

GOVP 12001309

635.210285
↳ 2936

최	종
연구보고서	

인터넷 기반 실시간 감자역병관리 전문가시스템 개발

Development of Expert System of Potato Late Blight
Management based on Internet in Real Time

주관연구기관
고령지농업시험장

협동연구기관
승실대학교, 경희대학교

농 립 부



제 출 문

농림부 장관 귀하

본 보고서를 “인터넷 기반 실시간 감자역병관리 전문가시스템 개발” 과제의 최종보고서로 제출합니다.

1999. 10. .

주관연구기관명 : 고령지농업시험장

총괄연구책임자 : 신 관 용

연 구 원 : 함 영 일

연 구 원 : 안 재 훈

연 구 원 : 권 민

연 구 원 : 김 점 순

연 구 원 : 서 효 원

연 구 원 : 조 우 석

연 구 원 : 강 용 구

연 구 원 : 전 환 홍

연 구 원 : 권 영 기

연 구 원 : 김 응 종

협동연구기관명 : 숭실대학교

협동연구책임자 : 박 영 택

협동연구기관명 : 경희대학교

협동연구책임자 : 이 승 룡

요 약 문

I. 제 목

인터넷 기반 실시간 감자역병관리 전문가시스템 개발

II. 연구개발의 목적 및 중요성

감자역병은 저온다습한 기후조건에서 다발하는데, 이러한 온·습도 등 호적한 환경은 대관령 여름기후조건과 거의 일치한다. 이러한 역병은 거의 매년 발생하여 상당한 감수를 가져 오는 것으로 보고되어 있다. 강원고랭지에서는 감자역병 방제를 위하여 관행적으로 1작기에 10회 정도 약제를 살포하여 왔는데, 현재 병 초발생 이전에는 예찰시스템의 실용화에 의하여 3-4회 정도의 약제살포 횟수의 감소가 가능하게 되었으나 발병후에는 여전히 관행방제에 의하여 7-10일 간격으로 약제를 살포하고 있다.

약제살포를 최소한으로 줄이기 위해서는 해당지역의 기후자원의 정밀한 분석자료를 토대로 한 병의 정확한 조기에찰과 합리적인 방제대책 기술개발로 최적 방제스케줄을 제공할 때에 가능할 것이다. 이를 위해서는 기상·동 환경자료의 정보화에 병발생 및 진전도, 방제모형 개발이 필수적이다.

현재 국내에서 가동중인 감자역병 예찰정보체제는 대관령 주변의 일부

농장을 대상으로 하고 있으며 또한 병 초발생일 예찰에 한하여 수행되고 있어 종합적인 관리체제로서는 미흡한 실정이다. 해당지역 농업생태계를 고려한 최적 방제계획(spray schedule) 수립을 위해서는 지역환경에 정확한 예찰시스템에 의하여 약제살포를 최대한 늦추고, 감자에 발생하는 여러 병중에서 역병을 정확히 조기 진단하고, 병발생 확인 후에는 합리적인 병방제 스케줄에 의하여 방제를 실시하여야 할 것이다.

이러한 병방제 시스템을 위해서는 병 진단, 예찰 및 방제에 각종 모형 및 진단 시스템이 개발되고 이어서 실제 농업현장에서 필요로 하는 양질의 정보가 실시간(real-time)에 전달되어야 하는데 이를 위한 수단으로 최근에 급속하게 확산되고 있는 인터넷을 활용하여 병진단, 예찰 및 방제정보를 농업현장에 전달할 수 있는 방법을 개발코자 본 연구를 실시하였다. 즉 감자역병의 정확한 조기에찰과 방제대책 제공으로 약제살포를 최소한으로 줄이면서 경제적으로 손실이 없는 수준에서 역병을 관리하는 시스템작성이 본 연구의 목적이다.

Ⅲ. 연구개발 내용 및 범위

1. 연구개발 내용

본 연구과제에서 개발하고자 하는 '인터넷 기반 감자역병 전문가시스템'은 다음과 같다.

