aamodt, Agnar; Plaza, Enric. 1994. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. AICOM. 7(1).

Adam, Johan D. 1991. Failure diagnostic expert systems: A case study in fault diagnosis. Master's thesis in systems engineering, Virginia Polytechnic Institute and State University.

Adrion. W; Branstad . M; Cherniavsky. J. 1982. Validation, verification, and testing of computer software. ACM computer surveys. 14(2).

Bain, W. 1986. Case-based reasoning: A computer model of subjective assessment. Ph. D dissertation., Department of computer science, Yale University.

Boehlje, M. D. and Eidman, V. R. 1984. Eds. Farm Management,. John Wiley and Sons. New York.

Brady, Nyle C. 1990. The nature and properties of soils. Tenth Edition. Macmillan Publishing company, New York.

Bridge, Galen. 1993. Is Whole-Farm Conservation planning the answer? Journal of Soil and Water conservation. July.

Buick, R. D.; Stone, N. D.; Scheckler, R. K.; Roach, J. W. 1992. CROPS: A whole-farm crop rotation planning system to implement sustainable agriculture. AI Applications 6(3).

Buick, R. D.; Youngman, R. R. and Stone, N. D. 1993. VICE-Corn: An expert system for insect pest management in virginia field corn. AI Applications 7(2&3.).

Buta, Paul. 1994. Mining for financial knowledge with CBR. AI Expert 9(2).

Carrascal, M. J.; Pau, L. F. 1992. A survey of expert systems in agriculture and food processing. AI Applications 6(2).

Chi, R. T. H.; Kiang, M. Y. 1991. An integrated approach of rule-based and case based reasoning for decision support. 1991 ACM Computer science conference, Preparing fro the 21st century. Proceedings.

Coulson, Robert N.; Saunders, Michael C. 1987. Computer-assisted Decision-making as applied to Entomology. Annual Review of Entomology 32(?).

DeJong, D. and Mooney, R. 1986. Explanation-based learning: an alternative view. Machine learning 1(2).

Dunn, G. and Everitt, B. S. 1982. An introduction to mathematical taxonomy. Cambridge university press.

Dutta, Soumitra and Bonissone, Piero P. 1993. Integrating case and rule based reasoning. International journal of approximate reasoning 8.

Engel, B. A.; Jones, D. D. and Thompson, T. L. 1992. Integrating expert systems with traditional computer-based problem solving techniques. AI Applications 6(2).

Farris, J. W. 1970. Methods for computing wagner trees. Systematic Zoology 19.

Fleming, Skov. 1990. Rule-based expert systems. Paper presented at a workshop on expert systems in agricultural research; Tjele; Denmark December.

Flint, M. L.; Roberts, P. A. 1988. Using crop diversity to manage pest problems: Some california examples. American Journal of Alternative Agriculture 3(163).

Flint, Mary Louise; Van Den Bosch, Robert. 1981. Introduction to Integrated Pest Management. Plenum Press, New York and London.

Funderburk, J.; Higley, L. and Buntin, G. 1993. Concepts and directions in arthropod pest management. Advances in agronomy 51.

Gaines, B. 1991. The trade-off between knowledge and data in knowledge acquisition,. In G. Patetsky-Shapiro and W. Frawley (Eds.), Knowledge discovery in databases. Cambridge, MA: AAAI/MIT Press.

Goodman, M. 1989. CBR in battle planning. In DARPA Los Altos, CA: Morgan Kaufmann.

Hammond, K. J. 1989. Case-based planning: Viewing planning as a memory task. Boston: Academic press.

Harrison, S. R. 1991. Validation of expert systems. Agricultural systems 35.

Hennessy, D. and Hinkle, D. 1992. Applying case-based reasoning to autoclave loading. IEEE Expert 7(5).

Higley, L. G. , and Wintersteen, W. K. 1992. A novel approach to environmental risk assessment of pesticides as a basis for incorporating environmental costs into economic injury levels. American Entomologist 38(1).

Higley, L. G. and Pedigo, L. P. 1991. Soybean yield responses and intraspecific competetion from simulated seedcorn maggot injury. Agronomy journal 83.

Hill, S. B. 1990. Pest control in sustainable agriculture. Proceedings of the entomological society of Ontario 121.

Hopgood, Adrion. 1993. . Knowledge-based systems for engineers and scientists. CRC Press Inc., Boca Raton, FL. USA.

Huffaker, C. B. , and Smith, R. F. 1980. Rationale, Organization, and development of a national integrated pest management project. In "New technology of pest control" (C. B. Huffaker, Ed.). Wiley-Interscience, New York.

Hunt, R.; Middleton, D. A. J.; Grime, J. P.; Hodgson, J. G. 1991. TRISTAR: an expert system for Vegetation Processes. Expert Systems Nov. 8(4).

