

Exploring Pest Management Practices and Development of Knowledge Base System for Pepper Disease Diagnosis

By

Bethlehem Asferi

**A thesis submitted to Addis Ababa University, the Faculty of Informatics, in
partial fulfillment of the requirements for the degree of Master of Science in
Information Science**

August, 2010

Acknowledgements

Any piece of written work requires commitment, long hours and lots of thinking which can be painful at times. Sometimes, this task is just as difficult for those loved ones and family, who share this journey perhaps unexpectedly, but with the same wishes and expectations that the work will one day be concluded.

I thank Jehovah God for giving me insight where and when I needed it, and for giving me the courage to finish what I started.

I would like to say thank you to my supervisor, Ato Tibebe Beshah for his guidance. He has been willing to pay out his time to help. I also would like to thank the experts who were willing to share their knowledge and guidance during the domain knowledge acquisition task. Especially Dr. Tameru Alemu, I would like to say thank you!

I would also like to thank my family for keeping on helping me.

Last but not least gratitude goes to my dear husband whose love and understanding kept me going through out my school time. He has never complained when I had to spend more time on this research and always offered words of encouragement when I was feeling down.

Thank you Darling!

Bethlehem Asferi.

Abstract

This paper presents survey results of farmers' current pest management practice and knowledge and associated problems. It presents a knowledge base system in the area of agriculture and describes the design and development of the rule based expert system using prolog shell. The designed system is intended for the diagnosis of common diseases occurring in pepper plant. A knowledge base system is a computer program composed of a knowledge base, an inference engine and a user interface. The proposed knowledge base system has a user interface and provides diagnosis knowledge on the basis of response(s) of the user made against the queries related to particular disease symptoms. The system integrates a structured knowledge base that contains knowledge about symptoms and remedies of diseases in the pepper plant appearing during their life span. Survey results of current pest management practice and associated problems are included in the research to show the importance of developing a knowledge base system. The system has been tested with domain dataset, and results given by the system have been validated with domain experts.

TABLE OF CONTENTS

| | |
|---|-----------|
| LIST OF TABLES..... | 7 |
| LIST OF FIGURES..... | 7 |
| ACRONYMS..... | 8 |
| CHAPTER ONE: INTRODUCTION..... | 9 |
| 1.1 BACKGROUND..... | 9 |
| 1.2 STATEMENT OF THE PROBLEM | 11 |
| 1.3 OBJECTIVE..... | 12 |
| 1.3.1 <i>General objective</i> | 12 |
| 1.3.2 <i>Specific Objective</i> | 12 |
| 1.4 SIGNIFICANCE AND BENEFICIARIES OF THE RESEARCH..... | 13 |
| 1.5 SCOPE AND LIMITATION | 14 |
| CHAPTER TWO: PEPPER AND ITS DISEASES IN ETHIOPIA | 15 |
| 2.1 INTRODUCTION | 15 |
| DEFINITION | 15 |
| 2.2 HISTORY OF PEPPER | 15 |
| 2.3 PEPPER IN ETHIOPIA..... | 16 |
| 2.4 ECONOMIC IMPORTANCE OF PEPPER PESTS | 18 |
| 2.4.1 <i>Bacterial soft rot</i> | 19 |
| 2.4.2 <i>Powdery mildew</i> | 19 |
| 2.4.3 <i>Bacterial spot</i> | 19 |
| 2.4.4 <i>Red spider mites</i> | 20 |
| 2.4.5 <i>Aphid</i> | 20 |
| 2.4.6 <i>Flea beetle</i> | 20 |
| CHAPTER THREE: KNOWLEDGE BASE SYSTEM | 22 |
| 3.1 INTRODUCTION | 22 |
| 3.2 DEFINITION | 22 |
| 3.3 PERSPECTIVES OF KNOWLEDGE..... | 23 |
| 3.4 KNOWLEDGE BASE SYSTEM (KBS) | 24 |
| 3.4.1 <i>History of Knowledge Base System</i> | 24 |
| EARLY, SIGNIFICANT EXPERT SYSTEMS | 25 |
| 3.4.2 <i>Components of a Knowledge Base System</i> | 27 |
| 3.4.2.1 <i>The Knowledge base</i> | 27 |
| 3.4.2.2 <i>The Inference Engine</i> | 28 |
| 3.4.2.3 <i>The explanation facility</i> | 31 |
| 3.4.2.4 <i>The user Interface</i> | 31 |
| 3.4.3 <i>Knowledge Base System Development Procedures</i> | 32 |
| 3.4.3.1 <i>Knowledge acquisition</i> | 33 |
| 3.4.3.2 <i>Knowledge representation</i> | 33 |

| | |
|--|-----------|
| 3.4.3.3 Implementation..... | 36 |
| 3.4.3.4 System evaluation | 37 |
| 3.4.4 <i>Knowledge Base System Application Areas</i> | 37 |
| 3.4.4.1 Knowledge base systems in agriculture | 37 |
| 3.4.4.2 Knowledge based system applications in plant disease diagnosis | 38 |
| CHAPTER FOUR: DESIGN AND METHODOLOGY OF THE STUDY | 41 |
| 4.1 INTRODUCTION | 41 |
| 4.2 PRELIMINARY ASSESSMENT | 41 |
| 4.2.1 <i>General approach</i> | 41 |
| 4.2.2 <i>Sampling Technique</i> | 42 |
| 4.2.3 <i>Data collection tool and procedure</i> | 42 |
| 4.2.4 <i>Data Analysis and Interpretation</i> | 44 |
| 4.3 KNOWLEDGE ENGINEERING METHODOLOGY | 44 |
| 4.3.1 <i>Knowledge Acquisition</i> | 45 |
| 4.3.2 <i>Knowledge Verification</i> | 47 |
| 4.3.2.1 Multiple Experts Conflict Resolution | 47 |
| 4.3.2.2 Review Procedure | 48 |
| 4.3.3 <i>Knowledge Representation</i> | 48 |
| 4.4 SOFTWARE ENGINEERING METHODOLOGY | 49 |
| 4.4.1 <i>Requirements Specification</i> | 49 |
| 4.4.2 <i>Design</i> | 49 |
| 4.4.2.1 Knowledge Representation | 49 |
| 4.4.2.2 Interfaces..... | 50 |
| 4.4.2.3 Explanation Module | 50 |
| 4.4.3 <i>Implementation tools & process</i> | 50 |
| 4.4.4 <i>System Testing</i> | 51 |
| 4.4.4.1 Verification | 51 |
| 4.4.4.2 Validation | 51 |
| CHAPTER FIVE: DATA ANALYSIS AND KNOWLEDGE BASE SYSTEM DEVELOPMENT | 53 |
| 5.1 INTRODUCTION | 53 |
| 5.2 PRELIMINARY ASSESSMENT ANALYSIS | 53 |
| <i>Interpretation of survey results</i> | 56 |
| 5.3 KBDIAP: A KNOWLEDGE BASE SYSTEM TO DIAGNOSE PEPPER DISEASE | 57 |
| 5.3.1 <i>Knowledge Acquisition</i> | 57 |
| 5.3.2 <i>Knowledge Model</i> | 57 |
| Semantic Networks | 58 |
| 5.3.3 <i>Knowledge Representation</i> | 60 |
| Production rules: IF_THEN representation of the Knowledge Base System..... | 63 |
| 5.3.4 <i>Implementation</i> | 69 |
| 5.3.5 <i>System Evaluation</i> | 70 |
| 5.3.5.1 Verification | 70 |
| 5.3.5.2 Validation | 70 |

| | |
|---|-----------|
| CHAPTER SIX: CONCLUSIONS AND RECOMMENDATIONS | 72 |
| 6.1 CONCLUSIONS..... | 72 |
| 6.2 RECOMMENDATIONS..... | 73 |
| REFERENCES..... | 74 |
| APPENDIX I: QUESTIONNAIRE..... | 79 |
| APPENDIX II: QUESTIONS PRESENTED TO DOMAIN EXPERTS | 82 |
| APPENDIX III: TEST CASES | 83 |
| APPENDIX IV : SAMPLE CODE FROM THE EXPERT SYSTEM SHELL AND THE RULE BASE | 99 |

List of Tables

| | |
|---|---------|
| Table 4.1 Experts profile----- | page 38 |
| Table 5.1 Frequency table----- | page 46 |
| Table 5.2 weighted mean and mode of the survey. ----- | page 47 |
| Table 5.3 Insect registration sheet----- | page 51 |
| Table 5.4 Symptoms registration sheet----- | page 53 |
| Table 5.5 Insect identification form----- | page 53 |
| Table 5.6 Disease diagnosis form----- | page 54 |
| Table 5.7 Disease diagnosis form for Bacterial spot----- | page 55 |
| Table 5.8 Disease diagnosis form for Powdery mildew----- | page 55 |
| Table 5.9 Disease identification form for Bacterial Soft Rot----- | page 56 |
| Table 5.10 Disease identification form for Stem rot----- | page 57 |
| Table 5.11 Insect identification form for Red spider mite----- | page 58 |
| Table 5.12 Insect identification form for African Bull Worm----- | page 58 |
| Table 5.13 Insect identification form for Aphid----- | page 59 |
| Table 5.14 Insect identification form for White fly----- | page 59 |
| Table 5.15 Insect identification form for Flea beetle----- | page 60 |
| Table 5.16 Insect identification form for Leaf miner----- | page 61 |
| Table 5.17 Disease identification form for Virus----- | page 62 |

List of Figures

| | |
|---|---------|
| Fig 3.1 Knowledge base system architecture----- | page 23 |
| Fig 4.1 Spiral Model for the Knowledge base System Development----- | page 36 |
| Fig 5.1 Semantic network for pepper and its diseases----- | page 50 |
| Fig 5.2 Semantic network for pepper and its diseases----- | page 57 |
| Fig.5.3 Semantic network for insects causing Pepper disease.----- | page 58 |

Acronyms

KBS = Knowledge base system

EARO = Ethiopian Agricultural Research Organization

SNNPR = Southern nation nationalities and people

EPMV = Ethiopian Pepper Mottle Virus

EEPA= Ethiopian Export Promotion Agency

ESEF= Ethiopian Spice Extraction Factory

GUI = Graphical User Interface

FOL = First Order Logic

DDT = DichloroDiphenylTrichloroethane

Chapter one

Introduction

1.1 Background

Crop protection research in Ethiopia dates back to the establishment of the then Imperial Ethiopian College of Agriculture and Mechanical Arts, now Alemaya University, in the late 1950s. The emphasis during that period was on survey and identification of arthropod pests associated with crop damage. The establishment of the Institute of Agricultural Research, now Ethiopian Agricultural Research Organization, during the second half of the 1960s saw a more focused and organized approach to research on general crop pest management aspects. Since then initiatives have been undertaken to establish the identity, geographic distribution and to some extent economic importance of pests. Studies have been conducted on cultural and chemical control practices, screening for host plant resistance, and establishing the identity and complex of natural enemies of major pests (Abate 1997a).

To date, a fair amount of knowledge exists on pest management of several crop pests in Ethiopia. But there is still a problem of improper pest management practice by Ethiopian farmers (Mohammed Y., et al, 2006).

Ethiopia has wide ranges of agro-ecological diversity and therefore produces wide ranges of crops. The areas in the north and north central parts of the country are dominated by cereal based farming systems whereas root crops based agriculture is most dominating in the south. Major crops include cereals (such as tef, maize, sorghum, barley, wheat, finger millet), roots and tubers (enset, sweet potatoes, potatoes), pulses (dry beans, faba bean, dry peas, grass pea, chickpeas, lentils), oilseeds (rapeseed, groundnuts, safflower, sesame, seed cotton, castor beans, linseed), vegetables (tomatoes, onions and shallots, pepper), fruits (bananas, citrus, pineapple, mangoes) and cash crops (coffee, tea, sugarcane, cotton, tobacco).

While cash crops such as coffee and tobacco have shown sustained yield increases over the years, it has not been possible to attain sustainable increased yield of food crops (FAO Agricultural Database 2002).

Yields of food crops per unit area of land have either declined or stayed almost stagnant over the years; by contrast, the overall cultivated area has shown substantial increases.

Pests and diseases, coupled with a low level of improved agricultural technology, recurrent droughts, and decreases in soil fertility levels are some of the major contributors to the low and unstable crop yields in Ethiopia (Tsedeke A, 2006). A report from Holetta Research Station in 1986 indicated that the average pre harvest loss for horticultural crops such as Root and Tubers, Vegetables and Fruits ranges between 13-29%. Pepper is among the vegetable crops produced in Ethiopia and its production has been hindered by recurrent pest incidence.

Pepper is produced in all the continents except Antarctica. In Antarctica there are stories about pepper being kept in flower pots to spice up their food (Boseland and Votava, 2000). It is believed to have originated in Central and South America. Peru and Mexico might have been the second centers of origin, after which it spread into the New World Tropics before its subsequent introduction into Asia and Africa in 1493 (Bosland and Votava, 2000). Tropical Asia like India, Malaysia, Thailand, Indonesia and Philippines, tropical Africa, South America and the Caribbean became the main producers over time.

Ethiopian Economy is mainly based on agriculture. Major crops produced in the country include cereals, roots and tubers, pulses, oilseeds, vegetables like tomatoes, onions, shallots and pepper, fruits and cash crops. The focus of this research is Pepper.

Portuguese had probably introduced hot pepper to the country in early 17th century (Huffnagel, 1961). It has since been grown as important spice and vegetable everywhere in the country. Nowadays pepper is produced all over Ethiopia but districts in the rift valley in the south are the major production segments (Tameru A., 2004). The pepper production area in the country is estimated to be about 340,000 ha (FAOstat, 2002). Today the crop not only attained economical, but also traditional importance. The powder from dried ripe fruits of hot pepper is used as spice to flavor 'wot', an Ethiopian stew in a daily traditional meal. Besides, the crop is exported as dried ripe fruit or as oleoresin (Yoseph et al., 1989). In addition, it serves as income generating crop for small scale farmers.

1.2 Statement of the problem

The steady decline in price of other Ethiopian main cash crops such as coffee requires diversification of agricultural export commodities. In this regard, pepper could play important role, if constraints are identified and solved. Lesson could be learnt from China and India. Today over 48% of the world pepper is produced in Asia, China being the leading country. The production in China alone exceeds the entire production of European countries (Rubatzky and Yamguchi, 1997). India is the major exporter of dry peppers, followed by China, and the major importing countries are the U.S.A. and Germany (Berke, 2002).

However, frequent and severe outbreaks of diseases in pepper are challenging the production of the crop. Pepper constantly shows severe disease symptoms throughout Ethiopia and in the rift valley districts particularly. State farms in the past and recently private investors have often banned the production of the crop because of unacceptably high loss due to disease (Tameru A., 2004). Thus yield and production areas of pepper remained low in Ethiopia over the last ten years (FAOstat, 2002). This has resulted in the inflationary increase in price of the crop and the appearance of forged pepper powder products in local markets that were harmful to human health.

To reduce yield loss and to combat pest problems, vegetable growers in the central rift valley region of Ethiopia mainly use pesticides. However a study done by Tadesse Amare and Asferachew Abate shows that there is irrational use of pesticides among the farmers (Tadesse A. et.al., 2008). On their report the researchers indicated that the farmers contacted for the study had a practice of using pesticides without adequate knowledge of the pesticide i.e. it's dosage and whether it is obsolete or not. The study showed the existence of farmers who even use internationally banned pesticides like DDT (Tadesse A. et.al. 2008). Such use of pesticides has aggravated the problem of pests through resistance development, development of new pests and declining of natural controlling factors such as parasitoids. Even though there are also some non chemical options available for pest control, they have not been used by growers due to poor extension and farmers linkage (Mohammed Y, et. al., 2006).

Experiences have showed that knowledge base systems can be used as a tool to disseminate the available best ways to combat pest problems. For example Dr. Wheat, A Web-based Expert System for Diagnosis of Diseases and Pests in Pakistani Wheat was successfully developed by

Fahad Shahbaz Khan and his friends (Fahad K. et. al., 2008). The researchers indicated that the system was developed so that it can act as a powerful tool with extensive potential in wheat disease diagnosis where agricultural specialist assistance is not readily available. CUPTEX is another expert system developed for handling management of cucumber disorders and it was successful (Rafea et al, 1995). Yialouris and Sideridis (1996) also developed an expert system for tomato. The system was designed to address tomato disease identification problem.

Hence, when successfully built, knowledge base systems can help non experts to make decisions by providing valuable information in an understandable and easily accessible form (Knight and Mumford 1994). Pest management experts are often in great demand but there is small number of them to reach everywhere they are required. But the use of diagnosing knowledge base system allows their expertise to be available as necessary.

The present research intends the assessment of existing pest management practices and problems and the development of a knowledge base system for pepper disease diagnosis so that disease occurrences in pepper can be diagnosed by the knowledge base system with accuracy comparable to human experts. By combining expert knowledge such a system would improve access to timely and comprehensive pest management knowledge.

1.3 Objective

1.3.1 General objective

The general objective of this research is to explore the pest management practice and associated problems and to come up with a knowledge base prototype system that identifies and diagnoses pepper diseases.