Hutchinson, J. 1926. The families of flowering plants, I. Dicotyledons. Macmillan and Co,. ltd.

Hutchinson, J. 1934. The families of flowering plants, II. Monocotyledons. Macmillan and Co, ltd.

Kay, R. D. and Edwards, W. M. (Eds.). 1994. Farm Management. 3rd Ed. McGraw-Hill., New York.

Kolodner, J. L. 1993. Case-based reasoning. Morgan Kaufmann Pubs.

Kolodner, J. L. , and Simpson, R. L. 1985. The Mediator: Analysis of an early case-based problem solver. Cognitive science 13(4).

Kolodner, J. L. 1983. Reconstructive memory: A computer model. Cognitive science 7(4).

Koton, P. 1989. Using experience in learning and problem solving. Ph.D. dissertation., Department of computer science, MIT.

Mark, W. 1989. Case-based reasoning for autoclave management. In Proceedings: Workshop on case-based reasoning (DARPA), Pensacola beach, Florida. San Mateo, CA: Morgan Kaufmann.

McKinion, J. M.; Lemmon, H. E. 1985. Expert systems for Agriculture. Computers and Electronics in Agriculture May; 1.

McKinion, J. M. 1992. Basics of Modeling Strategies, in Basics of Insect Modeling. Goodenough, L. L and McKinion, J. M Eds. ASAE Monograph number 10.

Metcalf, R. L.; Luckmann, W. H. 1994. Introduction to insect pest management. Third edition. John Wiley and Sons, inc.

Mitchell, T. M.; Keller, R. M. and Kedar-Cabelli, S. T. 1986. Explanation-based generalization: a unifying view. Machine learning 1(1).

Nevo, Amnon; Oad, Ramchand; Podmore, Terence H. 1994. An integrated expert system for optimal crop planning. Agricultural systems 45.

O'Leary, Daniel E. 1993. Verification and Validation of Case-Based systems. Expert systems with applications 6.

Patterson, D. W. 1992. Introduction to artificial intelligence and expert systems. Prentice-Hall of India private limited.

Pedigo, L. P. 1989. Entomology and pest management. Macmillian, New York.

Plant, R. E. and Stone, N. D. 1991. Knowledge-based systems in agriculture. McGraw-Hill, Inc., NY.

Porter, B. W.; Bareiss, E. R. and Holte, R. C. 1990. Knowledge acquisition and heuristic classification in weak-theory domains.

Rabb, R. L. 1972. Principles and concepts of pest management. Pages 6-29 in Implementing practical pest management strategies. Proceedings of a national pest management workshop. Purdue University, Lafayette, IN.

Renard, K. G.; Foster, G. R.; Yoder, D. C. and McCool, D. K. 1994. RUSLE revisited: Status, questions, answers, and the future. Journal of soil and water conservation. May.

Riesbeck, Christopher K.; Schank, Roger C. 1989. Inside Case-Based Reasoning. Lawrence Erlbaum Associates, Publishers, Hillsdale, NJ.

R.T.H. Chi; M.Y. Kiang. 1993. Reasoning by coordination: an integration of case-based and rule-based reasoning systems. Knowledge-Based systems. Jun; 6(2).

Ruby, D. and Kibler, D. 1988. Exploration of case-based problem solving. In J. Kolodner (Ed.), Proceedings: Case-based reasoning workshop. San Mateo, CA: Morgan Kaufmann.

Saarenmaa, H. 1992. Integrated pest management in forests and information technology. Journal of applied entomology 114.

SAS institute inc. 1988. Cary, North Carolina. USA.

Schank, Roger C. 1982. Dynamic memory: A theory of learning in computers and people. Cambridge university press. NY.

Shoemaker, C. A. and Onstad, D. W. 1983. Optimization analysis of the integration of biological, cultural, and chemical control of Alfalfa weevil (Coleoptera: Curculionidae). Environmental Entomology(12) April.

Simpson, R. L. 1985. Computer model of case-based reasoning in problem solving: An investigation in the domain of dispute mediation. Georgia Institute of Technology, School fo information and computer science technical report no. GIT-ICS-85/18. Atlanta.

Sneath, Peter H. A and Sokal, Robert. R. 1973. Numerical taxonomy: The principles and practices of numerical classification. W. H. Freeman and Co., San Francisco.

Starfield, A. M. and Bleloch, A. L. 1983. Expert systems: An approach to problems in ecological management that are difficult to quantify. Journal of environmental management 16.

Stone, N. D. 1989. Knowledge-based systems as a unifying paradigm for IPM. Proceedings national integrated pest management symposium/workshop. Las Vegas, Nevada. USA.

Stone, N. D. 1995. Whole farm planning for crop/livestock farms. Proceedings of farm animal computer technologies conference (FACTs) '95. Orlando, FL .

Stone, N. D.; Buick, R. D.; Roach, J. W.; Scheckler, R. K. and Rupani, Rajnish. 1992. The planning problem in agriculture: Farm level crop rotation planning as an example. AI Applications 6(1).

VCE publication 456 - 016. 1993. VCE Publication 456-016 for field crops. Virginia Cooperative Extension publication.

VCE Publication 446 - 0447. 1993. Virginia farm management crop and livestock enterprise budgets. Virginia Co-operative Extension publication.

Warren, P. L 1994. VICE-Wheat: An expert system for insect pest management in virginia winter wheat. Personal Communication.

Wintersteen , W. K. and Higley, L. G. 1993. Advancing IPM systems in corn and soybeans. In "Successful implementation of integrated pest management for agricultural crops" (A. R. Leslie and G. W. Cuperus, eds.). Lewis publsihers., Chelsea, MI.

Wischmeier, W. H. 1970. Cropping management factor evaluations for a universal-soil loss equation. Soil science society proceedings.

Wischmeier, W. H. and Smith, D. D., 1958. Rainfall energy and its relationship to soil loss. Transactions of the american geo-physics union 39:285-291.

Glossary

Adaptation: Is the process of fixing up an old solution to meet the demands of the new situation (Kolodner, 1993).

Case: A case is a contextualized piece of knowledge representing an experience that teaches a lesson fundamental to achieving the goals of the reasoner (Kolodner, 1993).

Case-based reasoning: A case-based reasoner solves new problems by adapting solutions that were used to solve old problems (Riesbeck and Schank, 1989).

Derivational adaptation: Rules that generated the original solution are re-run to henerate a new solution (Riesbeck and Schank, 1989).

Feature: A feature of a case is an attribute-value pair used in the description of a case (Kolodner, 1993) (syn. Descriptor).

Indexing: In broad terms, it means finding in memory the experience closest to a new situation (Kolodner, 1993).

MOP: Memory Organization Package. Is the basic unit in dynamic memory and is used to represent knowledge about classes of events, especially complex events (Riesbeck and Schank, 1989).

Norms: Norms represent the basic features of a MOP (Riesbeck and Schank, 1989).

Pest Risk: Indicates if a particular pest requires intervention.

Planning: Is the process of generating a sequence of operations to achieve an end state or goal (Hendler et al, 1990).

Transformational adaptation: Adaptation rules from a general working memory apply directly to the solution in a stored case to modify it for the new case (Riesbeck and Schank, 1989).

Whole-farm planning: Is a systems approach to agricultural management that involves the development of cropping plans with careful consideration to environmental and economic issues.

APPENDIX

Example cases from Pandora

$NEWCASE$ MC1

RotationSequence: CornS, Fallow
TillageSequence: FallConventional, Fallow
ResidueSequence: Removed, Fallow
RotDescriptor: LargeGrain, Fallow
RotFamilies: Graminae, Fallow
TillDesc: Conventional, Fallow
CropsPurposes: Silage, Fallow
FieldOrganicMatter: Moderate
FieldSoilType: Mixtures
FieldCondition: Moderate
PriorPestHistory: Anthills Present
RotationPests: CRAModerate, ECBLow, SBLow, AWLow, RWHigh, BCWLow

$NEWCASE$ MC2

RotationSequence: CornS, Fallow
TillageSequence: SpringConventional, Fallow
ResidueSequence: Removed, Fallow
RotationLength: 1
RotDescriptor: LargeGrain, Fallow
RotFamilies: Graminae, Fallow
TillDesc: Conventional, Fallow
CropsPurposes: Silage, Fallow
FieldOrganicMatter: Moderate
FieldSoilType: Mixtures
FieldCondition: Moderate
PriorPestHistory: Billbug
RotationPests: WWModerate, BBHigh, SBLow, ECBLow, BCWLow, RWHigh
PestsInfoSource: PMG/Interview; VICECorn

$NEWCASE$ MC5

RotationSequence: CornS, Rye
TillageSequence: SpringConventional, MinimumTill
ResidueSequence: Removed, Left
RotationLength: 1
RotDescriptor: LargeGrain, SmallGrain
RotFamilies: Graminae, Graminae
TillDesc: Conventional, Conservation
CropsPurposes: Silage, WinterCover

FieldOrganicMatter: Moderate
FieldSoilType: Mixtures
FieldCondition: Moderate
PriorPestHistory: None
RotationPests: SCMLow, WWLow, AWModerate, CSBModerate, CRALow, RWModerate
PestsInfoSource: PMG/Interview; VICE-Corn

$NEWCASE$ MC7

RotationSequence: CornS, Rye
TillageSequence: MinimumTill, MinimumTill
ResidueSequence: Removed, Removed
RotDescriptor: LargeGrain, SmallGrain
RotFamilies: Graminae, Graminae
TillDesc: Conservation, Conservation
CropsPurposes: Silage, WinterCover
FieldOrganicMatter: Moderate
FieldSoilType: Clay
FieldCondition: Weedy
PriorPestHistory: None
RotationPests: BCWHigh, SCMHigh, WGModerate, ECBLow, RWHigh
PestsInfoSource: PMG/Interview; VICE-Corn
Other: Poor drainage soils and weedy fields are at high risk for cutworms.