1.3.2 Specific Objective

The specific objectives of the study are:

- To assess existing problem in relation to pest management practice
- To extract expert knowledge on diseases, symptoms and diagnosis
- To build the knowledge base and inference rules
- To test and evaluate the system
- To forward recommendation for future work

1.4 Significance and beneficiaries of the research

Due to frequent occurrence of pepper disease, yield loss occurs on the vegetable. A report revealed that an estimated 13-29% yield loss was registered on vegetable crops like pepper due to diseases (Tadesse A., et al, 2008). Farmers usually use chemical pesticides to control diseases. But researches showed that there is improper use of pesticides by growers due to lack of knowledge on pest identification and use of the correct pesticide and dosage. Non pesticidal control options such as biological, cultural and varietal control methods have been developed in research centers. For example, efforts have been made at the Melkassa Research Center of EARO (Ethiopian Agricultural Research Organization) to develop non pesticidal control for the management of major pests and some promising results have been reported (Tesfay and Habtu, 1985; Tsedeke and gashawbeza, 1994; Tsedeke and Gashawbeza, 1997). But research results have not reached farmers because of a poor research extension system (Mohammed Y, et. al., 2006). Due to this poor linkage between farmers and experts, pepper production in the country remained low. The present research is conducted to provide a way to disperse expert knowledge on pepper disease diagnosis where experts are not available. Hence the research is carried out to:

- Benefit farmers by making easy and timely access to disease identification and diagnosis knowledge. Access to the right information will help the farmers to effectively combat pest and yield loss due to diseases will be reduced and this in turn increases their income. Non experts can help farmers to identify and diagnose pepper diseases.
- Increase yield production throughout the country and increase Ethiopia's share in the global pepper market.
- To help non experts in disease identification and diagnosis work. Non experts can improve their knowledge and be effective in pepper disease identification and diagnosis work as expert knowledge is readily available.
- Knowledge base system professionals can use the system as a base to conduct further research in the area. Agriculture is the base on which Ethiopia's economy is built. Loss of yield means a lot to the country. But as a matter of fact, significant yield loss occurs due to crop diseases and lack of knowledge by farmers on effective ways to combat the diseases. Hence Information science researchers can use this system as a reference to build similar works on other crops.

- The researcher benefited from this research by gaining hands on experience in knowledge base systems development. Besides the researcher gained experience in research planning and design, conducting survey and analyzing and interpretation of survey data.

1.5 Scope and limitation

The current research addressed pepper diseases in Ethiopia. 11 pests, i.e. 5 diseases caused by Fungi/bacteria/virus and 6 insects that attack pepper were considered. Symptoms shown on leaf, stem and pod are considered to identify pepper diseases. For the insects that attack pepper, body structure of the insect, color of the insect and feeding habit were considered and treatments suggested by experts were included in the research. Site condition i.e. soil condition, rain, humidity, temperature, can also be an indicator to identify a disease or an insect. But it was not considered in this research for the reason of easy management. Besides, pepper weed diagnosis and post-harvest pest management is not included again for similar reason. Location and season of growth as well are not included in this research because of inaccessibility of complete data.

Chapter Two

Pepper and Its Diseases in Ethiopia

2.1 Introduction

In this chapter researches done on pepper (*Capsicum* spp.) disease worldwide and in the country are revised. Our country, Ethiopia is one of the leading countries that have been producing this vegetable for many years. However the production of pepper in Ethiopia has been generally low over the last several years mainly due to severe infections (FAO, 2002). Pepper threats that have been identified over the years are discussed in this section. Economic importance of pepper to Ethiopia and damage caused by infecting organisms is explained.

Definition

Pests are organisms that interfere with human interest and leading to certain level of losses. Weeds, insects and diseases are considered as pests if they result in 5-10% yield loss (Aberra D., 2006). In this thesis the term pepper pest is used to refer pepper disease and/or insects that attack pepper. However, weed is not the considered in this research.

2.2 History of Pepper

History points to South America as the earliest known home of pepper. Pottery found in Inca graves—believed to be thousands of years old—is decorated with pictures of the pepper plant. Peppers are even among the foods that were buried with Inca mummies (Awake, May, 2000). Some historians believe that a doctor who sailed to America with Columbus carried pepper seeds to Spain in 1493. From there the pepper plant made its way to Britain and to southern France, where it was more popular as a houseplant than as a seasoning or a source of food. Later, Greek tradesmen took pepper plants to the countries around the Mediterranean and the Black Sea. (Awake, May, 2000.)

Today, pepper is produced in all the continents except Antarctica. In Antarctica there are stories about pepper being kept in flower pots to spice up their food (Boseland and Votava, 2000).

Chili peppers are important in almost every Asian country. They are the number one vegetable in Malaysia and Bhutan, and rank at or near to the top in terms of growing area in most Asian countries (Berke, 2002).

Pepper is the world's second important vegetable ranking after tomatoes and it is the most produced type of spice flavoring and color to food while providing essential vitamins and minerals. The nutritional value of hot pepper merits special attention. It is a rich source of vitamin A and E. Both hot and sweet peppers contain five to six times as much vitamin C as an orange or a lemon and this fact makes pepper ideal to prevent flu colds more than any other vegetable crop (Boselad and Votava, 2000). The color and flavor extracts from pepper are used in both the food and feed industries, e.g., ginger beer, hot sauces and poultry feed. In some countries, the shoot tips are cooked as herb or as vegetable (Rubatzky and Yomaguchi, 1997).

In many households, pepper provides the only needed flavor to enhance intake of otherwise bland diets. The range of food products that contain pepper or its chemical constituent is broad, and it includes ethnic foods, meat, salad dressings, mayonnaise, dairy products, and candies, packed foods, snack foods, salsa, and hot sauces. Rubatzky and Yomaguchi (1997) pointed out in addition to their uses as food, uses for cosmetic production, condiment and medicine.

2.3 Pepper in Ethiopia

The history of pepper in Ethiopia is perhaps more ancient than the history of any other vegetable product (EEPA, 2003). Ethiopians have strong attachment to dark red pepper, which has high value principally for its high pungency. The fine powdered pungent product is an indispensable flavoring and coloring ingredient in the common traditional sauce “*Wot*” whereas; the green pod is consumed as a vegetable with other food items. There is a general belief among Ethiopians that a person who frequently consumes hot pepper has resistance to various diseases. Pepper is in the daily diet of most Ethiopians. The average daily consumption of hot pepper by Ethiopian adult is estimated 15 gram, which is higher than tomatoes and most other vegetables (MARC, 2004).

Production of pepper is well known in Ethiopia. Pepper is an ‘annual’ plant which grows at altitude ranging from 1400 up to 2100 meter above sea level (m.a.s.l.).

Pepper plants thrive in flat fields of brown, sandy soil bathed in plenty of sunlight and about 600-650 mm rainfall.

Some pepper plants are sown from seed; others are transplanted. Pepper plants for transplanting need to be started off in a controlled environment, such as a well-ventilated greenhouse or nursery. There the tiny plants are carefully watered, fertilized, and weeded. Before being transplanted, they are gradually exposed to outdoor conditions. Seedlings are raised starting April and transplanted as the main rainy seasons begins, which is June/July. Depending on the area, harvesting starts 4 to 5 months from transplanting. The seedlings are transplanted 40–50 days after planting. Planting is carried out in the beginning of the main rain season (Roukens, 2005).

Red pepper is the leading vegetable and spice grown in the country. The central (eastern and southern Shoa), western, north western (Wollega, Gojjam) and the southern part of the country are the potential pepper producing areas. Currently most of the produce comes from Alaba, Meskanina Mareko and Siltie zone (CSA, 2003). Birr Sheleko and Didessa valleies also produce a good amount of it (MARC, 2004).

Even though Ethiopia has the potential to produce pepper, yield and production area remained declining. The major factor for this is plant infection by harmful organisms. For example, production of the crop was banned in some years by farmers due to unacceptably high proportion of viral infection (Tameru unpublished data). Due to this there is an increase of the price of the crop at local markets.

In addition to having major role in Ethiopians daily dish it also plays an important role in the national economy. It is an important cash crop today; on average 79% of pepper production is for market in SNNPR (CSA, 2003). It is a crop of high value in both domestic and export markets. Since it is a commercial and industrial crop, it generates employment to urban and rural workers. Oleoresin (coloring) and capsaicin (hot) are extracted from red pepper for export purpose. The deep red colored and large podded cultivars (sweet/hot) have a very high processing demand in the country. The main processed product, oleoresin, is exported to different countries and the spiced ground is supplied to local market.

From 1992/93 to 2003/04, a total of 616.16 tones of oleoresin, which worth 106.6 million Birr, was exported to different countries by Ethiopian Spices Extracting Factory (ESEF, 2005).

2.4 Economic importance of pepper pests

Even though pepper is an important crop worldwide, its production has been hindered by an attack from different kind of pests. For example in India it was reported that in Wyanad and Idukki districts of Kerala and Kodagu district of Karnataka, up to 80% of crop loss has been recorded on a few severely infested plantations (Davis H., 2005). In Carolina, a research showed that southern stem blight produces sclerotia that are small round structures able to persist in the soil for years. Heavily infested fields of this particular disease alone can result in substantial crop losses of 80% or more (Frank J. Louws et. al., 2005). America estimated total losses including cost of control and damage for pepper weevil alone in 1997 was \$1.4 million, cost of control at \$201,000 and damage at \$1,236,000 (Frank J. Louws et. al., 2005).

In Ethiopia, viral disease incidences of more than 90% and complete crop failure were reported in some localities (Agranovsky, 1993; Tameru et al., 2003). In order to determine and verify the identity of viruses involved, 400 samples collected from framers field in 2001/2001 cropping season were subjected to different virus identification methods on the research conducted by Tameru in 2003 (Tameru A., 2003). The result revealed that 60% and 10% of the samples were infected by EPMV alone and in mixed infection with the other potyviruses respectively. Interviews with farmers during the research indicated a lack of knowledge concerning the cause of their crop losses. The researcher revealed that trained extension workers in the field of plant virology who can help farmers in virus diagnosis are lacking. Combinations of these factors have contributed to the increasing importance of the virus. The researcher indicated the severity of the situation and it is time to make farmers aware of their problem and to devise suitable management measures against the virus. (Tameru A., et.al., 2003)

In addition to viral infections

- Bacterial Spot /xanthomonas campestris/

- Powdery Mildew /*Leveillula taurica*/
- Bacterial soft rot /*Erwinia carotovora*/
- Stem rot /*Sclerotinia sclerotiorum*/

are active enemies of the crop in the country (Tameru et al., 2003). Brief description of some of the insects and diseases of pepper is given below.

2.4.1 Bacterial soft rot

Initial symptoms of bacterial soft rot often appear in pepper leaves, which show dark necrotic spot. As the disease progresses, pod decays containing water often resulting in the pod fall. The affected plants wilt and die later.

2.4.2 Powdery mildew

Powdery mildew of pepper is caused by *Leveillula taurica*. *Leveillula taurica* is a species complex that infects over 1000 plant species in 74 families, including tomato and eggplant as well as pepper (Margaret Tuttle McGrath, 2001). Pepper powdery mildew is different in several ways from the mildews that infect tomato (*Erysiphe*, *Oidium lycopersicum*), or cucumber, (*Erysiphe cichoracearum*, *Sphaerotheca fuliginea*). Pepper powdery mildew grows unseen, within the leaf tissue for a latency period of up to 21 days. Unlike tomato and cucumber powdery mildew which is easily seen on the top side of the leaves, pepper powdery mildew grows on the underside of leaves. In general, pepper crops become more susceptible to this powdery mildew as they mature. Severely infected leaves wither and drop off causing plants to die.

2.4.3 Bacterial spot

Bacterial spot of peppers can be recognized by numerous spots on the leaves. Initially, the spots are watersoaked and infected pepper leaves drop prematurely.

The disease is caused by the bacterium *Xanthomonas campestris* pv. *vesicatoria* (proposed names, *X. vesicatoria* and *X. axonopodis* pv. *vesicatoria*). The bacteria are microscopic and occur in enormous numbers in affected areas. They are rod-shaped and have a long whip-like tail that propels them in water; this helps them invade wet leaves and cause infection (David F. Ritchie, 1996).

2.4.4 Red spider mites

Spider mites are so small that they can hardly be seen without a magnifying glass, though they can be seen as tiny moving specks with the naked eye when shaken onto a piece of white paper. They are usually round to nearly round in outline and have eight legs with the body thinly clothed with long, rigid hairs. Red spiders live on the underside of the leaves where they suck the sap from the host plant. Red spider infestations are favored by warm dry weather. As a control mechanism, a thorough spraying with water through a nozzle will help control the mites. Wet weather also favors their natural enemies. However, when insect build up is sufficient to cause damage, use Isotox, Malathion, or Dragon Pyrethrin Concentrate. These should be mixed and applied as directed on the label. Plants should be thoroughly covered with the spray mixture with special care to get the underside of the leaves. It is advised in severe cases a second spraying in 7 to 10 days.

2.4.5 Aphid

Several aphid species may be commonly found infesting peppers during most of the growing season. Large numbers of aphids can affect pepper production in two ways. Honeydew produced by aphids can leave a sticky film on the surface of the fruit and cause the development of sooty mold fungi. Various species of aphids can also transmit viruses that can reduce yields. In Ethiopia Aphids were identified to transmit Ethiopian Pepper Mottle Virus (EPMV) (Tameru A. et.al, 2009).

Natural enemies such as lady beetles, green lacewings, damsel bugs, and hover fly larvae usually control aphid populations adequately. Broad spectrum insecticides, particularly pyrethroid insecticides, can delete these natural enemies and allow aphid populations to develop unchecked. Insecticides should only be applied for other insects when necessary, as determined by trap catches and scouting, and care should be taken to select insecticides that do not favor secondary aphid problems.

2.4.6 Flea beetle

Flea beetles are small, hard shelled insects, so named because their enlarged hind legs allow them to jump like fleas from plants when disturbed. They usually move by walking or flying, but when alarmed they can jump a considerable distance. Most adult flea beetle damage is unique in

appearance. They feed by chewing a small hole often smaller than 1/8 inch in a leaf, moving a short distance, then chewing another hole and so on. The result looks like a number of “shot holes” in the leaf. Most plants can tolerate considerable flea beetle feeding. However, younger plants could be killed, and older ones may suffer from water loss and reduced photosynthetic capability.

Botanical pesticides recommended for controlling flea beetles include neem, rotenone, pyrethrin, sabadilla, and formulations of these in some combination. Sprays combining rotenone with insecticidal soap are considered very effective. Other sources suggest that garlic sprays are useful.

Chapter Three

Knowledge Base System

3.1 Introduction

In this chapter definition for knowledge and knowledge base systems is provided. History of knowledge base systems is also presented. Building blocks of a knowledge base system i.e. the knowledge base, the inference engine, the explanation facility and the user interface are explained. Application of knowledge base systems for agriculture in general and for pepper disease diagnosis in particular is reviewed.

3.2 Definition

Data can be described as being unstructured facts (Avison and Fitzgerald, 1998). Further, it can also be said that data can be a specific, discrete and / or finite quantity which can describe the specific state or being of something. For example, it can be said that the data which describes the temperature on a given day, is the specific reading on a thermometer, say 15 degrees celcius. Information, on the other hand, is the interpretation of data. This is in the sense that information is a refinement on the context of a set of data, which as a whole implies some specific meaning. Hence, a collection of temperatures can be classified as being climate related information. That is, certain data which is related to each other within a specific context and for a particular meaning.

In its simplest sense, knowledge is the natural outcome of understanding and using information within a particular context. Since knowledge is based upon the refinement of both of these concepts, there can exist many definitions of knowledge. Probst, Raub and Romhardt (1994), gave one such definition of as:

'...the whole body of cognitions and skills that individuals use in order to solve problems...'
(Probst et al. 1994, pp.24)

In this case, the authors view knowledge as pertaining specifically to decision making tasks which require the applicable usage of context specific information. Another view of knowledge was given by Davenport and Prusak (1998), who suggest that knowledge is more of a collection of experiences and values, which provides the individual or organization with the ability to evaluate and incorporate new ideas and information (Davenport and Prusak, 1998). Indeed, Polanyi has famously stated that knowledge is such a thing, that it is impossible to define fully, as ‘we know more than we can tell’ (Polanyi, 1966). Polanyi recognized that the tacit dimension forms an indispensable part of human knowledge, although we might not always be aware of having this knowledge. Explicit knowledge implies knowledge that is already extracted from human and is recorded on books, literature etc. Later in their book on knowledge creating companies, Nonaka and Takeuchi explain the different kinds of knowledge by talking about tacit and explicit knowledge.(Nonaka and Takeuchi, 1995).

Hence a Knowledge base system can be defined as a large database of known theoretical knowledge (‘Deep’ knowledge) or acquired / heuristic knowledge (‘Surface’ knowledge). Expert knowledge bases can be combined together in order to apply different heuristics (intuitive rules of thumb) to solve difficult problems (Amir S., 2004).

3.3 Perspectives of Knowledge

One method of providing a view on the context of knowledge is through a research area which centers on the notion of mapping knowledge in some data centric or computable (algorithmic) form. Of the two approaches some researchers such as Galliers and Newell (2000) have even suggested that knowledge in itself should only be relevant where the data that defines it is relevant and accurate. A much less radical approach in this light is instead to focus on how knowledge can be structured and represented, in terms of language and logic. This field is known more generally as Knowledge Representation and is an important subfield of Artificial Intelligence research. In John Sowa’s words, knowledge representation applies theories and techniques from the fields of logic, ontology and computation in order to represent something in the real world (Sowa, 2000). In this sense, the representational view of knowledge is purely structural, knowledge which is embodied via the use of semantic and logical propositions (Davis, Shrobe and Szolovits, 1993).

3.4 Knowledge base system (KBS)

A knowledge base system can be defined as a computer program that attempts to replicate the reasoning processes of a human expert (Shu-Hsien 2005). It can make decisions and recommendations and perform tasks based on user input. The expert's knowledge can be available when the human expert might not be and so that the knowledge can be available at all times and in many places, as necessary. Knowledge base systems derive their input for decision making from prompts at the user interface, or from data files stored in the computer.

3.4.1 History of Knowledge Base System

Knowledge based systems are one of the two major paradigms for developing intelligent systems within the field of Artificial Intelligence. Expert/knowledge-based systems are an example of the symbolic paradigm; the other major paradigm is the connectionist paradigm that has led to the development of Neural Network technology. Expert/knowledge-based systems were the first major successful application technology to evolve from Artificial Intelligence research.