$NEWCASE$ MC8

RotationSequence: CornS, Rye
TillageSequence: MinimumTill, MinimumTill
ResidueSequence: Removed, Left
NumMeadowsInRot: 0
RotationLength:1
RotDescriptor: LargeGrain, SmallGrain
RotFamilies: Graminae, Graminae
TillDesc: Conservation, Conservation
CropsPurposes: Silage, WinterCover
FieldOrganicMatter: Moderate
FieldSoilType: Mixtures
FieldCondition: Moderate
PriorPestHistory: None
RotationPests: AWModerate, CSBModerate, RWHigh, ECBLow
PestsInfoSource: PMG/Interview; VICE-Corn

$NEWCASE$ MC10

RotationSequence: CornS, Rye
TillageSequence: MinimumTill, NoTill
ResidueSequence: Removed, Left

RotDescriptor: LargeGrain, SmallGrain
RotFamilies: Graminae, Graminae
TillDesc: Conservation, Conservation
CropsPurposes: Silage, WinterCover
FieldOrganicMatter: Moderate
FieldSoilType: Mixtures
FieldCondition: Moderate
PriorPestHistory: Billbugs
RotationPests: BBHigh, ECBLow, CSBHigh, RWHigh, AWHigh
PestsInfoSource: PMG/Interview; VICE-Corn
Other: Rye residue if left, can cause common stalk borer problems.

$NEWCASE$ MC11
RotationSequence: CornS, Rye
TillageSequence: NoTill, MinimumTill
ResidueSequence:Removed, Removed
RotDescriptor: LargeGrain, SmallGrain
RotFamilies: Graminae, Graminae
TillDesc: Conservation, Conservation
CropsPurposes: Silage, WinterCover
FieldOrganicMatter: Moderate
FieldSoilType: Mixtures
FieldCondition: Moderate
PriorPestHistory: None
RotationPests: ECBLow, AWModerate, RWHigh, CSBModerate
PestsInfoSource: PMG/Interview; VICE-Corn
Other: Corn is deemed continuous

$NEWCASE$ MC12

RotationSequence: CornS, Rye
TillageSequence: NoTill, MinimumTill
ResidueSequence: Removed, Left
RotDescriptor: LargeGrain, SmallGrain
RotFamilies: Graminae, Graminae
TillDesc: Conservation, Conservation
CropsPurposes: Silage, WinterCover
FieldOrganicMatter: Moderate
FieldSoilType: Mixtures
FieldCondition: Moderate
PriorPestHistory: None
RotationPests: CSBHigh, AWHigh, ECBLow, RWHigh
PestsInfoSource: PMG/Interview; VICE-Corn

$NEWCASE$ MC13

RotationSequence: CornS, Rye

TillageSequence: NoTill, NoTill
ResidueSequence: Removed, Removed
RotDescriptor: LargeGrain, SmallGrain
RotFamilies: Graminae, Graminae
TillDesc: Conservation, Conservation
CropsPurposes: Silage, WinterCover
FieldOrganicMatter: Moderate
FieldSoilType: Mixtures
FieldCondition: Moderate
PriorPestHistory: None
RotationPests: RWHigh, AWModerate, CSBModerate, ECBLow
PestsInfoSource: VICE-Corn
Other: Corn is deemed continuous

$NEWCASE$ MC14

RotationSequence: CornS, Rye
TillageSequence: NoTill, NoTill
ResidueSequence: Removed, Left
RotDescriptor: LargeGrain, SmallGrain
RotFamilies: Graminae, Graminae
TillDesc: Conservation, Conservation
CropsPurposes: Silage, WinterCover
FieldOrganicMatter: Moderate
FieldSoilType: Mixtures
FieldCondition: Moderate
PriorPestHistory: Billbugs
RotationPests: RWHigh, AWHigh, CSBHigh, BBHigh, ECBLow
PestsInfoSource: PMG/Interview; VICE-Corn
Other: Corn is deemed continuous

$NEWCASE$ MC15

RotationSequence: CornS, Fallow
TillageSequence: FallConventional, Fallow
ResidueSequence: Removed, Fallow
RotDescriptor: LargeGrain, Fallow
RotFamilies: Graminae, Fallow
TillDesc: Conventional, Fallow
CropsPurposes: Silage, Fallow
FieldOrganicMatter: High
FieldSoilType: SandyLoams
FieldCondition: Clean
PriorPestHistory: None
RotationPests: BCWLow, SCMModerate, WGModerate, ECBLow, RWModerate, AWLow
PestsInfoSource: PMG/Interview; VICE-Corn