The foundations of the field of Artificial Intelligence can be traced from many different disciplines including philosophy, mathematics, psychology, computer engineering, and linguistics (S.J. Russell and P. Norvig, 1995).

The first cited work in the area of Artificial Intelligence dates back to McCulloch and Pitts in 1943. They proposed a model of artificial neurons that mimic the structure of the human brain; this area later became the connectionist paradigm.

In the summer of 1956, John McCarthy organized a two month workshop at Dartmouth and ten leading U.S. researchers interested in automata theory, neural networks, and the study of intelligence were invited. Two researchers from Carnegie Tech, Allen Newell and Herbert Simon were the focus of the workshop due to their reasoning program known as the Logic Theorist (LT). Simon claimed "we have invented a computer program capable of thinking non numerically, and thereby solved the venerable mind-body problem." The Dartmouth workshop accomplished two major outcomes. First, it served as a forum to introduce the leading researchers to each other; for the next twenty years, the field of AI would be dominated by these ten individuals, their students, and colleagues at MIT, CMU, Stanford, and IBM. The second

major accomplishment of the workshop—and a more lasting one—was an agreement to adopt John McCarthy’s new name for the field: Artificial Intelligence.

The work of Newell and Simon is the first documented work using the symbolic programming paradigm of AI. Their work on LT led them to develop another program known as general Problem Solver (GPS). The success of GPS was not as widely heralded however because of the limited class of problems that it could solve. GPS was designed from the start to imitate human problem solving protocols regardless of the information contained in the domain. These so called “weak” methods, because they use weak information about the domain, turned out to show weak performance in solving problems in more complex domains.

Researchers then took the opposite approach in the development of the DENDRAL program (R.K. Lindsay et.al., 1980). They applied the knowledge of analytical chemists to infer the molecular structure from the information provided by a mass spectrometer. DENDRAL holds a significant place in the history of knowledge based systems because it was the first system to use the expertise of human problem solvers and translate that knowledge into a large numbers of special purpose rules, known as a rule based system.

Early, Significant Expert Systems

The work on DENDRAL lead to many others successful applications of this new technology known as expert systems. Feigenbaum and others at Stanford began the Heuristic Programming Project (HPP) to investigate other problem domains that could benefit from this new technology. The next major effort was in the area of medical diagnosis. Bruce Buchanan and Dr. Edward Shortliffe developed MYCIN to diagnose blood infections (B.G. Buchanan and E.H. Shortliffe, 1985). Using about 450 rules, MYCIN was able to perform as well as some experts, and considerably better than some junior doctors were. MYCIN is one of the most widely known of all knowledge base system applications developed. However, MYCIN is significant to the history of knowledge-based systems for two particular reasons.

First, unlike DENDRAL, which used a model of a particular molecule as the basis for its reasoning, MYCIN was constructed from interviews with various doctors in the particular domain. Therefore, MYCIN contains a number of heuristic rules that are used by physicians in

the identification of certain infections. The second major contribution of MYCIN was the later development of EMYCIN (Empty MYCIN). EMYCIN was the first expert/knowledge-based system shell. It took approximately 20 years to develop the MYCIN program. The researchers realized that if expert systems were to become a viable problem solving technique this development time must be cut. In an effort to do reduce the time to develop an expert system the researchers developed EMYCIN by taking all of the rules out of the system and leaving just an empty “shell” in which other developers in other domains could then just “plug-in” their new knowledge base.

There were other significant knowledge base system applications that were also developed in the early days of expert systems. These systems include PUFF, which used EMYCIN in the domain of pulmonary disorders, DELTA/CATS, which was developed at General Electric Company to assist railroad personnel in the maintenance of GE’s diesel electric locomotives (J.P. Ignizio, 1991). Also at this time, researchers at CMU developed the first truly successful commercial application of expert systems. The system, developed for Digital Equipment Corporation (DEC), was used for computer configuration and known as XCON (R1). XCON, originally titled R1, was developed by John McDermott at CMU for aiding in the configuration of VAX and PDP-11 computer systems at DEC. There exist an enormous number of configurations for VAX and PDP-11 computer system. DEC attempts to configure each system to meet specific customer needs. XCON was originally developed as a 500-rule prototype that examined the specific needs of the customer and decided the exact configuration of components necessary to meet the customer requirements. In particular, XCON’s function was to select and arrange the components of a computer systems including: the CPU, the memory, the terminals, the tape and disk drives, and any other peripherals attached to the system. XCON works with a large database of computer components and its rules determine what makes a complete order.

The development effort for XCON began in 1978 and by September 1979, XCON was able to configure more than 75 percent of all customer orders that it was given. By 1981, XCON was being used by DEC on a regular basis and DEC estimates that its cost savings in 1983, 1984, and 1985 were a combined \$83 million. Today, XCON is still being used by DEC to configure all VAX orders. There is a development team dedicated to keeping the rules in XCON current and keeping the users of XCON trained on the latest updates. A new copy of XCON is released

practically every 3 months and the latest version handles nearly 12,000 different computer components that could possibly be configured into a customer order (D.N. Chorafas, 1992). XCON was one of the major, early success stories in the field of expert systems, for its high visibility domain, its continued use and expansion, and its tremendous impact on the bottom line i.e. profit.

3.4.2 Components of a Knowledge Base System

A knowledge base system is constructed from a knowledge base that contains rules, an inference engine where the inferencing technique is defined, an explanation facility that gives explanation on the activities of the system to the users as necessary and a user interface through which the user interacts to the system. Fig 3.1 shows the interaction of these components.

3.4.2.1 The Knowledge base

The knowledge base is the most important component of an expert system as reflected in the saying “Usually, the most powerful expert systems are those containing the most knowledge.” The knowledge base of expert systems contains both factual and heuristic knowledge (Robert S. Englemore, 1993). As described by Robert S. Englemore (1993) factual knowledge is that knowledge of the task domain that is widely shared, typically found in textbooks or journals, and commonly agreed upon by those knowledgeable in the particular field. Whereas heuristic knowledge is the less rigorous, more experiential, more judgmental knowledge of performance. In contrast to factual knowledge, heuristic knowledge is rarely discussed, and is largely individualistic. It is the knowledge of good practice, good judgment, and plausible reasoning in the field. It is the knowledge that underlies the "art of good guessing." Knowledge representation techniques are used in knowledge base system development process to formalize and organize the knowledge (Robert S. Englemore, et.al. 1993).

Amir Sherif (2004) indicated four major ways to represent knowledge in the knowledge base. The first technique is by means of formulas in first order predicate calculus. This involves representing declarative facts as instantiated predicates. The second approach to representing knowledge is the use of frames as developed by Minsky in 1975. Frames are data structures in which all knowledge about a particular object is stored together. Each frame contains a number of labeled slots in which the slots may be blank, or specified by terminals referring to other frames. This allows the knowledge to be organized for modularity and accessibility. A special

kind of frame is a “script” or “case” which allows various scenes to be represented in a frame like orientation.

Closely aligned with the frames concept is the use of semantic or associative networks. Under this method, knowledge is organized around the objects being described, but here the objects are represented by nodes in a graph, and the relations among them are represented by labeled arcs. The main advantage of semantic networks over logical representation like predicate calculus is that, for each object, all the relevant information is collected together. The last major technique of representing knowledge in the knowledge base is the use of production rules or simply rules. A rule consists of an IF part and a THEN part. The IF part lists sets of conditions in logical combination. If the IF part of the rule is satisfied consequently the THEN part can be concluded, or its problem solving action will be taken. Expert systems whose knowledge is represented in rule form are called rule based systems.

3.4.2.2 The Inference Engine

The inference engine uses the knowledge provided to come to some conclusions and/or give advice about the specific problem (Plant and Stone, 1991). To arrive at conclusions about a problem, the inference engine must search for a solution in an efficient and effective manner. Any search will be guided by a variety of data and constraints. The search strategy used can be forward chaining, backward chaining or both of them. The search strategy that begins with data and constraints and uses them to filter a large selection down to a few choices is called forward chaining (Ignizio 1991). Here the inference engine examines a set of rules, each assertion being evaluated to determine its truth. For those rules that evaluate to true, their conclusions or consequents are added to the knowledge base. This process continues until no further consequents can be determined. Forward chaining allows the knowledge engineer to use rules to develop information from a limited set of initial data.

To see how forward chaining works, consider a system with three rules:

- 1 If someone is a third year, then they need a job.
- 2 If someone is a third year, then they live in.
- 3 If someone needs a job, they will become an accountant.

Suppose we put the following fact:

John is a third year.

Being a forward chainer, the system is constantly on watch for new data. As soon as this data arrives, the system searches all the rules for any whose conditions weren't true before but are now. It then adds their conclusions.

In this case, rules 1 and 2 have conditions which match this new fact. So the system will immediately create and add the two facts below.

John needs a job.
John lives in.

These facts in turn can trigger rules. As each arrives, the system would look for yet more rules that are made true. In this case, the fact `John needs a job` would trigger rule 3, resulting in the addition of another fact:

John will become an accountant.
The fact `John lives in` would not trigger anything else though.

This type of reasoning is appropriate, for example, in a monitoring situation where it is desirable to learn as much as possible about the state of a monitored system based upon the available data (Walters and Nielsen 1988).

Another search strategy called backward chaining starts with a goal and works backward to check data and constraints to determine if the goal is feasible (Ignizio, 1991). In backward chaining, the inference engine identifies one or more hypotheses and begins searching for rules that contain the hypothesis as a consequent (i.e., concluding that the hypothesis is correct). For any such rule found, the inference engine tests the truth of the predicates i.e. the if clauses of the rule. If the predicates are true, then the hypothesis is confirmed, and the inference engine moves on to the next hypothesis. If the truth of a predicate is unknown, the hypothesis that the unknown predicate is true is added to the inference engine's list of hypotheses to check. This initiates a search for rules with the new hypotheses as a consequent. This process forms a chain, linking rule predicates backward to consequents. Backward chaining is 'goal-driven'. The system starts with a *conclusion* which the engine tries to satisfy. If this conclusion cannot be satisfied the engine searches for *sub goals*, conclusions that will help satisfy a part of the current goal. It continues this process until either the initial conclusion is satisfied or there are no more unsatisfied sub goals.

So, if we had this knowledge base again

1 If someone is a third year, then they need a job.
2 If someone is a third year, then they live in.
3 If someone needs a job, they will apply to be an accountant.
and we were to add
John is a third year
the system would do nothing at all.

But if we were then to ask the question

Is there anyone who is going to become an accountant?
the system would try to answer. It would begin by searching either for a fact that gives the answer directly, or for a rule by which the answer could be inferred. To find such a rule, it searches the entire knowledge base for rules whose conclusions, if made true, will answer the question.

In this example, there are no facts giving the answer; there's one rule whose conclusion, if true, would supply an answer, and that's rule 3.

The system next checks the rule's conditions. Is there anyone who needs a job? As with the original question, we look either for a fact that answers directly, or for a rule. There are no facts, but rule 1 is relevant.

So we now check *its* conditions. Is there a third year? This time, there is a fact that answers this: John is a third year. So we've proved rule 1, and that's proved rule 3, and that's answered the question.

This strategy is often used in selection, classification and diagnostic applications in which one item is to be selected from a fixed set of items (Walters and Nielsen 1988).

Both strategies will ultimately lead to a conclusion, but the efficiency of the search is dependent on the nature of the problem faced. A problem with few premises and many conclusions would generally be better off with a forward chaining strategy whereas a problem with many premises and few conclusions would normally do better with a backward chaining strategy. There are instances in which it would be wise to employ a combination of the two strategies (Ignizio 1991).

In this research back ward chaining is the strategy the inference engine employs as this strategy is appropriate for diagnostic applications where there are large number of premises and few conclusions.

3.4.2.3 The explanation facility

This part of the knowledge base system provides explanation on the reasoning the inference engine used when the engine forwards a certain recommendation. It gives justification for the system's conclusions and explanations of why the system needs a particular piece of data (Amir S., 2004).

3.4.2.4 The user Interface

The input/output interface defines the way in which the expert system interacts with the user and other systems such as databases. Interfaces are usually graphical with screen displays, windowing, and mouse control. They receive input from the user and display output to the user (Nikolopoulos 1997). Some systems use natural language front ends that accept English like responses but most use a graphical user interface (GUI) with a mouse device to allow the user to choose from selections in dialog boxes and menu bars (Schmoldt and Rauscher 1996).

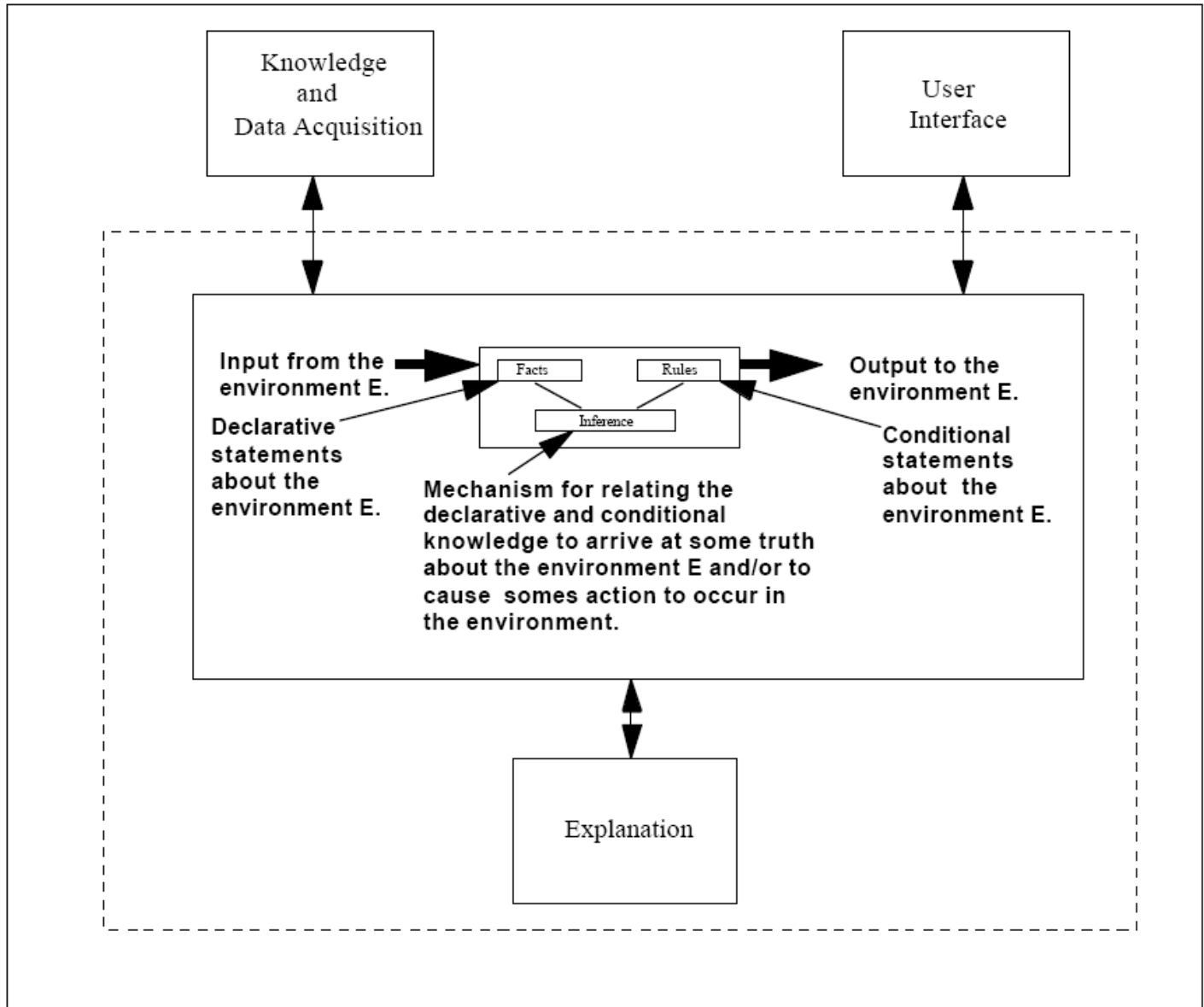


Fig 3.1 Knowledge base system architecture (James J. et.al., 1999)

3.4.3 Knowledge Base System Development Procedures

The development of knowledge base systems passes through different procedures. These are knowledge acquisition, knowledge representation, implementation and testing. Explanation of each procedure is given in the following sections.

3.4.3.1 Knowledge acquisition

Knowledge acquisition is the process of collecting the knowledge necessary for solving a problem and encoding it into a form that allows for efficient computer manipulation. This task is usually take place throughout the development life cycle as deficiencies in the knowledge base are realized and modifications are made. In knowledge elicitation, domain knowledge is obtained through various means including interviews with experts and book and journal references. (Nikolopoulos 1997). Methods of collecting, organizing, and formalizing knowledge are many and vary widely depending on the source. When knowledge is extracted from human specialists, the acquisition process is often called knowledge elicitation. The job of knowledge elicitation from human experts can be very difficult due to the inexplicit nature of human knowledge (Schmoldt and Rauscher 1996). There is no universal agreement on which knowledge elicitation technique to use when. It is most common to start with interviews and then use other methods when considered useful. The knowledge engineer must be versatile and willing to weigh the various methods in order to please the experts and elicit the most information (Foley and Hart 1992).

3.4.3.2 Knowledge representation

The ultimate goal of any knowledge representation is to allow information to be efficiently structured, modified, and reasoned with. As the basis of knowledge representation is from a purely computational basis, it is therefore fitting to view this approach as being based upon a series of structural components. That is, a series of models and concepts which require knowledge to be abstracted in a particular manner. Some of the better known approaches in this light include production rules, semantic networks, frame systems, predicate logic and the use of formal ontologies.

Although different knowledge representation methodologies exist, rule based KR is the most commonly used methodology in agricultural knowledge base systems (Kramers et al., 1998).

a. Production rules

In a rule base knowledge based system the knowledge of the domain is represented by production rules (Hayes Roth F., 1985). The rule base is typically populated with rules of the following form:

$$A \text{ ----> } B$$

This is interpreted as “if condition A is satisfied then do B”. The “A” portion of the rule is called the antecedent or LHS (Left Hand Side) of the rule. The “B” portion of the rule is called the consequent or RHS (Right Hand Side) of the rule. If A is true and whatever actions specified in B are accomplished then the rule is said to have been “fired”. The condition “A” may be a conjunction of conditions A1, A2, ..., An which must all be satisfied in order to trigger any actions stipulated by B. Any component of this conjunction may involve a negative. Likewise “B” may be a sequence of actions B1, B2, ..., Bk all of which will be taken if the conditional part of the rule is satisfied and the rule is fired.

The relationship between the rule base and the fact base is quite straightforward. If there is a fact in the fact base like “Var1 = n” and there is a rule in the rule base that states that “If Var1 = n then B” then this rule is considered for execution or firing.

b. Semantic Networks and Frame Systems

Briefly, a semantic net, is a graphical method of representing real world concepts via nodes in a directed graph (Quillian, 1967). Knowledge and meaning between concepts, is implied through the interconnection between each concept. By reading this directed graph, a language or semantic structure of the knowledge can be formed and hence can be abstracted through a computer language. Such methods have been used successfully to model knowledge which is well defined, as in classification problems, and in applications such as in medical prognosis (Genesereth and Nilsson, 1987).

Frame systems were introduced by Minsky (1975), as a means to structuralize a semantic network in order to describe specific instances of an occurrence. Here, a frame is a named piece of data, which exhibits particular attributes known as slots. Due to the fact that each frame has

certain properties, more complex knowledge structures can be inferred by artificially replicating and inheriting semantic node attributes.

As such, frames are typically used as a method for reasoning with a given amount of knowledge. A more rigorous approach to formalizing knowledge is through the use of First order logic (FOL), or predicate logic / calculus. In this method, a series of logical assertions are made about each component of knowledge, from which an overall bounding set of knowledge can be inferred.

c. First-Order Logic

First order logic (FOL), also known as predicate calculus or predicate logic is a well understood formalism for reasoning. Although the logic and knowledge representation communities are distinct, the expressivity of FOL nevertheless makes it a powerful knowledge representation language. From the perspective of FOL, the world consists of objects and the relations that hold between them. A FOL language consists of logical and non logical symbols. The logical symbols represent quantification, implication, conjunction and disjunction; while the non logical symbols are constants, predicates, functions, and variables. Constant, variable and function symbols are used to build terms, which can be combined with predicates to construct formulas (Amir S., 2004).

FOL is an extremely expressive representation, and can be used to describe semantic networks and frame systems.

d. Ontology

In order for information from different sources to be integrated, there needs to be a shared understanding of the relevant domain. Knowledge representation formalisms provide structures for organizing this knowledge, but provide no mechanisms for sharing it. Ontologies provide a common vocabulary to support the sharing and reuse of knowledge. As discussed by Guarino and Garetta (1995), the meaning of the term ontology is often vague. It was first used to describe the philosophical study of the nature and organization of reality. In AI, the most cited definition is due to Tom Gruber (1999):

“An ontology is an explicit specification of a conceptualization.”

In this definition, a conceptualization is an abstract view of the world. Ontology associates vocabulary terms with entities identified in the conceptualization and provides definitions to constrain the interpretations of these terms. In a later paper, Guarino refines this model and provides the following definition for an ontology (2002). An ontology is a logical theory accounting for the intended meaning of a formal vocabulary, i.e., its ontological commitment to a particular conceptualization of the world. The intended models of a logical language using such a vocabulary are constrained by its ontological commitment.

Most researchers agree that an ontology must include a vocabulary and corresponding definitions, but there is no consensus on a more detailed characterization. Typically, the vocabulary includes terms for classes and relations, while the definitions of these terms may be informal text, or may be specified using a formal language like predicate logic. The advantage of formal definitions is that they allow a machine to perform much deeper reasoning; the disadvantage is that these definitions are much more difficult to construct. Numerous ontologies have been constructed, with varying scopes, levels of detail, and viewpoints. Noy and Hafner (1997) provide a good overview and comparison of some of these projects. One of the more prominent themes in ontology research is the construction of reusable components. The advantages of such components are clear: large ontologies can be quickly constructed by assembling and refining existing components, and integration of ontologies is easier when the ontologies share components.

One of the most common ways to achieve reusability is to allow the specification of an inclusion relation that states that one or more ontologies are included in the new theory. If these relationships are acyclic and treat all elements of the included ontology as if they were defined locally then an ontology can be said to extend its included ontologies.

3.4.3.3 Implementation

Compared to the wide variation in domain knowledge, only a small number of AI methods are known that are useful in expert systems. That is, currently there are only a handful of ways in which to represent knowledge, or to make inferences, or to generate explanations. Thus, systems

can be built that contain these useful methods without any domain specific knowledge. Such systems are known as skeletal systems, shells, or simply AI tools (Robert S. Englemore, 1993).

Building expert systems by using shells offers significant advantages. A system can be built to perform a unique task by entering into a shell all the necessary knowledge about a task domain. The inference engine that applies the knowledge to the task at hand is built into the shell. If the program is not very complicated and if an expert has had some training in the use of a shell, the expert can enter the knowledge himself (Robert S. Englemore, 1993).

Many commercial shells are available today, ranging in size from shells on PCs, to shells on workstations, to shells on large mainframe computers. They include JEFF, PESS and PyCLIPS.

3.4.3.4 System evaluation

System evaluation is a process of determining the quality of the knowledge base system and the advice it provides. A knowledge base system evaluation involves two processes: Verification and Validation.

Verification is the process of inspecting the system looking for any error and to see how well the system works as per the requirements specification. During validation the system will be evaluated whether it is useful or not. For this purpose test cases will be prepared and will be given to the system and to the experts who initially provided the knowledge. The answer from the system and the experts will be compared to see how accurate the system is (Amir S., 2004).

3.4.4 Knowledge Base System Application Areas

Over the years, there have been various application areas that have been successful for expert system development. Waterman, Durkin, and Liebowitz present representative problem areas where expert systems have been successfully built and these are Interpretation, Prediction and Diagnosis. (Waterman, 1986, Durkin,1995, Liebowitz 1991; 1994a; 1994b)

3.4.4.1 Knowledge base systems in agriculture

Although the technology of expert systems was developed during the 1960s the first expert systems in agriculture, PLANT/ds was only reported in 1983 (Michalski et al., 1983). PLANT/ds was developed for the diagnosis of soybean diseases in Illinois, USA.

Agriculture requires information and application of knowledge from different interacting fields of science and engineering to make a suitable decision-making that in turn depends on interplay of these data and knowledge. This needs agricultural specializations and technical awareness to help the farmers in decision making. Existence of agricultural specializations and full awareness with technological progress in a farmer is a very rare thing in our country. Human experts are not always available, may not be accessible to every farmer or if available consultation may be very expensive. The other complications are that the decisions in agriculture practice depend on large number of factors. Thus even for a human expert it becomes awkward to take all factors into consideration while making a decision. All such problems have resulted in the development and evolution of the concept of knowledge base systems (Blackmore, 1999; Maloy, 2005). The use of information technologies improved the knowledge base and increased the capacity to control the production practices which in turn reduces the threat and uncertainty, improved the efficiency of decision making and better recognized the variations in diverse influencing features thus depicting enhanced management policies for the farm (Blackmore, 2000). In addition, it is possible to store much of the information that an expert needs to make decisions and can make them on hand for others; therefore the notion of knowledge based agriculture has an adequate prospective to improve the agricultural production.

At the Egyptian Ministry of Agriculture and Land Reclamation, a series of knowledge base systems have been developed to assist agricultural managers with crop Management. One such system is for advising managers of cucumber Farming in plastic tunnels (Rafea et al., 1991). The knowledge base system and the corresponding methodology were developed via the assistance of The American University in Cairo.

3.4.4.2 Knowledge based system applications in plant disease diagnosis

A Diagnostic Advisory Rule based Knowledge Base System for Pest Management (DIARES) was developed by B.D. Mahaman and his friends in 2003. The study was based on pests encountered in Solanaceous crops in Greece. The development tool used to implement DIARES was EXSYS Professional, Ver. 5.1.0 Beta-W (EXSYS, 1992). This tool represents knowledge in rules of the form ‘‘IF THEN. . .ELSE’’. The knowledge of the various disorders was obtained from specific literature and symptom descriptions and the rules from domain experts, i.e.

entomologist and psychopathologists. Rule-based knowledge representation was the methodology they used. The user operates the system through screens of a graphical user interface (GUI). The screens include images object, push buttons, radio buttons, mouse regions, etc. Regarding the limitations they faced, the researchers indicated that their system does not incorporate high quality pictures which depict the earliest symptoms in order to perform early diagnosis and apply the appropriate measures of timely control. Besides their knowledge base system was restricted only to few vegetable crops. This particular research emphasized that plant's earliest symptoms should be given to the system in order to take measures for timely control.

Another knowledge base system was developed by S.S. Abu-Naser, K.A. Kashkash and Fayyad in 2008. The researchers were from Faculty of Engineering and Information Technology, Al-ahzar University, Gaza, Palestine. They developed a knowledge base system that diagnoses variety of diseases. Data and knowledge for this system were collected from different sources, primarily from agricultural engineers and farmers. As a secondary source they used specialized data bases and few electronic websites. The facts and rules were represented using CLIPS. CLIPS stands for C Language Integrated Production System (Giarratano, 2002). CLIPS is a forward reasoning and pattern matching expert system shell. It provides a good inference engine mechanism that automatically matches facts against patterns in the rules then determines which rules are applicable.

The researchers used two different ways to diagnose plant diseases. One is the description given by the concerned person about the symptom of a particular disease where the person provides comprehensive, accurate and step by step description to get a complete view of symptoms. In the second method used, one or more images of the symptoms stored in the knowledge base are used to compare the case under consideration. Visual basic was used to develop the graphical user interface. During the evaluation phase, a group of agricultural engineers, farmers and other concerned persons in plants were consulted regarding the two methods of diagnosis used in the knowledge base system. To a certain degree the majority of the group preferred the second method used, i.e. the graphical representation because it gives more accurate results; especially when it comes to similar symptoms between diseases.

But to the researcher's knowledge, no research has been done to the application of knowledge based systems to pest management in Ethiopia. Symptoms of diseases, disorders and pests have due geographical variations. So there is always a need to develop a new expert system for a different geographical region (Fahad Khan, et.al., 2008).

Chapter Four

Design and Methodology of the Study

4.1 Introduction

In this chapter the methodology used to carry out the research is discussed. Quantitative research methods were used to assess current practices and problems of pepper pest management. Both knowledge engineering methodologies and software engineering methodologies used are explained in detail. How the knowledge for the research is acquired, modeled and verified is provided in this section. The system development life cycle used in the research are explained. At last techniques employed for system evaluation are provided.

4.2 Preliminary assessment

4.2.1 General approach

The assessment was conducted in order to explore pest management practices of farmers, the level of perception of farmers on pest control mechanisms and to explore problems faced by farmers. Abela, a small village around Awassa, was chosen for the survey. The region was chosen because it is characterized by rugged mountains and flat landscapes and rain fed pepper and other vegetable crops are commonly produced. Selected participants answered a survey questionnaire structured in Likert format. Data gathered from this research instrument were then computed for interpretation. The descriptive method was used for this study. To define the descriptive type of research, Creswell (1994) stated that the descriptive method of research is to gather information about the present existing condition. The descriptive approach is quick and practical in terms of the financial aspect. Moreover, this method allows a flexible approach, thus, when important new issues and questions arise during the duration of the study, further investigation may be conducted.

4.2.2 Sampling Technique

In order to find out current pest management perception and practice of pepper growing farmers, a total of 60 individuals were included in the survey. The number of farmers to be included in the study (participants) was determined using single population proportion formula.

$$n = (Z\alpha/2)^2 p (1-p)/d^2$$

Where, $(Z\alpha/2)$ = Reliability coefficient = 1.96

n = Sample size

p = 4% this is because similar studies were difficult to find and taking the assumption that 96% of the farmers had low level of knowledge, attitude and practice regarding chemical pesticides.

d = assumed marginal error (5%)

$$n = (1.96)^2 (0.04) (0.96) / (0.05)^2 = 60$$

Hence, 60 individuals were contacted for the survey. Simple random sampling was used to select the individuals. Samples were taken along main and rural roads. To achieve pertinent information, certain inclusion criteria were imposed. The participants qualified for sample selection must be farmers who grow pepper. The farmers were contacted on April 2010.

4.2.3 Data collection tool and procedure

Primary data that were derived from the answers the participants gave during the survey process were used in this survey. Quantitative data collection methods are centered on the quantification of relationships between variables. Quantitative data gathering instruments establish relationship between measured variables. When these methods are used, the researcher is usually detached from the study and the final output is context free. Measurement, numerical data and statistics are the main substance of quantitative instruments. With these instruments, an explicit description of data collection and analysis of procedures are necessary. The quantitative approach is more on the detailed description of a phenomenon using a statement of statistical probability. Quantitative approach is useful as it helps the researcher to prevent bias in gathering and presenting research data. The quantitative data gathering methods are useful especially when a study needs to measure the cause and effect relationships evident between pre selected and discrete variables. The purpose of the quantitative approach is to avoid subjectivity by means of

collecting and exploring information which describes the experience being studied (Creswell, 1994).

The survey questionnaire was used as data gathering instrument for this study (See Appendix I). In general the questions used were designed to assess the current pest management knowledge and practice by farmers and to assess associated problems. Hence, the questionnaire was designed to measure the degree of the farmers' use of pesticides, ability to identify disease, if they know the dose of pesticides they are using, if they got enough consultation on pest management and on the effectiveness of current pest management practice. The questions were structured using five scale Likert format. In this survey type, five choices were provided for every question or statement. The choices represent the degree of agreement each respondent has on the given question. Below is the scale used and its interpretation

| Scale | Interpretation |
|-------|-------------------|
| 5.00 | Strongly Agree |
| 4.00 | Agree |
| 3.00 | Undecided |
| 2.00 | Disagree |
| 1.00 | Strongly disagree |

The Likert type survey was selected as it enabled the respondents to answer the survey easily. In addition, this research instrument allows the researcher to carry out the quantitative approach effectively with the use of statistics for data interpretation. In order to test the validity of the questionnaire used for the study, pilot testing was used and the researcher presented the questionnaire to five farmers. These respondents as well as their answers were not part of the actual study process and were only used for testing purposes. The researcher revised the survey questionnaire based on the suggestion of the respondents. The researcher then excluded irrelevant questions and changed vague or difficult terminologies into simpler ones in order to ensure comprehension.

4.2.4 Data Analysis and Interpretation

The filled questionnaires were collected and total responses for each item were obtained and tabulated. The data collected was analyzed using weighted mean and mode. Weighted mean is the average wherein every quantity to be averaged has a corresponding weight. These weights represent the significance of each quantity to the average. To compute for the weighted mean, each value must be multiplied by its weight. Products should then be added to obtain the total value. The total weight should also be computed by adding all the weights. The total value is then divided by the total number of respondents. The analyzed data was presented as a weighted mean. Mode is also used to find out the opinion of the majority. The parameters were taken from the questions presented to the interviewees. Hence, the analyzed data were used to measure the degree of pesticide use, disease identification knowledge, disease control knowledge, adequacy of expert advice and effectiveness of current disease management practice.

The weighted mean and mode of the analyzed data were interpreted as follows: weighted mean value equal to or greater than 3 shows the respondents' agreement with the statement mentioned on the questionnaire where as below three indicates their dissatisfaction. The mode value shows which measure of satisfaction value is chosen by most of the respondents i.e whether most of the respondents agree, strongly agree, strongly disagree, or disagree on the issue.

4.3 Knowledge Engineering Methodology

Following the preliminary survey, knowledge base system development tasks were accomplished. Knowledge engineering and system engineering methodologies were used to develop the system. To accomplish the knowledge engineering task three main activities were done. These are knowledge acquisition/elicitation, knowledge verification & knowledge modeling and representation. These tasks were done in iterations to produce successive versions of the expert system. In fact knowledge engineering tasks were input for system engineering tasks. Hence, at the end of every knowledge engineering cycle, begins system engineering cycle. The system engineering cycle, incorporates requirement specification, design, implementation and verification & validation. The output from knowledge acquisition process serves as system requirement specification throughout the system development process. Based on the system requirement specification system design was developed. The system design incorporated

knowledge modeling and representation, interface and explanation module design. In the implementation stage, the designed system was translated in to prolog code for system implementation. Prototypes were developed progressively as new knowledge was acquired from the domain experts. The final version of the system was tested using test cases to validate the accuracy of the system.

The figure given below best describes how these two cycles were carried out simultaneously and iteratively.