$NEWCASE$ MC16

RotationSequence: CornS, Fallow
TillageSequence: SpringConventional, Fallow
ResidueSequence: Removed, Fallow
RotDescriptor: LargeGrain, Fallow
RotFamilies: Graminae, Fallow
TillDesc: Conventional, Fallow
CropsPurposes: Silage, Fallow
FieldOrganicMatter: Moderate
FieldSoilType: Mixtures
FieldCondition: Moderate
PriorPestHistory: None
RotationPests: SCMLow, AWLow, WWModerate, SWWLow, RWModerate, ECBLow
PestsInfoSource: PMG; VICE-Corn; Interview

$NEWCASE$ MC17

RotationSequence: CornS, Fallow
TillageSequence: MinimumTill, Fallow
ResidueSequence: Removed, Fallow
RotDescriptor: LargeGrain, Fallow
RotFamilies: Graminae, Fallow
TillDesc: Conservation, Fallow
CropsPurposes: Silage, Fallow
FieldOrganicMatter: Moderate
FieldSoilType: Mixtures
FieldCondition: Moderate
PriorPestHistory: Billbugs
RotationPests: SWWModerate, WWModerate, AWLow, BBHigh, SCMHigh, RWModerate, ECBLow
PestsInfoSource: PMG/Interview; VICE-Corn

$NEWCASE$ MC18

RotationSequence: CornS, Fallow
TillageSequence: NoTill, Fallow
ResidueSequence: Removed, Fallow
RotDescriptor: LargeGrain, Fallow
RotFamilies: Graminae, Fallow
TillDesc: Conservation, Fallow
CropsPurposes: Silage, Fallow
FieldOrganicMatter: Moderate
FieldSoilType: Mixtures
FieldCondition: Moderate

PriorPestHistory: None
RotationPests: WWHigh, AWLow, RWModerate, SCMHigh, ECBLow, CSBLow
PestsInfoSource: PMG/Interview; VICE-Corn

$NEWCASE$ MC19

RotationSequence: CornS, Rye
TillageSequence: SpringConventional, MinimumTill
ResidueSequence: Removed, Left
RotDescriptor: LargeGrain, SmallGrain
RotFamilies: Graminae, Graminae
TillDesc: Conventional, Conservation
CropsPurposes: Silage, WinterCover
FieldOrganicMatter: Moderate
FieldSoilType: Mixtures
FieldCondition: Moderate
PriorPestHistory: None
RotationPests: ECBLow, CRALow, BCWLow, RWWLow, RWModerate, CSBLow, AWLow
PestsInfoSource: PMG/Interview; VICE-Corn

$NEWCASE$ MC23

RotationSequence: CornS, Rye
TillageSequence: NoTill, MinimumTill
ResidueSequence: Removed, Removed
RotDescriptor: LargeGrain, SmallGrain
RotFamilies: Graminae, Graminae
TillDesc: Conservation, Conservation
CropsPurposes: Silage, WinterCover
FieldOrganicMatter: Moderate
FieldSoilType: Mixtures
FieldCondition: Moderate
PriorPestHistory: None
RotationPests: ECBLow, RWHigh, SCMHigh, WWHigh, CSBLow, AWLow
PestsInfoSource: PMG/Interview; VICE-Corn

$NEWCASE$ MC24

RotationSequence: CornS, Rye
TillageSequence: NoTill, MinimumTill
ResidueSequence: Removed, Left
RotDescriptor: LargeGrain, SmallGrain
RotFamilies: Graminae, Graminae
TillDesc: Conservation, Conservation

CropsPurposes: Silage, WinterCover
FieldOrganicMatter: Moderate
FieldSoilType: SiltLoams
FieldCondition: Moderate
PriorPestHistory: None
RotationPests: AWLow, SWWHigh, CSBHigh, WWHigh, RWHigh, ECBLow
PestsInfoSource: PMG/Interview; VICE-Corn

$NEWCASE$ MC105

RotationSequence: CornS, Rye, Meadow
TillageSequence: SpringConventional, MinimumTill, MinimumTill
ResidueSequence: Removed, Removed
RotDescriptor: LargeGrain, SmallGrain, Meadow
RotFamilies: Graminae, Graminae, Meadow
TillDesc: Conventional, Conservation, Conservation
CropsPurposes: Silage, WinterCover, Meadow
FieldOrganicMatter: Moderate
FieldSoilType: Mixtures
FieldCondition: Moderate
PriorPestHistory: None
RotationPests: WWHigh, SCMHigh, AWLow, RWLow, CSBLow, ECBLow
PestsInfoSource: PMG/Interview; VICE-Corn