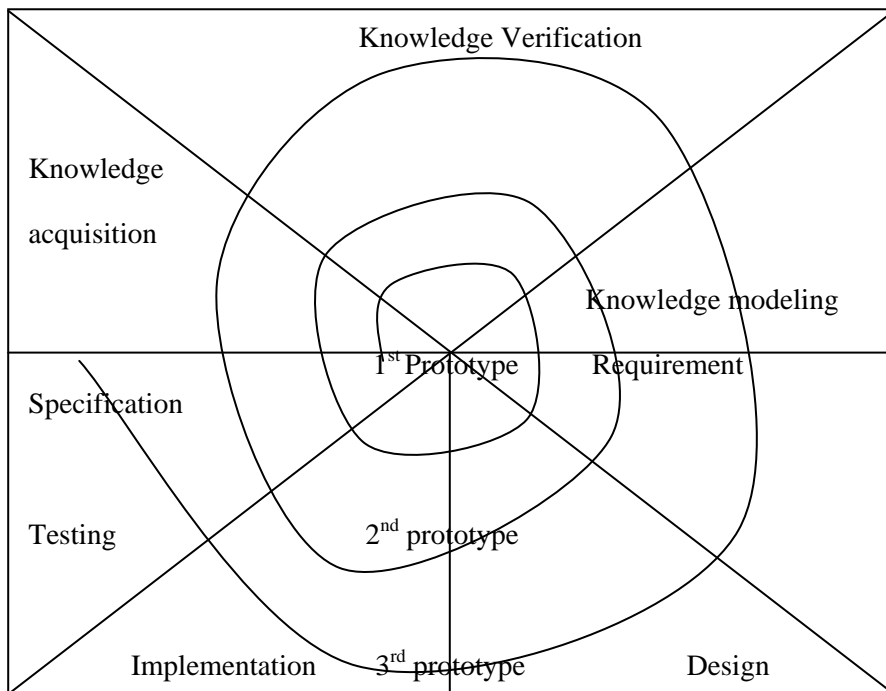


Fig 4.1 Spiral Model for the Knowledge base System Development (Yasser A. et.al, 1993).

4.3.1 Knowledge Acquisition

Knowledge acquisition is considered the bottleneck of the knowledge base system building process. One of the major difficulties at this stage is to explicitly identify and capture knowledge relevant to the intended application.

Experts primarily and literature secondly were consulted to acquire knowledge. Domain knowledge was obtained through individual interviews with experts in plant pathology, reviews of published research and pest management guides. Three experts from Hawassa University were selected and interviewed. Rudimentary knowledge of the domain was gained by consulting literatures. Based on this knowledge, questions were prepared to interview domain experts in the first meeting. Knowledge acquisition task was done iteratively throughout the system development process. During the iterative knowledge acquisition process it became apparent that changes to the initial system were needed and so two prototypes and a final system evolved as new knowledge was obtained.

In the first meeting with the experts, the objective was to communicate the purpose of this research and begin the knowledge acquisition process with interview to elicit some basic information. Questions were prepared in advance and notes were taken at the meeting. The experts were asked to explain the identified pepper pests in the country their symptoms that and available treatments. Part of the knowledge acquisition process was capturing procedural information, information about the sequences of inferences or actions taken by an expert. The second meeting with the experts was designed to review the rule representation of knowledge obtained in the first meeting and to elicit changes and additions to this knowledge. Follow up interviews were conducted as needed at irregular intervals throughout the lifecycle of the project.

| Education | Gender | Position | Experience | Remark |
|-----------|--------|----------|------------|---|
| Msc | Female | Lecturer | 2 years | Conducted masters research on pepper mottle virus |
| Ph.D. | Male | Lecturer | 12 years | Conducted both masters and doctorate researches on paper disease. |
| Ph.D. | Male | Lecturer | 17 years | Plant pathologist |

Table 4.1 Experts profile

A pepper disease control guide prepared by Melkasa and Wondo Genet Agricultural centers was used to get high level brief on the existing diseases. Articles published by experts in the area were consulted to get a detailed description of diseases and symptoms. Symptoms describe the injury resulting from the unknown agent. In addition thesis and articles prepared by Tameru Alemu were consulted (Tameru A, 2004, 2008, 2009).

On the knowledge acquisition process it was identified that symptoms can appear on different parts of the plant. A stem, leaf or the entire plant could be infected; and the effect on a particular part of the plant may differ for each pest. The effect could be the appearance of a spot, a scarce, rot or some kind of color. Signs are specific evidence that indicate the presence of a particular pest. Because the literatures were very descriptive and exhaustive, it proved a good source to begin enumerating the factors for each of the pests. Experts were then presented with the lists and their comments were solicited. The experts provided additional factors that were not identified in the literature indicated which factors were more important, and eliminated some that were erroneous.

4.3.2 Knowledge Verification

Knowledge verification is the stage whereby we make quality assurance of the acquired knowledge (Yasser A., 1993). Actually there are two ways to do this: one is Multiple expert conflict resolving procedure and the other is Review procedure.

4.3.2.1 Multiple Experts Conflict Resolution

Multiple experts conflict resolution is considered as a way of verifying the acquired knowledge. Because when two experts give different knowledge for the same thing, then trying to resolve this conflict yields more reliable knowledge, hopefully, agreed upon by both of them. If no consensus is reached the expert who is recognized to be more specialized in the area of disagreement is considered. But the experts were unable to meet at one time due to their schedule so review procedure was used to verify the knowledge.

4.3.2.2 Review Procedure

Establishing a review procedure at the knowledge acquisition stage reduces the efforts to be done later in the verification and validation of the developed system. Knowledge was reviewed at the end of different phases: knowledge elicitation, knowledge modeling and representation and implementation.

Reviewing at the elicitation stage was conducted by letting the domain experts review the results of the knowledge elicitation sessions. At the modeling and representation stage, the domain experts reviewed the represented knowledge. Since task and inference knowledge are documented in knowledge base representation notation, which was hard to be understood by non specialists, the knowledge engineer performed this activity by walking through them with the presence of the domain experts.

Reviewing at the implementation stage was conducted by letting the domain experts review any early prototype.

4.3.3 Knowledge Representation

The model of knowledge produced as an output of the knowledge acquisition activities is used as a generic representation of knowledge. Production rules, semantic networks, frames, first order logic and ontologies are among the available knowledge representation techniques. The knowledge representation technique employed in this research was production rule i.e. the knowledge was represented using IF-THEN statements. This is because rule based representations are best for agricultural knowledge base systems (Kramers et al., 1998). Semantic networks were used to map the generic model of the knowledge and this knowledge was further represented using production rules.

Following the first meeting, the elicited knowledge was converted to a rule based structure. The elicited knowledge was converted to rules in the form of IF-THEN statements where satisfying the conditions of the IF clause allows the inference of conclusions in the THEN clause.

After the initial set of rules was constructed, experts were asked to assign confidence values to each rule to indicate how strongly they felt the premises imply the conclusion. Each rule was considered independently, and the weight assigned was dependent only on that rule's antecedent conditions. Rules that the experts did not consider important were removed. Some new rules were also added at this time.

4.4 Software engineering methodology

As a software, building knowledge based system entails doing software system development activities that are accomplished in parallel with knowledge engineering activities. These activities go through different stages, these stages are: requirements specification, design, implementation, and testing. These activities were done in successive iterations, each of which ends with the delivery of a new, more mature version of the Knowledge base system.

The approach applied in this research was rapid prototyping. Rapid prototyping was used first to reach an agreement on the initial set of the system requirements. As the knowledge acquisition continues, advanced version of the system will be developed.

4.4.1 Requirements Specification

The outcome of the early knowledge elicitation activity was an initial set of requirements specifications. During knowledge elicitation activity domain experts were interviewed individually on identified pepper pests in the country, their symptoms and treatments. In the first meeting with the experts, research objectives were communicated and the knowledge acquisition process begun with unstructured interview to elicit basic knowledge. This knowledge served as the basis to develop the initial requirement specification, based on which the first prototype was developed. In the second meeting, structured interview was used to capture procedural knowledge i.e. knowledge about the sequences of actions taken by experts. This knowledge served as requirement specification to develop the second prototype. At the end of the second prototype implementation and testing, the requirements specification document was revised again. Additions, deletions and changes were made based on the domain experts' suggestion.

4.4.2 Design

A preliminary design was done just after the set of initial requirements specification was determined. This design was the basis for the initial prototype system. As new knowledge had been acquired through the knowledge acquisition task, the design document was revised accordingly. The following were the main subjects that were considered in the design document.

4.4.2.1 Knowledge Representation

Both semantic networks and production rules were used in the research to represent the acquired knowledge. Semantic networks were developed based on the initial requirement specification

document. Then production rules were used to represent details of the knowledge. These representations were rewritten so that to incorporate changes made on the initial requirement specification. Based on the revised representation was used to develop the second prototype. Hence the model of knowledge produced as an output of the knowledge engineering activities was used as a generic representation of knowledge in both the first prototype design and the second prototype design documents. The final design document was used to develop the system.

4.4.2.2 Interfaces

A command driven dialogue built using prolog was used as user interface for the user to interact with the system. As more knowledge was introduced to the system, information included in the dialogue had grown from prototype to prototype. In the first prototype, small numbers of questions were forwarded by the system to prompt the user about disease symptoms. As the acquired knowledge expands, the prompts provided to the user also expanded to reach a conclusion. When a user consults the program, greetings and introduction about the system will be displayed by the system. The system then prompts the user to write a word and strike the Enter key to trigger an action corresponding to the word. When the user does what was asked by the system, prompts associated to disease/insect symptoms will be displayed asking the user to enter input. Results will be displayed accordingly.

4.4.2.3 Explanation Module

In the system development process a special explanation facilities concerning why the system is asking a certain question and how the system has reached a certain conclusion were provided. Term explanation and detailed information concerning questions asked by the system were included. As the system grows from prototype to prototype, the explanation facility had expanded consequently. The last version used the same explanation capabilities of the second prototype.

4.4.3 Implementation tools & process

Although system shells speed up the implementation process, customized tools provide more flexibility in implementation. For this reason, a general purpose knowledge representation language, prolog 5.10.0 was used. It was chosen because it had been used successfully to develop diagnosing knowledge base systems and it is readily available.

The implementation started using the requirements specifications. In initial stage a first prototype was created from initial knowledge captured from experts. The prototype had pepper diseases, their symptoms and their treatment in its knowledge base. In the second stage a more capable prototype was derived from the initial stage prototype by testing and reviewing the knowledge with experts. The second prototype included insects that attack pepper, their symptoms and their treatments. After the second prototype was tested, the rule base was rearranged based on experts suggestion. Once the prototyping stages were successfully terminated the implementation of the last version of the system started using a valid specification. To make searching efficient, factors that highly distinguish diseases were identified and were given more significance. Hence when the searching takes place factors with more significance were considered first.

4.4.4 System Testing

The developed knowledge base system was tested and evaluated to ensure that the software performance is converging towards established goals. Prototypes were tested whether sufficient and appropriate expert knowledge had been obtained and represented properly for solving the class of problems associated with the given application. Two tests were involved in the evaluation of the knowledge base system. These were system verification and validation.

4.4.4.1 Verification

This test was used to evaluate the completeness and consistency of the knowledge base. The developer practiced different functions of the implemented system looking for potential errors. This was accomplished using different techniques. Generally, these techniques fall into two broad categories. Non Case based techniques which include tracing, spying and other traditional debugging techniques, and Case based verification techniques which are applied by going through the requirements spelled out in the requirements specification document were used.

4.4.4.2 Validation

The system was tested against real problems from its domain. The objective was to determine the effectiveness of the system in solving pepper defects and uncover system deficiencies. Past cases were used for testing. The test cases were then submitted to the expert system and to the three experts from whom the rules had been elicited, and all output was recorded. Output

from the system consisted of disease names and their treatment of 11 complexes (six insect pests and five pepper diseases). Each of the three human experts also responded on the occurrence and treatment of each pepper pest. The output from the knowledge base system was compared to that of the domain experts for accuracy.

An evaluation criterion i.e. if the system's disease identification and diagnosis matches those of the experts who generated the knowledge, was designed to enable a formal judgment on solutions generated by human experts, and the expert system. The selected criteria provides both quantitative and qualitative evaluation basis for judgment. The following is an example for qualitative and quantitative evaluation criteria:

| Grade | Abbreviation | Points |
|--------------|--------------|--------|
| Excellent | E | 3 |
| Good | G | 2 |
| Acceptable | A | 1 |
| Unacceptable | U | 0 |

$$P_i = \frac{3*NE_i + 2*NG_i + 1*NA_i + 0*NU_i}{3*N} * 100$$

Where :

P_i the performance percentage score for expert i

NE_i number of cases evaluated as excellent

NG_i number of cases evaluated as good

NA_i number of cases evaluated as acceptable

NU_i number of cases evaluated as unacceptable

N total number of cases

According to this performance measure, as P_i approaches to 100, the performance of the system will go higher. This is because 'Excellent' has a weight of 3, 'good' has a weight of 2, 'Acceptable' has a weight of 1 and 'Unacceptable' has a weight of 0. Higher performance can be obtained if the cases provided to the system are solved correctly and evaluated as excellent or good by the expert.

Chapter Five

Data Analysis and Knowledge Base System Development

5.1 Introduction

In this chapter the results from the preliminary assessment survey were presented. The knowledge acquisition procedures followed are explained. The represented and implemented knowledge in the computer system is shown. Semantic networks were employed to show the generic representation of the system and detailed knowledge representation was presented using IF-THEN representation technique.

5.2 Preliminary Assessment Analysis

Questionnaire was distributed to pepper growing farmers that are found in Abela. The farmers were asked if pest incidence occurs, if they identify diseases, if they are well aware of control mechanisms and the effectiveness of current pest management practice. There was a 100% response rate because I myself contacted each respondent and filled out the questionnaire for them. After the data was analysed, response average for the occurrence of pest incidence was found to be 4.49, respondents' knowledge of pest identification was 2.01, respondents' knowledge of control mechanisms was 2.35 and adequacy of expert advice was 1.79. The complete data that was collected is tabulated and summarized using percentage and average mean in table 5.1.

| | Strongly agree | Agree | Undecided | Disagree | Strongly disagree | Response Average |
|---|-----------------------|--------------|------------------|-----------------|--------------------------|-------------------------|
| 1. Pest or Disease incident occurs | 58% | 34% | 7% | 1% | 0% | 4.49 |
| 2. I usually identify pests/diseases. | 4% | 9% | 0% | 58% | 29% | 2.01 |
| 3. I usually know which control mechanism to use | 9% | 23% | 2% | 26% | 40% | 2.35 |
| 4. I get adequate advice from experts on pest/disease control. | 0% | 6% | 17% | 27% | 50% | 1.79 |
| 5. Current pest management practices are effective to combat pests. | 13% | 8% | 3% | 40% | 36% | 2.22 |
| 6. Current pest control mechanism increased crop yield. | 7% | 5% | 23% | 35% | 30% | 1.54 |
| 7. Channel is available to report pest/disease incidence | 30% | 62% | 7% | 1% | 0% | 4.21 |
| 8. I usually use chemical pesticides | 33% | 54% | 9% | 4% | 0% | 4.16 |
| 9. I usually know proper dosage of pesticides. | 9% | 20% | 26% | 23% | 22% | 2.71 |
| 10. Usually pesticides have hazards. | 28% | 49% | 17% | 6% | 0% | 3.99 |
| Total respondents | | | | | | 60 |

Table 5.1 Frequency table

The weighted mean and mode was used to analyze the data. Mode shows the attitude of the majority. The mode for the pest incidence parameter was 5, for the knowledge of respondents' to identify pest was 2, knowledge of respondents' on control mechanisms was 1 and adequacy of expert advice was 1. The weighted mean and mode of each question of the survey is presented in table 5.2.

| | N | Weighted Mean | Mode |
|------------------------------|----------|----------------------|-------------|
| Q1 incidence□ | 60 | 4.49 | 5 |
| Q2 identification□ | 60 | 2.01 | 2 |
| Q3 control mechanism□ | 60 | 2.35 | 1 |
| Q4 adequacy□ | 60 | 1.79 | 1 |
| Q5 combat pest□ | 60 | 2.22 | 2 |
| Q6 crop yield | 60 | 1.54 | 2 |
| Q7 channel | 60 | 4.21 | 4 |
| Q8 use of pesticide | 60 | 4.16 | 4 |
| Q9 dosage | 60 | 2.71 | 3 |
| Q10 hazard | 60 | 3.99 | 4 |

Table 5.2 weighted mean and mode of the survey.

Interpretation of survey results

The weighted mean values for most of the parameters were found to be less than 3 which indicates a problem in the corresponding area. For example, the weighted mean values for the responses regarding knowledge to identify diseases/pests, on what type of control mechanism to use, effectiveness of the current pest/disease management practice and knowledge of dose of pesticide they are using were found to be less than 3 which indicates that the farmers are facing problem in these areas. Regarding adequacy of expert advice, the mean value was 1.79 which means the respondents barely believe that they are gaining sufficient advice from experts.

The mode values for the parameters revealed similar results as that of the weighted mean. Large number of respondents i.e. 58% indicated that pest incidence occurs in their field. However 87% of the respondents indicated that they can hardly identify type of disease occurring in their field.

Regarding which control mechanism to use, 66% of the respondents does not know which control mechanism to use. 77% of the respondents pointed out that the advice they are getting from experts was not adequate. Larger number of the respondents i.e. 65% specified current pest management practice is not effective in combating disease and in increasing crop yield. Regarding the proper dosage of pesticide use, 45% of the respondents indicated that they don't know the right dosage.

The result from the survey showed that farmers in the region are facing problems in identifying pepper diseases, use of control mechanism, access to adequate expert advice, knowledge of pesticide dosage, awareness on pesticide hazards and the rest of the parameters used in the questionnaire. The result from the survey also revealed that the respondents indicated that current pest management practice is not doing well in increasing crop production and in combating diseases.

Hence in the present research, a knowledge base system that diagnoses pepper diseases is presented to address these problems.

The system allows easy access to pepper disease diagnosing knowledge so that the knowledge can be used by non experts to advice pepper growers. Hence the system can give the necessary advice on pepper disease treatment where experts are not available.

5.3 KbDiaP: A Knowledge base system to Diagnose Pepper Disease

The Knowledge Base System to Diagnose Pepper Disease (KbDiap) is a knowledge base system that was designed to combine the best available knowledge regarding pepper pest management of disease pathogens and insects to provide potential treatment information to the farmer or non experts. From inputs describing how the symptom or the pest looks, KbDiaP produces a pest or disease identification information plus the measure that needs to be taken. The insects included were the Aphid, Red Spider, Leaf Miner, Flee Beetle, White Fly, African Bull Worm; and the diseases included were Powdery Mildew, Bacterial Spot, Bacterial Soft Rot, “ATEWLIG” and Virus.

5.3.1 Knowledge Acquisition

Mainly interviewing was used to acquire the knowledge. Interviewing consists of asking the domain expert questions about the domain of interest and how they perform their tasks. The interview used was unstructured in which answers for the questions were given by the domain experts without answers being determined first. The experts were asked mainly the questions, “Which diseases attack pepper in the country?”, “What is the symptom of each disease; on stem, leaf, seed?” “What are the treatments?” Initially disease, symptoms and treatments were identified. Then rules were edited via the iterative knowledge elicitation.

5.3.2 Knowledge Model

The model was built by the knowledge engineer based on information obtained during interviews with experts and then reviewed with the domain expert. Semantic networks were used to model the knowledge. Fig 5.1 shows the semantic network for pepper diseases and Fig 5.2 shows the semantic network for pepper insects.

Semantic Networks

In the semantic network Diseases, Pepper, Stem, Leaf, pod were put as a node and relationships such as is-a and has-a were used to form associations between nodes.

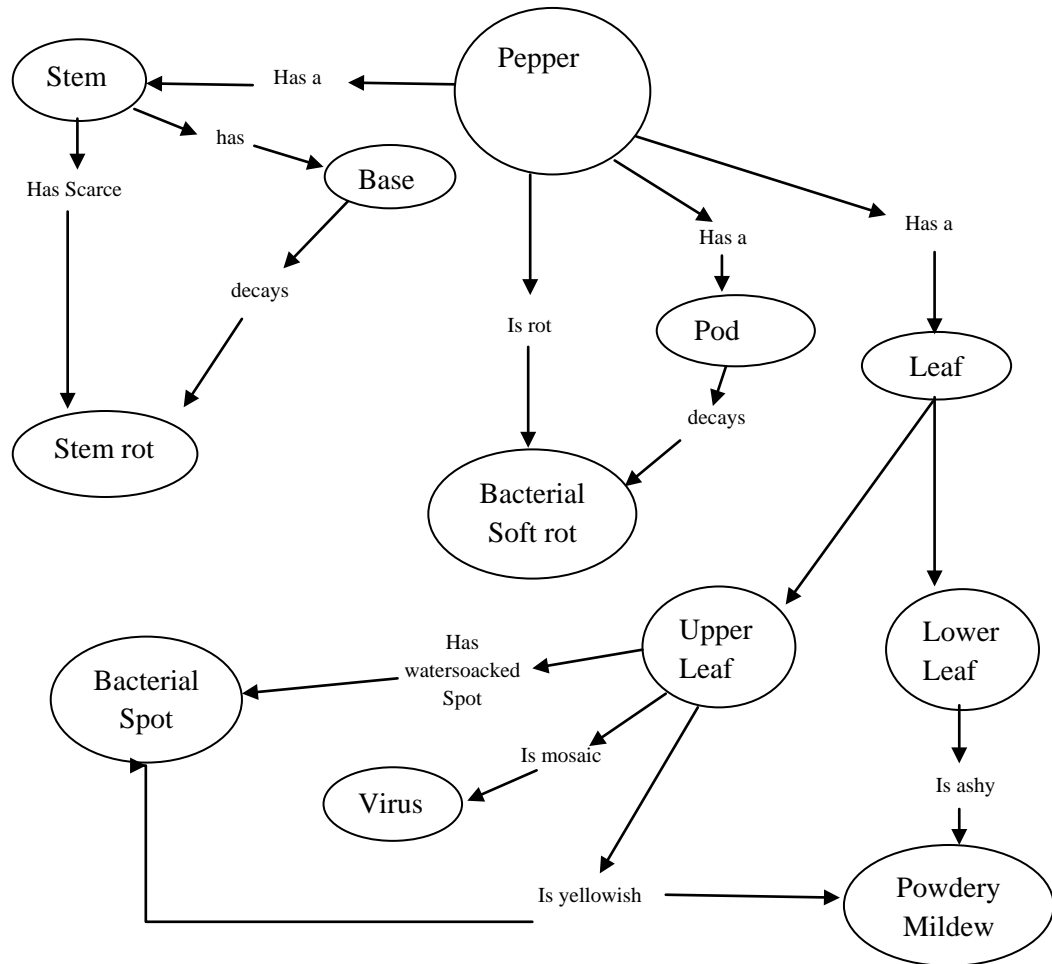


Fig 5.1 semantic network for pepper and its diseases.

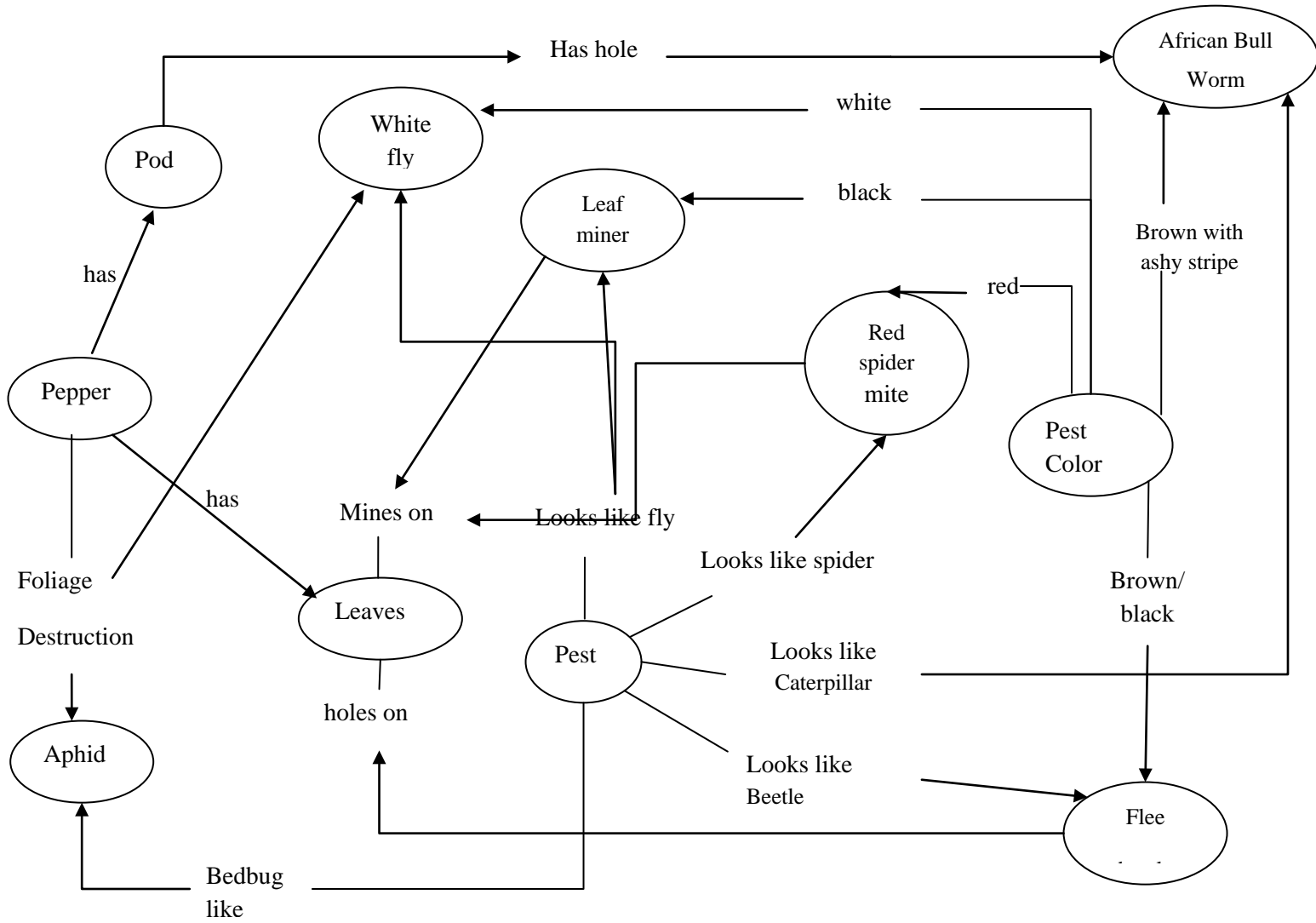


Fig.5.3 Semantic network for insects causing Pepper disease.

5.3.3 Knowledge Representation

Production rules were created to represent the detailed knowledge, based on expert interviews and these were later translated into prolog code. The rules were made up of “Qualifiers”, “Choices” and “Variables”. A choice is a possible final conclusion of the system (e.g. a particular disease). Qualifiers form the conditions that must be met for individual choices to be selected (Norton and Mumford, 1993). Variables are used to receive user inputs in the form of text, string or numerical values.

| |
|--|
| Qualifier 1: The observed pest is (Q1) <ol style="list-style-type: none">1. Fly-like2. Beetle- like3. Caterpillar-like4. bedbug- like5. Spider-like |
| Qualifier 2. The damage appears as (Q2) <ol style="list-style-type: none">1. Holes (on leaf, on pod)2. Foliage destruction3. Mines in leaves |
| Qualifier 3 pest color (Q3) <ol style="list-style-type: none">1. White2. Black3. Red4. Brown with ashy stripe |

Table 5.3 Insect registration sheet

Qualifier 1. Symptoms appear on (Q1):

1. Leaf
2. Stem
3. pod
5. Seedlings
6. Entire plant

Qualifier 2. Are spots visible (Q2)?

1. Yes
2. No

Qualifier 3. Color of the spots (Q3)?

1. Watersoaked
2. Ashy and yellowish
3. Mosaic
4. Black
5. yellowish

Qualifier 4 Is scarce visible?

1. Yes
2. No

| |
|--|
| <p>Qualifier 5. Scarce appears on</p> <ol style="list-style-type: none"> 1. Stem 2. Leaf |
| <p>Qualifier 6. Does the pod rot?</p> <ol style="list-style-type: none"> 1. Yes 2. No |
| <p>Qualifier 7. Does leaf roll?</p> <ol style="list-style-type: none"> 1. Yes 2. No |

Table 5.4 Symptoms registration sheet

| |
|---|
| <p>Feeding habitat___leaves</p> <p>Damage__mines on leaves (1)/destroyed leaves (2)</p> <p>Pest___Fly-like (1)/ Beetle-like (2)</p> <p>Pest_Color___Yellow_Black</p> |
| <p>Possible pests</p> <ol style="list-style-type: none"> 1. Leafminers (<i>Liriomyza</i> spp.) 2. Colorado Potato Beetle (<i>Leptinotarsa decemlineata</i>) |

Table 5.5 Insect identification form

| |
|--|
| Symptoms appear on___leaves Are spots visible__Yes Color of the spots___Watersoaked greyish On Pod___yellow green hole |
| Possible Diagnosis Bacterial speck (<i>Pseudomonas syringae</i>) Bacterial spot (<i>Xanthomonas vesicatoria</i> ; <i>X. campestris</i> .) |

Table 5.6 Disease diagnosis form

Production rules: IF_THEN representation of the Knowledge Base System

Once the qualifiers, choices and variables are identified, the next task done was to represent the knowledge of each disease using in the rule base. The methods and rules were represented using IF-THEN statements. The entire rule base for KbDiaP is presented in the following tables.

| |
|---|
| IF: Symptoms appear on___leaves, Are spots visible__Yes, Color of the spots___ Watersoaked yellowish, Plant_fall_down_and_withers |
| THEN: Possible Diagnosis Bacterial spot (<i>Xanthomonas vesicatoria</i> ; <i>X. campestris</i> .) |
| Treatment Spray a solution of 15 liter water and 25 gram cosaid once in 10 days. |

Table 5.7 Disease diagnosis form for Bacterial spot

| |
|--|
| <p>IF:Symptoms appear on___leaves, Are spots visible__no, Is scarce visible_no, bottomleaf_ashy, upperleaf_yellowish.</p> |
| <p>THEN: Possible Diagnosis Powdery Mildew (<i>Leveillula Taurica.</i>)</p> |
| <p>Treatment Spray a solution of 15 liter water and 400 gram bailaton once in a week.</p> |

Table 5.8 Disease diagnosis form for Powdery mildew

| |
|---|
| <p>IF: Symptoms appear on___stem, Are necrotic spots visible__no, Pod contains water_yes Does the Pod rot_and_decay_yes.</p> |
| <p>THEN: Possible Diagnosis Bacterial Soft Rot (Erwinia Carotovora.)</p> |
| <p>Treatment No known effective pest control measure. Avoid plant injury or wound. Drain water. Crop rotation for prevention.</p> |

Table 5.9 Disease identification form for Bacterial Soft Rot.

| |
|---|
| <p>IF:Symptoms appear on___stem, Is scarce visible_yes, Plant_withers stem_whitish_and_decay.</p> |
| <p>THEN: Possible Diagnosis Stem Rot</p> |
| <p>Treatment Avoid planting in infested fields or during warm, wet periods if possible. Do not introduce diseased transplants into the field. Eliminate diseased plants and weeds from the field. Do not injure pepper stems during cultivation.Crop rotaion.</p> |

Table 5.10 Disease identification form for Stem rot.

| |
|--|
| <p>IF: Feeding habitat___leaves, Damage__mines on leave, Pest___spider-like, Pest_Color___red.</p> |
| <p>THEN:Possible pests Red spider mite</p> |
| <p>Treatment Spray nim products. Or spray a solution of water and soap.</p> |

Table 5.11 Insect identification form for Red spider mite

| |
|---|
| <p>IF: Feeding habitat___pod_and_leaf , Damage__hole on pod and leaf, Pest___Caterpillar-like, Pest_Color___brown with ashy stripe.</p> |
| <p>THEN: Possible pests African Bull Worm</p> |
| <p>Treatment Spray insecticide if the plant is at its early stage.</p> |

Table 5.12 Insect identification form for African Bull Worm.

| |
|---|
| <p>IF: Feeding habitat___entire plant, Damage__foilage destruction, Pest___ bedbug-like, Cluster seen_yes</p> |
| <p>THEN: Possible pests Aphid</p> |
| <p>Treatment Avoid spraying insecticide as it aggravates aphid attack. Spray a solution of 1 tea spoon of dishwashing soap water.</p> |

Table 5.13 Insect identification form for Aphid

| |
|---|
| <p>IF: Feeding habitat___entire plant Damage__suck tender foilage Pest___ fly-like Pest_Color___white Fly like cloud_yes</p> |
| <p>THEN: Possible pests White fly (Bemisia tabaci)</p> |
| <p>Treatment Spray nicotinoids.</p> |

Table 5.14 Insect identification form for White fly

| |
|--|
| <p>IF: Feeding habitat__seedling</p> <p>Damage__hole on leaf</p> <p>Pest__ beetle like</p> <p>Pest_Color__brown/black</p> |
| <p>THEN: Possible pests</p> <p>Flea Beetle</p> |
| <p>Treatment</p> <p>Sprays combining rotenone with insecticidal soap</p> |

Table 5.15 Insect identification form for Flea beetle.

| |
|--|
| <p>IF: Feeding habitat__leaf</p> <p>Damage__ White line in the middle if the infected leave</p> <p>Pest__ fly like</p> <p>Pest_Color__brown/black</p> |
| <p>THEN: Possible pests</p> <p>Leaf Miner</p> |
| <p>Treatment</p> <p>Spray insecticide.</p> |

Table 5.16 Insect identification form for Leaf miner

| |
|---|
| <p>IF: Symptom appear on ___ leaf</p> <p>Color Mosaic</p> <p>Curly_small_deformed_leaf</p> <p>Very small Pod</p> |
| <p>THEN: Possible Disease</p> <p>Virus</p> |
| <p>Treatment</p> <p>Screen for insect control, use stylet oil spray, and remove weeds.</p> |

Table 5.17 Disease identification form for Virus.

5.3.4 Implementation

KbDiaP is a rule-based expert system and was developed on the Microsoft Windows platform using the prolog development environment. Prolog was chosen on the basis that it had been used successfully to develop a similar knowledge-based system (Hailu M., 2009) and that it was readily available.

The knowledge represented using IF_THEN rules in the design phase was used to construct the knowledge base of the system. The IF_THEN rules were converted to prolog code so that they can be easily inferred by the expert system shell. The implementation of the expert system shell uses backward chaining logic in its inferencing. Primarily, symptoms of insects and/or diseases were recognized. Subsequently, particular disease corresponding to the symptoms and actions that need to be taken were identified. This process, proceeding from the end result through a logical or causal chain back to symptoms, was well suited to developing a backward chaining system.

The explanation module is included in the expert system shell so that the system explains why it reached a certain conclusion when prompted by the user.

A user interacts with the system through dialogue. The system forwards successive questions to identify the disease/insect occurrence and to recommend treatment. Based on the symptoms observed and fed to the system, the system displays a disease/insect associated to the symptoms and appropriate actions that need to be taken.

When the system runs in a consulting mode, it is initialized by trying to solve for the desired end results, i.e., the insect or disease. For example, when asked for consultation the system will ask the existence of several symptoms associated with a particular disease. The system prompts the user to input information pertaining to the individual pest management scenarios being analyzed and then the system checks whether any of the symptoms is present for the entered scenario. Accordingly a disease associated to the symptoms fed and the treatment will be displayed to the user.

5.3.5 System Evaluation

5.3.5.1 Verification

The developer practiced different functions of the implemented systems, looking for potential errors. The task was accomplished using different techniques. Basically, these techniques fall into two broad categories. Non Case based techniques which include tracing, spying and other traditional debugging techniques, and Case based verification techniques which are applied by going through the requirements spelled out in the requirements specification document. Both techniques were applied to verify the current system and corrections were made accordingly.