$NEWCASE$ MC43

RotationSequence: CornS, Rye, Meadow
TillageSequence: MinimumTill, MinimumTill, MinimumTill
ResidueSequence: Removed, Removed
RotDescriptor: LargeGrain, SmallGrain, Meadow
RotFamilies: Graminae, Graminae, Meadow
TillDesc: Conservation, Conservation, Conservation
CropsPurposes: Silage, WinterCover, Meadow
FieldOrganicMatter: High
FieldSoilType: Clay
FieldCondition: Weedy
PriorPestHistory: BB
RotationPests: BBHigh, AWHigh, BCWHigh, WWHigh, RWLow, SCMHigh, ECBLow
PestsInfoSource: PMG/Interview; VICE-Corn

$NEWCASE$ MC53

RotationSequence: CornS, Rye, Meadow
TillageSequence: NoTill, MinimumTill, MinimumTill
ResidueSequence: Removed, Removed

RotDescriptor: LargeGrain, SmallGrain, Meadow
RotFamilies: Graminae, Graminae, Meadow
TillDesc: Conservation, Conservation, Conservation
CropsPurposes: Silage, WinterCover, Meadow
FieldOrganicMatter: High
FieldSoilType: Clay
FieldCondition: Weedy
PriorPestHistory: None
RotationPests: SCMHigh, WWHigh, WGHigh, AWHigh, BCWHigh, ECBLow, RWLow, CSBHigh
PestsInfoSource: PMG/Interview; VICE-Corn

$NEWCASE$ MC3008

RotationSequence: CornG, WheatG, SoybeansDC, Rye
TillageSequence: NoTill, FallConventional, NoTill, MinimumTill
ResidueSequence: Left, Removed, Left, Left
RotDescriptor: LargeGrain, SmallGrain, Legume, SmallGrain
RotFamilies: Graminae, Graminae, Leguminae, Graminae
TillDesc: Conservation, Conventional, Conservation, Conservation
CropsPurposes: Grain, Grain, Beans, WinterCover
FieldOrganicMatter: Moderate
FieldSoilType: Mixtures
FieldCondition: Moderate
PriorPestHistory: None
RotationPests: WGHigh, AWHigh, SCMHigh, CSBModerate, RWLow, ECBHigh
PestsInfoSource: PMG/Interview; VICE-Corn

$NEWCASE$ MC3001

RotationSequence: CornG, WheatG, SoybeansDC, Rye
TillageSequence: NoTill, FallConventional, NoTill, MinimumTill
ResidueSequence: Left, Left, Left, Left
RotDescriptor: LargeGrain, SmallGrain, Legume, SmallGrain
RotFamilies: Graminae, Graminae, Leguminae, Graminae
TillDesc: Conservation, Conventional, Conservation, Conservation
CropsPurposes: Grain, Grain, Beans, WinterCover
FieldOrganicMatter: Moderate
FieldSoilType: Mixtures
FieldCondition: Weedy
PriorPestHistory: None
RotationPests: SCMHigh, AWHigh, WGHigh, MBBHigh, BLBModerate, RWLow, ECBHigh, CSBModerate
PestsInfoSource: PMG/Interviews; VICE-Corn
Other: Mexican Bean Beetle high late in the season on Soybeans. CLAHigh on Wheat

$NEWCASE$ MC3006

RotationSequence: CornG, WheatG, SoybeansDC, Rye
TillageSequence: NoTill, MinimumTill, NoTill, MinimumTill
ResidueSequence: Left, Removed, Left, Left
RotDescriptor: LargeGrain, SmallGrain, Legume, SmallGrain
RotFamilies: Graminae, Graminae, Leguminae, Graminae
TillDesc: Conservation, Conservation, Conservation, Conservation
CropsPurposes: Grain, Grain, Beans, WinterCover
FieldOrganicMatter: Moderate
FieldSoilType: Mixtures
FieldCondition: Moderate
PriorPestHistory: None
RotationPests: WGHigh, AWModerate, SCMModerate, MBBHigh, RWLow, CSBLow, ECBHigh
PestsInfoSource: PMG/Interviews; VICE-Corn

$NEWCASE$ MC3007

RotationSequence: CornG, WheatG, SoybeansDC, Rye
TillageSequence: NoTill, MinimumTill, NoTill, MinimumTill
ResidueSequence: Left, Left, Left, Left
RotDescriptor: LargeGrain, SmallGrain, Legume, SmallGrain
RotFamilies: Graminae, Graminae, Leguminae, Graminae
TillDesc: Conservation, Conservation, Conservation, Conservation
CropsPurposes: Grain, Grain, Beans, WinterCover
FieldOrganicMatter: Low
FieldSoilType: SiltLoams
FieldCondition: Clean
PriorPestHistory: None
RotationPests: AWHigh, RWLow, CSBLow, ECBHigh, MBBModerate, BLBLow, CRAModerate
PestsInfoSource: PMG/Interviews; VICE-Corn

$NEWCASE$ MC3005

RotationSequence: CornG, Rye, CornG, WheatG, SoybeansDC, Rye
TillageSequence: NoTill, MinimumTill, NoTill, FallConventional, NoTill, MinimumTill
ResidueSequence: Left, Left, Left, Removed, Left, Left
RotDescriptor: LargeGrain, SmallGrain, LargeGrain, SmallGrain, Legume, SmallGrain
RotFamilies: Graminae, Graminae, Graminae, Graminae, Leguminae, Graminae
TillDesc: Conservation, Conservation, Conservation, Conventional, Conservation, Conservation

CropsPurposes: Grain, WinterCover, Grain, Grain, Beans, WinterCover
FieldOrganicMatter: High
FieldSoilType: Clay
FieldCondition: Weedy
PriorPestHistory: CLA
RotationPests: RWModerate, ECBHigh, CSBHigh, CLBLow, MBBHigh, WGHigh, CEWModerate, SCMHigh, AWHigh, BLBHigh
PestsInfoSource: PMG/Interviews; VICE-Corn

$NEWCASE$ MC3002

RotationSequence: CornG, Rye, CornG, WheatG, SoybeansDC, Rye
TillageSequence: NoTill, MinimumTill, NoTill, FallConventional, NoTill, MinimumTill
ResidueSequence: Left, Left, Left, Left, Left, Left
RotDescriptor: LargeGrain, SmallGrain, LargeGrain, SmallGrain, Legume, SmallGrain
RotFamilies: Graminae, Graminae, Graminae, Graminae, Leguminae, Graminae
TillDesc: Conservation, Conservation, Conservation, Conventional, Conservation, Conservation
CropsPurposes: Grain, WinterCover, Grain, Grain, Beans, WinterCover
FieldOrganicMatterv Moderate
FieldSoilType: Mixtures
FieldCondition: Moderate
PriorPestHistory: BB, CEW
RotationPests: RWModerate, WGHigh, AWHigh, WWModerate, MBBHigh, BBHigh, CEWModerate, CLBLow, ECBModerate, CSBHigh
PestsInfoSourcev PMG/Interviews; VICE-Corn
Other: CornLeafAphid can be a problem on Wheat

$NEWCASE$ MC3004

RotationSequence: CornG, Rye, CornG, WheatG, SoybeansDC, Rye
TillageSequence: NoTill, MinimumTill, NoTill, MinimumTill, NoTill, MinimumTill
ResidueSequence: Left, Left, Left, Removed, Left, Left
RotDescriptor: LargeGrain, SmallGrain, LargeGrain, SmallGrain, Legume, SmallGrain
RotFamilies: Graminae, Graminae, Graminae, Graminae, Leguminae, Graminae
TillDesc: Conservation, Conservation, Conservation, Conservation, Conservation, Conservation
CropsPurposes: Grain, WinterCover, Grain, Grain, Beans, WinterCover
FieldOrganicMatter: Moderate
FieldSoilType: Mixtures
FieldCondition: Moderate
PriorPestHistory: None
RotationPests: CEWModerate, MBBModerate, BLBModerate, CLBLow, CSBHigh, RWHigh, CRAHigh, ECBHigh
PestsInfoSource: PMG/Interviews; VICE-Corn

$NEWCASE$ MC3003

RotationSequence: CornG, Rye, CornG, WheatG, SoybeansDC, Rye
TillageSequence: NoTill, MinimumTill, NoTill, MinimumTill, NoTill, MinimumTill
ResidueSequence: Left, Left, Left, Left, Left, Left
RotDescriptor: LargeGrain, SmallGrain, LargeGrain, SmallGrain, Legume, SmallGrain
RotFamilies: Graminae, Graminae, Graminae, Graminae, Leguminae, Graminae
TillDesc: Conservation, Conservation, Conservation, Conservation, Conservation, Conservation
CropsPurposes: Grain, WinterCover, Grain, Grain, Beans, WinterCover
FieldOrganicMatter: High
FieldSoilType: Clay
FieldCondition: Moderate
PriorPestHistory: None
RotationPests: BLBHigh, MBBModerate, WWLow, CEWModerate, CLBModerate, RWModerate, ECBHigh, CSBHigh
PestsInfoSource: PMG/Interviews; VICE-Corn
Other: Potential for CornLeafAphidon Wheat. CLB Primarily affects late wheat