5.3.5.2 Validation

Validation was performed to test the hypothesis that the expert system's recommendations came from the same underlying knowledge as that of the experts. In other words, the results of the expert system should be indistinguishable from the opinions of the experts it was meant to mimic, when applied to independent test scenarios.

In this validation, 15 hypothetical test cases were created using randomly generated values for the user input. Inputs to the system were allowed to vary independently, so to check that the resulting test cases were realistic and representative. The results were reviewed by an expert who was not involved in the development of the system.

Test cases are prepared manually by knowledge engineers. A copy of the selected test cases was given to the three domain experts. The same cases are introduced to the expert system. Each of the domain experts as well as the Expert system works out the test cases independently. Solutions of test cases were evaluated in a blind manner, so that distinguishing between solutions of the expert system and solutions of domain experts becomes impossible. A domain expert, other than those who participated in the knowledge acquisition process and other than those who solved the test cases was given test cases solutions for evaluation according to the previously prepared formula. A score was given to each solution, and solutions are ranked according to these scores. Based on this the system's performance was calculated 86.6%. Test cases used are attached in appendix II.

Chapter Six

Conclusions and Recommendations

It is interesting that the methods of evaluation used to validate this knowledge base system showed 86.6% of accuracy by the system. Statistical method was used to evaluate the system as mentioned in section 4.4.4.2. In fact, choosing a statistical test is not an exact science. This aspect of evaluating agricultural knowledge base systems clearly could benefit from further investigation. Hence this chapter explains the findings of this research. What has been done so far by the researcher and what can be done in the future are summarized under conclusions and recommendations.

6.1 Conclusions

The research was aimed at assessing the current pest management practice by farmers and the problems they are facing and to come up with a knowledge base system that diagnoses pepper diseases. In order to reach these goals a structured questionnaire was prepared and distributed to farmers. The questionnaire was prepared in likert scale format. The collected data was analyzed using weighted mean and mode. The results from the preliminary assessment revealed that farmers faced problem to identify type of disease occurred in their field and on dosage of pesticides use. The survey also revealed that the farmers are not well convinced on the effectiveness of their current pest/disease management in increasing their crop production. Hence to deal with such problems a knowledge base system was proposed and developed in this research. The knowledge base system addresses only a fraction of the entire problem i.e. pepper diseases due to time and resource constraints. The knowledge base was developed so that farmers would get the necessary knowledge to manage pepper pest/diseases as they occurred. The system development process employed knowledge engineering and software engineering methodologies. Knowledge engineering methodology used has three tasks namely Knowledge acquisition, Knowledge verification and knowledge representation. IF_THEN rules were used to represent the knowledge. Software engineering methodologies included requirement specification, design, implementation and testing. These tasks were done iteratively to produce successive prototypes until the final version of the system was delivered.

Test cases were used to validate and measure the accuracy of the system the system was found to be 86.6% accurate.

The treatments suggested by the system were obtained from experts who has years of experience in the field. Literatures were also consulted in the development process. The results showed that it is possible to simulate the expertise of pest management with a reasonable degree of accuracy. Combining the expertise of three specialists to cover the areas of insect and disease management makes this an efficient method for synthesizing a variety of pest management knowledge. It is only with this gathering of knowledge that it will be possible to make holistic choices about pest management in our diverse ecosystems.

6.2 Recommendations

After working on this research the following recommendations are forwarded:

- Pests and diseases attack not only pepper, but also almost every crop that has been grown in the country at different degree of intensity and this research showed that knowledge base systems can be developed and provide advice almost as accurate as human experts in the field. Thus concerned bodies i.e. the government and Ministry of Agriculture can take measures to initiate and support the development of similar systems for other crops.
- The developed system can further be expanded by taking location and season of production in to consideration. Including these factors enhances the capability of the system and produces better results.
- This knowledge base system needs more testing before it can be used in the field. While the insect component has reached an acceptable level of performance and is ready for further implementation, the disease identification components need more work. One area that needs more attention is the consideration of field condition i.e soil, water, temperature and humidity as a factor in disease incidence. In this research all weather events were given equal chances of occurring throughout the state. By incorporating historical weather patterns in to the knowledge base there may be a way to weight weather events by region or by county. Besides, although this system has shown that it agrees with the experts in the majority of cases, it has not been tested sufficiently by the target users, the agricultural Extension Agents and farmers in the field.

References

1. Abate T. 1997a. Integrated pest management in Ethiopia: an overview. In *Integrating biological control and host plant resistance*, Proceedings of a CTA/IAR/IIBC Seminar, 9-4 October 1995, pp. 24-37. CTA: Wageningen.
2. Agranovsky AA . 1993. Virus diseases of pepper (*Capsicum annum L*) in Ethiopia. *Phytopathology* 138:89-97.
3. Avison, D.E., and, Fitzgerald, G. 1998. *Information Systems Development: Methodologies, Techniques and Tools* (2nd Ed.). McGraw-Hill, London.
4. B.G. Buchanan and E.H. Shortliffe (eds.). 1985. *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*, Reading, MA: Addison-Wesley.
5. Berke, T. 2002. The Asian vegetable research and development center pepper project. pp. 1-16. Proceedings of the 16th International Pepper Conference. Tampico, Mexico. November 10-12, Tamaulipas.
6. Blackmore, B. 1999. Developing the principles of precision farming. Proceedings of 6grotech 99, Barretos, November 15-19, Barretos Institute of Technology, Brazil, pp: 133-250.
7. Blackmore, B. 2000. Using information technology to improve crop management. Proceedings of AgMet Millennium Conference, Feb.29, Institute for Jordbrugsvidenskab, Dublin, pp: 30-38.
8. Boseland, P.W. and Votava, E.J. 2000. *Pepper: Vegetable and Spice Capsicum*. CABI publishing, New York. pp. 1-16.
9. Builders, Information. 1994. *Level5 Object for Microsoft Windows*. New York: Information Builders, Inc.
10. Creswell, J. W. 1994 . *Research Design: Qualitative and Quantitative Approaches*. Thousand Oaks, CA: SAGE.

11. CSA. 2003. Area and Production of Crops. Country Level, Part II, Addis Ababa. pp. 323-27.
12. CSA. 2003. Ethiopian Sample survey Enumeration. Addis Ababa. 2(341). pp. 382-385.
13. D.N. Chorafas. 1992. *Expert Systems in Manufacturing*, New York, NY: Van Nostrand Reinhold.
14. Davenport, T.H., & Prusak, L. 1998. Working knowledge: how organisations manage what they know. Boston, MA: Harvard Business School Press.
15. Davis, R.; Shrobe, H.; and Szolovits, P. 1993. What Is a Knowledge Representation? Memo, AI Laboratory, Massachusetts Institute of Technology.
16. Durkin J. 1993. *Expert Systems: Catalog of Applications*, Akron, OH: Intelligent Computer Systems.
17. EEPA (Ethiopian Export Promotion Agency). 2003. Spice potential and market study, Addis Ababa. 103p
18. ESEF (Ethiopian Spice Extraction Factory), 2005. Unpublished data
19. EXSYS. 1992. Expert System Development Software. User Manual. EXSYS Inc.
20. F. Hayes-Roth. 1985. Rule-based systems, *Communications of the ACM*, **28** (9): 921-932.
21. Fahad Shahbaz Khan, Saad Razzaq, Kashif Irfan, Fahad Maqbool, Ahmad Farid, Inam Illahi, and Tauqeer ul amin. 2008. "Dr. Wheat: A Web-based Expert System for Diagnosis of Diseases and Pests in Pakistani Wheat," Proceedings of the World Congress on Engineering Vol.I.
22. FAOSTAT. 2002. Online database of Food and Agriculture organization of United Nations. Rome, Italy.
23. Galliers, R.D., and Newell, S. 2000. Back to the future : from Knowledge Management to Data Management. Department of Information Systems, Working Paper 92, London School of Economics and Political Science, London, UK.
24. Giarratano, J.C. 2002. CLIPS User's Guide, Software Technology Branch. Version 6.20, NASA Lyndon B. Johnson Space Center, Houston, TX.
25. Guarino, N. and Giaretta, P. 1995. Ontologies and Knowledge Bases: Towards a Terminological

Clarification. In N. Mars (ed.) *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing*. IOS Press, Amsterdam: 25-32.

26. Hart, Anna. 1992. *Knowledge Acquisition for Expert Systems*. New York, NY: McGraw-Hill, Inc.
27. Ignizio, J.P. 1991. *Introduction to Expert Systems*. 1st Edn. The Development and Implementation of Rule-Based Expert Systems. McGraw-Hill, Inc.
28. Knight, Jonathan D., and John D. Mumford. 1994. Decision Support Systems in Crop Protection. *Outlook on Agriculture* 23 (4): 281-285.
29. Kramers, M.A., Conijn, C.G.M., Bastiaansen, C. 1998. EXSYS, an expert system for diagnosing flowerbulb diseases, pests and non-parasitic disorders. *Agricultural Systems* 58 (1), 57–85.
30. Mahaman, B.D., H.C. Passam, A.B.Sideridis and C.P.Yialouris. 2003. “DIARES-IPM:a Diagnostic advisory rule-based expert system for the integrated pest management in the Solanaceous crop system,” *Agricultural system* vol. 76, pp.1119-1135.
31. Maloy, O.C. 2005. Plant disease management. The Plant Health Instructor, 10.1094/PHI-I-2005-0202-01.
32. MARC (Melkassa Agricultural Research Center). 2004. Method of pepper production. Broacher
33. Michael Genesereth and Nils J. Nilsson. 1987. *Logical Foundations of Artificial Intelligence*. Morgan Kaufmann, San Mateo, California.
34. Michalski, R. S., Carbonell, J. G. & Mitchell, T. M. 1983. *Machine Learning: An Artificial Intelligence Approach*. California, USA: Tioga Publishing Co.
35. M. Minsky. 1975. A framework for representing knowledge, In P. Winston (ed.), *The Psychology of Computer Vision*, New York, NY: McGraw-Hill.
36. Nikolopoulos, Chris. 1997. *Expert Systems*. New York: Marcel Dekker, Inc.
37. Nonaka, I. & Takeuchi, H. 1995. *The knowledge creating company: how Japanese companies create the dynasties of innovation*. Oxford: Oxford University Press.
38. Norton G. A, Mumford J. D. 1993. Decision Tool for Pest Management, Cab International, pp 279.
39. NOY, N.F. and HAFNER, C.D. 1997. “The State of the Art in Ontology Design – A Survey and Comparative Review”, *AI Magazine*, 36(3), 53-74.

40. Plant, R.E., Stone, N.D. 1991. Knowledge-based Systems in Agriculture. McGraw-Hill, New York, pp 364.
41. Polanyi. 1966. *The tacit dimension*. New York, NY: Doubleday.
42. Quillian, M. R. 1967. Word concepts: A theory and simulation of some basic semantic capabilities. *Behavioral Sciences*, 12, 410-430.
43. Rafea, A., Warkenin, M., Ruth, S. 1991. Knowledge engineering: creating expert systems for crop production management in Egypt. In: Ruth, S.R., Mann, C. (Eds.), *Expert Systems in Developing Countries: Practice and Promise*. Westview Press, Boulder, pp.89–104.
44. Rafea, A., El-Azhari, S., Ibrahim, I., Edrees, S., Mahmoud, M. 1995. Experience with the Development and Deployment of Expert Systems in Agriculture. Proceedings of IAAI-95 conference, Montreal-Canada.
45. Robert K. Lindsay, Edward A. Feigenbaum, B. G. Buchanan, J. Lederberg. 1980. *Applications of Artificial Intelligence for Chemical Inference: The Dendral Project*, McGraw-Hill, Inc., New York, NY.
46. Robert S. Englemore. 1993. Knowledge-based systems in Japan.
47. Rubatzky, V. and Yamaguchi, M. 1997. *World Vegetables*, 2nd Edition. International Thomson Publishing, USA. pp. 553-562.
48. S.J. Russell and P. Norvig. 1995. *Artificial Intelligence: A Modern Approach*. Englewood Cliffs, NJ: Prentice-Hall.
49. Schmoldt, Daniel L. and Rauscher, H. Michael. 1996. *Building Knowledge-Based Systems for Natural Resource Management*, New York: Chapman & Hall.
50. Sell, P. S. 1985. *Expert systems: a practical introduction*. New York: John Wiley & Sons, Inc.
51. Shu-Hsien, L. 2005. Expert system methodologies and applications. A decade review from 1995 to 2004 *Expert Syst. Applied*, 28: 93-103.
52. Sowa, J.F. 2000. *Knowledge Representation : Logical, Philosophical, and Computational Foundations*. Brooks Cole : Pacific Grove, CA, USA.
53. Tameru A. 2004. Characterization of virus of pepper (*Capsicum spp.*) and sweet potato (*Ipomoea batatas*) from Ethiopia. PhD thesis, University of Bonn, Germany. p. 150.

54. Tameru A, Hamacher J, Dehne HW. 2003. The increase in importance of Ethiopian pepper mottle Virus (EPMV) in the rift valley parts of Ethiopia: time to create awareness among researchers and extension workers. Paper presented at "Deutsches Tropentage", October 18-21, 2003, Gottingen, Germany.
55. Tesfaye Tedla and Habtu Assefa. 1985. A review of vegetable disease research in Ethiopia. Pp 495 – 518 In Tsedeke Abate (ed.). Proceedings of the 1st Crop Protection Symposium, 4 – 7 Feb. 1985, Addis Ababa, Ethiopia.
56. Tsedeke Abate and Gashawbeza Ayalew. 1997. Sources of fruit worms resistance in Tomato. PMJOE vol. 1 (1 & 2): 1-7.
57. Tsedeke Abate and Gashawbeza Ayalew. 1994. Progress in Vegetable Pest Management Research, 1985 – 1992. Pp 187 -193, In E. Herath and Lemma D. (eds.). Proceedings of the Second National Horticultural Workshop of Ethiopia, 1- 3 December 1992, Addis Ababa, IAR/FFAO.
58. Walters, J. R., and N. R. Nielsen. 1988. *Crafting knowledge-based systems: expert systems make realistic*. New York: Wiley & Sons.
59. Waterman D. A. 1986. A guide to expert systems. Addison-Wesley, Reading, MA. 419 p., Liebowitz.
60. Yasser Abdelhamid, Hesham Hassan, Ahmed Rafea. 1993. A Proposed System for Expert System Engineering, 2p., *Central Laboratory for Agricultural Expert Systems*.
61. Yialouris C.P. and Sideridis A.B. 1996. "An expert system for the tomato diseases," *Computers and Electronics in Agriculture* vol.14, pp.61-76.

Appendix I

Survey on “PEST MANAGEMENT KNOWLEDGE AND PRACTICE”

Dear Participant,

I am currently writing my thesis to finalize my masters study that I've been attending at Addis Ababa University, Addis Ababa.

I am researching in the field of knowledge base systems for pepper disease diagnosis. More precisely, I am exploring the current pest management knowledge and practice of farmers. After the assessment, a knowledge base system for pepper disease diagnosis will be developed as a solution. You have every right to withdraw from this survey at any point if you are not comfortable.

Thank you very much for taking time to read and answer this short questionnaire!

Please read and answer all questions. Thank you very much again for your support!

Bethlehem Asferi

Addis Ababa University

Addis Ababa

April 2010.

Knowledge of farmers in disease identification and control mechanism

| | Strongly Agree | Agree | Undecided | Disagree | Strongly disagree |
|---|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| 1. Disease usually occurs <i>(tick any that apply)</i> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 2. I usually identify the disease | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 3. I usually use pesticides | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 4. I usually know the proper doses of every pesticide I use | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 5. Pesticides has hazards | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 6. I usually know what control mechanisms to use | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

Effectiveness of current disease management practice

| | | | | | |
|--|--|-----------------------------------|---------------------------------------|--------------------------------------|---|
| 7. Current pest/disease control practice is effective to combat pest/disease | Strongly Agree <input type="checkbox"/> | Agree <input type="checkbox"/> | Undecided <input type="checkbox"/> | Disagree <input type="checkbox"/> | Strongly disagree <input type="checkbox"/> |
| 8. Current pest/disease control practice has increased crop production | Strongly Agree <input type="checkbox"/> | Agree <input type="checkbox"/> | Undecided <input type="checkbox"/> | Disagree <input type="checkbox"/> | Strongly disagree <input type="checkbox"/> |
| 9. Channel is always available to report any disease incident | Strongly Agree <input type="checkbox"/> | Agree <input type="checkbox"/> | Undecided <input type="checkbox"/> | Disagree <input type="checkbox"/> | Strongly disagree <input type="checkbox"/> |
| 10. Expert advice was adequate | Strongly Agree <input type="checkbox"/> | Agree <input type="checkbox"/> | Undecided <input type="checkbox"/> | Disagree <input type="checkbox"/> | Strongly disagree <input type="checkbox"/> |

Appendix II

Questions presented to domain Experts

1. what pepper pests/diseases are identified in our country?
2. which pepper pests/diseases have more economic importance?
3. what are the symptoms for each pest/disease?
4. what are the treatments for each pest/disease?
5. what steps do you follow to diagnose pest/disease?
6. which pest/disease symptoms are more discriminating?

Appendix III

Test cases

In this section sample of the test cases used to validate the system by an expert were presented. The input fed by the expert and corresponding output displayed by the system were provided.

Test case for DISEASE

% library(win_menu) compiled into win_menu 0.00 sec, 12,912 bytes

% library(swi_hooks) compiled into pce_swi_hooks 0.00 sec, 2,404 bytes

% c:/users/sweetie/appdata/roaming/swi-prolog/pl.ini compiled 0.00 sec, 524 bytes

Warning: c:/users/sweetie/documents/prolog/peppershell.pl:8:

Singleton variables: [X,CF]

ERROR: c:/users/sweetie/documents/prolog/peppershell.pl:343:

No permission to modify static_procedure `retractall/1'

Warning: c:/users/sweetie/documents/prolog/peppershell.pl:344:

Singleton variables: [Y]

ERROR: c:/users/sweetie/documents/prolog/peppershell.pl:344:

No permission to modify static_procedure `retractall/1'

Warning: c:/users/sweetie/documents/prolog/peppershell.pl:345:

Singleton variables: [X]

ERROR: c:/users/sweetie/documents/prolog/peppershell.pl:345:

No permission to modify static_procedure `retractall/1'

% c:/Users/Sweetie/Documents/Prolog/peppershell.pl compiled 0.02 sec, 14,444 bytes

Welcome to SWI-Prolog (Multi-threaded, 32 bits, Version 5.10.0)

Copyright (c) 1990-2010 University of Amsterdam, VU Amsterdam

SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software,

and you are welcome to redistribute it under certain conditions.

Please visit <http://www.swi-prolog.org> for details.

For help, use `?- help(Topic).` or `?- apropos(Word).`

1 ?-

% c:/Users/Sweetie/Documents/Prolog/pepper.pl compiled 0.00 sec, 5,336 bytes

1 ?- solve.

Hello. Please answer my questions with one of the following:

"y.", which means "yes" (confidence value 100).

"n.", which means "no" (confidence value -100).

A number followed by a period. The number, which should be between -100 and 100, is your confidence in the truth of the query.

"why." to get an explanation of why I ask this question.

"how(X)." to find out how confident I am in concluding X,
and how I reached that conclusion.

"rule(X)." to display all rules that conclude about X.

"halt." to terminate consultation.

"help." to print this message.

Is it true that `upper_leaf(yellowish)`

? y.

Is it true that `lower_leaf(ashy)`

? y.

Is it true that plant(withers)

? y.

treatment(Spray a solution of 15 liter water and 400 gram bailaton once in a week.) was concluded with certainty 90.25

The proof is

treatment(Spray a solution of 15 liter water and 400 gram bailaton once in a week.) CF= 90.25 :-

disease(powdery_mildew) CF= 90.25 :-

bad_part(leaf) CF= 95 :-

upper_leaf(yellowish) CF= 100 was given by the user

lower_leaf(ashy) CF= 100 was given by the user

plant(withers) CF= 100 was given by the user

treatment(powdery_mildew, Spray a solution of 15 liter water and 400 gram bailaton once in a week.)
CF= 100 was a fact in the knowledge base

true .

2 ?- solve.

Hello. Please answer my questions with one of the following:

"y.", which means "yes" (confidence value 100).

"n.", which means "no" (confidence value -100).

A number followed by a period. The number, which should be

between -100 and 100, is your confidence in the truth of the query.

"why." to get an explanation of why I ask this question.

"how(X)." to find out how confident I am in concluding X,

and how I reached that conclusion.

"rule(X)." to display all rules that conclude about X.

"halt." to terminate consultation.

"help." to print this message.

Is it true that upper_leaf(yellowish)

? n.

Is it true that upper_leaf(spoty)

? y.

Is it true that lower_leaf(_G339)

? y.

Is it true that plant(withers)

? y.

treatment(Spray a solution of 15 liter water and 400 gram bailaton once in a week.) was concluded with certainty 90.25

The proof is

treatment(Spray a solution of 15 liter water and 400 gram bailaton once in a week.) CF= -90.25 :-

disease(powdery_mildew) CF= -90.25 :-

bad_part(leaf) CF= -95 :-

upper_leaf(yellowish) CF= -100 was given by the user

lower_leaf(ashy) CF= 100 was given by the user

plant(withers) CF= 100 was given by the user

treatment(powdery_mildew, Spray a solution of 15 liter water and 400 gram bailaton once in a week.)
CF= 100 was a fact in the knowledge base

true .

3 ?- solve.

Hello. Please answer my questions with one of the following:

"y.", which means "yes" (confidence value 100).

"n.", which means "no" (confidence value -100).

A number followed by a period. The number, which should be between -100 and 100, is your confidence in the truth of the query.

"why." to get an explanation of why I ask this question.

"how(X)." to find out how confident I am in concluding X,

and how I reached that conclusion.

"rule(X)." to display all rules that conclude about X.

"halt." to terminate consultation.

"help." to print this message.

Is it true that upper_leaf(yellowish)

? n.

Is it true that upper_leaf(spoty)

? n.

Is it true that upper_leaf(ranger)

? n.

Is it true that stem(spoty)

? y.

Is it true that stem(whitish)

? n.

Is it true that stem(scarce)

? n.

Is it true that plant(rots)

? y.

treatment(Drain water. Crop rotation.) was concluded with certainty 90.25

The proof is

treatment(Drain water. Crop rotation.) CF= 90.25 :-

disease(bacterial_soft_rot) CF= 90.25 :-

bad_part(stem) CF= 95 :-

stem(spoty) CF= 100 was given by the user

stem(spoty) CF= 100 was given by the user

plant(rots) CF= 100 was given by the user

treatment(bacterial_soft_rot,Drain water. Crop rotation.) CF= 100 was a fact in the knowledge base

true .

5 ?- solve.

Hello. Please answer my questions with one of the following:

"y.", which means "yes" (confidence value 100).

"n.", which means "no" (confidence value -100).

A number followed by a period. The number, which should be between -100 and 100, is your confidence in the truth of the query.

"why." to get an explanation of why I ask this question.

"how(X)." to find out how confident I am in concluding X,
and how I reached that conclusion.

"rule(X)." to display all rules that conclude about X.

"halt." to terminate consultation.

"help." to print this message.

Is it true that upper_leaf(yellowish)

? n.

Is it true that upper_leaf(spoty)

? n.

Is it true that upper_leaf(ranger)

? y.

Is it true that lower_leaf(_G920)

? y.

Is it true that plant(withers)

? y.

treatment(Spray a solution of 15 liter water and 400 gram bailaton once in a week.) was concluded with certainty 90.25

The proof is

treatment(Spray a solution of 15 liter water and 400 gram bailaton once in a week.) CF= -90.25 :-

disease(powdery_mildew) CF= -90.25 :-

bad_part(leaf) CF= -95 :-

upper_leaf(yellowish) CF= -100 was given by the user

lower_leaf(ashy) CF= 100 was given by the user

plant(withers) CF= 100 was given by the user

treatment(powdery_mildew, Spray a solution of 15 liter water and 400 gram bailaton once in a week.)
CF= 100 was a fact in the knowledge base

true

Test case for INSECT

Is it true that pest_look(spider_like)

? n.

Is it true that pest_look(caterpillar_like)

? y.

Is it true that pest_color(brown_with_ashy_stripe)

? y.

Is it true that damage_on_plant(mines_on_leave)

? n.

Is it true that pest_look.bedbug_like)

? n.

Is it true that pest_look(fly_like)

? n.

Is it true that pest_look(beetle_like)

? n.

Is it true that damage_on_plant(hole_on_pot)

? y.

treatment(Spray pesticide if the plant is at its early stage.) was concluded with certainty 90.25

The proof is

treatment(Spray pesticide if the plant is at its early stage.) CF= 90.25 :-

insect(helicovera_armigera) CF= 90.25 :-

pest(infecting) CF= 95 :-

pest_look(caterpillar_like) CF= 100 was given by the user

pest_color(brown_with_ashy_stripe) CF= 100 was given by the user

damage_on_plant(hole_on_pot) CF= 100 was given by the user

treatment(helicovera_armigera,Spray pesticide if the plant is at its early stage.) CF= 100 was a fact in the knowledge base

true .

3 ?- solve.

Hello. Please answer my questions with one of the following:

"y.", which means "yes" (confidence value 100).

"n.", which means "no" (confidence value -100).

A number followed by a period. The number, which should be between -100 and 100, is your confidence in the truth of the query.

"why." to get an explanation of why I ask this question.

"how(X)." to find out how confident I am in concluding X,
and how I reached that conclusion.

"rule(X)." to display all rules that conclude about X.

"halt." to terminate consultation.

"help." to print this message.

Is it true that pest_look(spider_like)

? n.

Is it true that pest_look(caterpillar_like)

? n.

Is it true that pest_look(bedbug_like)

? y.

Is it true that pest_color(_G1258)

? y.

Is it true that damage_on_plant(mines_on_leave)

? n.

Is it true that pest_look(fly_like)

? n.

Is it true that pest_look(beetle_like)

? n.

Is it true that damage_on_plant(hole_on_pot)

? n.

Is it true that damage_on_plant(foilage_destruction)

? y.

treatment(spray pesticide if the plant is at its early stage of growth.) was concluded with certainty 90.25

The proof is

treatment(spray pesticide if the plant is at its early stage of growth.) CF= -90.25 :-

insect(aphid) CF= -90.25 :-

pest(infecting) CF= -95 :-

pest_look(spider_like) CF= -100 was given by the user

pest_color(red) CF= 100 was given by the user

damage_on_plant(foilage_destruction) CF= 100 was given by the user

treatment(aphid,spray pesticide if the plant is at its early stage of growth.) CF= 100 was a fact in the knowledge base

true.

4 ?- solve.

Hello. Please answer my questions with one of the following:

"y.", which means "yes" (confidence value 100).

"n.", which means "no" (confidence value -100).

A number followed by a period. The number, which should be between -100 and 100, is your confidence in the truth of the query.

"why." to get an explanation of why I ask this question.

"how(X)." to find out how confident I am in concluding X,
and how I reached that conclusion.

"rule(X)." to display all rules that conclude about X.

"halt." to terminate consultation.

"help." to print this message.

Is it true that pest_look(spider_like)

? n.

Is it true that pest_look(caterpillar_like)

? n.

Is it true that pest_look.bedbug_like)

? n.

Is it true that pest_look(fly_like)

? y.

Is it true that pest_color(white)

? y.

Is it true that damage_on_plant(mines_on_leave)

? n

|: n.

ERROR: Stream user_input:0:218 Syntax error: Operator expected

Exception: (10) solve(insect(_G1217), _G1307, [rule((treatment(_G1206):-insect(_G1217), treatment(_G1217, _G1206)), 100)], 20) ? creep

Exception: (8) solve(treatment(_G1206), _G1307, [], 20) ? creep

Exception: (6) solve ? creep

5 ?- solve.

Hello. Please answer my questions with one of the following:

"y.", which means "yes" (confidence value 100).

"n.", which means "no" (confidence value -100).

A number followed by a period. The number, which should be between -100 and 100, is your confidence in the truth of the query.

"why." to get an explanation of why I ask this question.

"how(X)." to find out how confident I am in concluding X,
and how I reached that conclusion.

"rule(X)." to display all rules that conclude about X.

"halt." to terminate consultation.

"help." to print this message.

Is it true that pest_look(spider_like)

? n.

Is it true that pest_look(caterpillar_like)

? n.

Is it true that pest_look.bedbug_like)

? n.

Is it true that pest_look(fly_like)

? y.

Is it true that pest_color(white)

? y.

Is it true that damage_on_plant(mines_on_leave)

? n.

Is it true that pest_look(beetle_like)

? n.

Is it true that pest_color(black)

? n.

Is it true that damage_on_plant(hole_on_pot)

? n.

Is it true that damage_on_plant(foilage_destruction)

? y.

treatment(spray pesticide if the plant is at its early stage of growth.) was concluded with certainty 90.25

The proof is

treatment(spray pesticide if the plant is at its early stage of growth.) CF= -90.25 :-

insect(aphid) CF= -90.25 :-

pest(infecting) CF= -95 :-

pest_look(bedbug_like) CF= -100 was given by the user

pest_color(white) CF= 100 was given by the user

damage_on_plant(foilage_destruction) CF= 100 was given by the user

treatment(aphid,spray pesticide if the plant is at its early stage of growth.) CF= 100 was a fact in the knowledge base

true.

Appendix IV

Sample code from the expert system shell and the rule base

This section contains part of a code taken from the expert system shell.

```
solve :- solve(treatment(X),CF), nl.
```

```
solve :- retractall(know(_,_)),  
        write('There is insufficient knowledge to make a diagnosis'),  
        nl.
```

```
% The notation "solve/2" means "solve with 2 arguments".
```

```
% solve/2 succeeds with
```

```
% argument 1 bound to a goal proven true using the current knowledge base
```

```
% argument 2 bound to the confidence in that goal.
```

```
%
```

```
% solve/2 calls solve/4 with appropriate arguments. After solve/4 has completed,
```

```
% it writes the conclusions and prints a trace.
```

```
solve(Goal, CF) :-
```

```
    retractall(know(_,_)),
```

```
    print_instructions,
```

```
% solve/4 succeeds with
```

```
% argument 1 bound to a goal proven true using the current knowledge base
```

```
% argument 2 bound to the confidence in that goal.
```

```
% argument 3 bound to the current rule stack
```

```
% argument 4 bound to the threshold for pruning rules.
```

```
%
```

```

% solve/4 is the heart of exshell. In this version, I have gone back to the
% simpler version. It still has problems with negation, but I think that
% this is more a result of problems with the semantics of Stanford Certainty
% factors than a bug in the program.
% The pruning threshold will vary between 20 and -20, depending whether,
% we are trying to prove the current goal true or false.
% solve/4 handles conjunctive predicates, rules, user queries and negation.
% If a predicate cannot be solved using rules, it will call it as a PROLOG predicate.
, _, _) :-
% call(Goal).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Certainty factor predicates. Currently, these implement a variation of
% the MYCIN certainty factor algebra.
% The certainty algebra may be changed by modifying these predicates.

% negate_cf/2
% argument 1 is a certainty factor
% argument 2 is the negation of that certainty factor
negate_cf(CF, Negated_CF) :-
    Negated_CF is -1 * CF.

% and_cf/3
% arguments 1 & 2 are certainty factors of conjoined predicates
% argument 3 is the certainty factor of the conjunction
and_cf(A, B, A) :- A =< B.

```

and_cf(A, B, B) :- B < A.

%rule_cf/3

% argument 1 is the confidence factor given with a rule

% argument 2 is the confidence inferred for the premise

% argument 3 is the confidence inferred for the conclusion

rule_cf(CF_rule, CF_premise, CF) :-

CF is CF_rule * CF_premise/100.

%above_threshold/2

% argument 1 is a certainty factor

% argument 2 is a threshold for pruning

%

% If the threshold, T, is positive, assume we are trying to prove the goal

% true. Succeed if CF >= T.

% If T is negative, assume we are trying to prove the goal

% false. Succeed if CF <= T.

above_threshold(CF, T) :-

T >= 0, CF >= T.

above_threshold(CF, T) :-

T < 0, CF <= T.

%invert_threshold/2

% argument 1 is a threshold

% argument 2 is that threshold inverted to account for a negated goal.

%

```
% If we are trying to prove not(p), then we want to prove p false.  
% Consequently, we should prune proofs of p if they cannot prove it  
% false. This is the role of threshold inversion.
```

```
invert_threshold(Threshold, New_threshold) :-  
    New_threshold is -1 * Threshold.
```

This section presents the sample code from the knowledge base system. Sample code was taken from pepper disease and pepper insect knowledge bases.

```
rule((treatment(Advice) :-  
    (insect(X),treatment(X, Advice))), 100).
```

```
% rules to infer:
```

```
rule((insect(aphid) :-  
    (pest(infecting),damage_on_plant(foilage_destruction))),95).
```

```
rule((insect(helicovera_armigera) :-  
    (pest(infecting),damage_on_plant(hole_on_pot))),95).
```

```
rule((insect(white_fly) :-  
    (pest(infecting),damage_on_plant(mines_on_leaf))),95).
```

```
rule((insect(flee_beetle) :-  
    (pest(infecting),damage_on_plant(hole_on_leaf))),95).
```

```
rule((insect(red_spider_mite) :-  
  (pest(infecting),damage_on_plant(mines_on_leaf))),95).
```

```
rule((insect(leaf_miner) :-  
  (pest(infecting),damage_on_plant(mines_on_leaf))),95).
```

```
% Rules to infer.
```

```
rule((pest(infecting) :-  
  (pest_look(bedbug_like), pest_color(_))),95).
```

```
rule((pest(infecting) :-  
  (pest_look(caterpillar_like), pest_color(brown_with_ashy_stripe))),95).
```

```
rule((pest(infecting) :-  
  (pest_look(fly_like), pest_color(white))),95).
```

```
rule((pest(infecting) :-  
  (pest_look(fly_like), pest_color(white))),95).
```

```
rule((treatment(Advice) :-  
  (disease(X),treatment(X, Advice))), 100).
```

% rules to infer bad part of the affected plant:

```
rule((disease(powdery_mildew) :-  
  (bad_part(leaf),plant(withers))),95).
```

```
rule((disease(bacterial_spot) :-  
  (bad_part(stem),bad_part(leaf))),90).
```

```
rule((disease(virus) :-  
  (bad_part(leaf))),95).
```

```
rule((disease(bacterial_soft_rot) :-  
  (bad_part(stem),plant(rots))),95).
```

```
rule((disease(sclerotine_stem_rot) :-  
  (bad_part(stem),plant(withers))),95).
```

% Rules to infer system that failed.

```
rule((bad_part(leaf) :-  
  (upper_leaf(yellowish), lower_leaf(ashy))),95).
```

```
rule((bad_part(leaf) :-
```

(upper_leaf(spoty), lower_leaf(_)),95).

rule((bad_part(stem) :-

(stem(spoty), stem(_)),95).

rule((bad_part(stem) :-

(stem(_), stem(whitish))),95).

rule((bad_part(stem) :-

(stem(scarce), stem(_)),95).

rule((bad_part(stem) :-

(stem(whitish), stem(_)),95).

rule((bad_part(leaf) :-

(upper_leaf(ranger), lower_leaf(_)),95).