$NEWCASE$ MC3009

RotationSequence: CornG, WheatG, Meadow
TillageSequence: NoTill, MinimumTill, MinimumTill
ResidueSequence: Left, Removed
RotDescriptor: LargeGrain, SmallGrain, Meadow
RotFamilies: Graminae, Graminae, Meadow
TillDesc: Conservation, Conservation, Conservation
CropsPurposes: Grain, Grain, Meadow
FieldOrganicMatter: Low
FieldSoilType: Mixtures
FieldCondition: Moderate
PriorPestHistory: None
RotationPests: HFLow, ChBHigh, SCMHigh, WWHigh, WGHigh, AWModerate, RWLow, ECBHigh
PestsInfoSource: PMG/Interviews; VICE-Corn
Other: Late planted wheat has low Hessian fly damage

$NEWCASE$ MC3010

RotationSequence: CornG, WheatG, Meadow
TillageSequence: NoTill, MinimumTill, MinimumTill
ResidueSequence: Left, Left
RotDescriptor: LargeGrain, SmallGrain, Meadow

RotFamilies: Graminae, Graminae, Meadow
TillDesc: Conservation, Conservation, Conservation
CropsPurposes: Grain, Grain, Meadow
FieldOrganicMatter: Moderate
FieldSoilType: Clay
FieldCondition: Weedy
PriorPestHistory: None
RotationPests: AWHigh, ChBHigh, BCWHigh, SCMHigh, WGHigh, WWModerate, RWLow, CSBHigh, ECBHigh
PestsInfoSource: PMG/Interviews; VICE-Corn

$NEWCASE$ MC3011

RotationSequence: CornG, WheatG, Meadow
TillageSequence: NoTill, MinimumTill, NoTill
ResidueSequence: Left, Removed
RotDescriptor: LargeGrain, SmallGrain, Meadow
RotFamilies: Graminae, Graminae, Meadow
TillDesc: Conservation, Conservation, Conservation
CropsPurposes: Grain, Grain, Meadow
FieldOrganicMatter: Moderate
FieldSoilType: Mixtures
FieldCondition: Moderate
PriorPestHistory: None
RotationPests: SCMHigh, WWHigh, WGHigh, AWHigh, CLBLow, CRAModerate, RWLow, ECBhigh, CSBModerate
PestsInfoSource: PMG/Interviews; VICE-Corn

$NEWCASE$ MC3012

RotationSequence: CornG, WheatG, Meadow
TillageSequence: NoTill, MinimumTill, NoTill
ResidueSequence: Left, Left
RotDescriptor: LargeGrain, SmallGrain, Meadow
RotFamilies: Graminae, Graminae, Meadow
TillDesc: Conservation, Conservation, Conservation
CropsPurposes: Grain, Grain, Meadow
FieldOrganicMatter: Low
FieldSoilType: Clay
FieldCondition: Weedy
PriorPestHistory: None
RotationPests: SCMModerate, WGHigh, AWHigh, BCWHigh, SCMLow, RWLow, ECBHigh
PestsInfoSource: PMG/Interviews; VICE-Corn.

Prabhakar Bhogaraju
Department of Entomology, VPI & SU
Blacksburg, VA 24061
(540) 231-9116
prabhakar@vt.edu

Education:

B.S. Agriculture -- August 1991. Agricultural College, Andhra Pradesh Agricultural University, A.P., India. OGPA: 8.93/10.00.

Recent Employment:

- Graduate Research Assistant -- Department of Entomology, VPI & SU -- August 1993 - May 1996.
- Graduate Research Assistant -- Department of Agronomy and Soils, Clemson University -- January 1993 - July 1993.

Presentations:

- Prabhakar Bhogaraju and Nicholas D. Stone. 1996. A Case for Case-Based Reasoning in Whole-Farm Planning. Poster presented at the Graduate Research Symposium, VPI & SU, Blacksburg, VA. USA.
- Prabhakar Bhogaraju and Nicholas D. Stone. 1996 SeePest: A Case-Based Reasoner for IPM decision support in whole-farm planning.Poster presented at the Entomological Society of America (ESA) -- Eastern Branch Meeting, Williamsburg, VA. USA.
- Prabhakar Bhogaraju and Nicholas D. Stone. 1996. C-Chest: A knowledge-based system for decision support in farm planning. Poster presented at the AAAS Annual Meeting and Science Innovation Exposition (AMSIE'96). Baltimore, MD.

Professional Affiliations:

- Member American Association for Artificial Intelligence (AAAI), 1994-Present.
- Member Entomological Society of America (ESA), 1994-Present.

Honors/Awards:

- Graduate Research Fellowship, Summer 1996. Waste Policy Institute, Blacksburg, VA.
- Display presentation received honorable mention at the AAAS Annual Meeting and Science Innovation Exposition (AMSIE'96), Baltimore, MD. in Feb. 1996.