- 농민수준에서 병진단이 가능한 진단시스템을 작성
- 기상정보를 이용하여 감자재배 농장의 감자역병 초기발생 및 진전정도를 예측하고, 약제 살포시기, 약제종류 등 최적 방제수단을 컴퓨터에 의해서 결정

- 여기에서 얻어진 지식정보를 인터넷을 통하여 영농현장에 전달할 수 있는 시스템 작성

2. 연구개발의 범위

본 연구는 1997년 10월 27일부터 1999년 10월 26일 까지 2개년에 걸쳐서 수행되었으며, 다음과 같은 내용과 범위로 연차별 연구가 진행되었다.

구분	연구개발목표	연구개발 내용 및 범위
1차년도 (1997)	인터넷기반 감자 역병 예찰 및 방제 모형 전문가 시스템 설계 및 기초연구	<ul style="list-style-type: none"> - 역병 예찰모형 실용화 - 감자역병 예찰 및 방제모형 설계 - 감자병해충 분류체계 작성 - 감자역병 예찰을 위한 지식습득 방법론 연구 - 감자 역병 예찰 전문가시스템 설계 - 사용이 편리한 인터넷 사용자 및 전문가시스템 인터페이스 설계 - AWS로부터 raw-data 획득 - 감자 역병예찰 모형 및 전문가 시스템을 위한 멀티미디어 데이터베이스 구축
2차년도 (1998)	인터넷기반 감자 역병 예찰 및 방제 전문가 시스템 구현 및 시스템 통합	<ul style="list-style-type: none"> - 역병 방제모형 최적화 - 감자역병 예찰 및 방제시스템 개발 - 인터넷과 전문가시스템 모듈 인터페이스 구축 - 인터넷상에서 통합 시스템 구축

또한 본 과제는 3 개의 세부과제로 구성되어 있으며 각 과제의 핵심 연구개발 내용 및 범위는 다음과 같다.

구 분	연구개발 내용 및 범위
제1세부과제 (감자역병 방제모형개발 및 시스템 관리)	<ul style="list-style-type: none"> - 감자 생장 및 수량 평가 - 기존 역병예찰체계 평가 및 보완 - 종합적 감자역병 방제모형 정립 - 감자병해충 분류체계 작성 - 감자 역병관리 시스템 운영기술 확보
협동과제 (감자병해충진단 전문가 시스템 개발)	<ul style="list-style-type: none"> - 감자병해충진단을 위한 지식습득 방법론 연구 및 개발 - 고시 20년 실험자료 및 지식에 기반을 둔 전문가시스템 구축 - 감자역병 전문가 시스템을 위한 멀티미디어 인터페이스 개발 - 이병감자 화상DB 구축 및 활용 - 병 방제정보DB 구축 및 활용 - 적정 방제시기 결정 등에 전문가시스템 접목
협동과제 (인터넷기반 사용자 및 시스템인터페이스구축)	<ul style="list-style-type: none"> - 기상자료의 실시간 수집체계 구축 - 인터넷 기반 사용자 및 시스템 인터페이스 구축 - 감자역병예찰 전문가 시스템을 위한 데이터베이스 구축 - 인터넷상에서 역병 예찰 시스템 및 전문가시스템 통합 - 감자역병 예찰모델 구현(프로그래밍)

IV. 연구개발결과 및 활용에 대한 건의

1. 연구개발결과

가. 우리시험장에서 개발한 감자 역병예찰법 'BLITECAST'시스템과 '이동평균법'의 정확도를 비교하여 실용가능성을 타진하여 본 결과, BLITECAST'시스템에 비하여 '이동평균법'에서 적중율이 높은 것으로 나타났다. BLITECAST'시스템에서는 다소 조기에측되는 결과였다. 실제 시스템작성시 두 시스템 모두를 이용하였는데 'BLITECAST'시스템에서 먼저 조건을 만족할 시 주의보가 발령되고 그 이후 '이동평균법'에서 병발생 호 조건으로 경과될 시 경보를 발령하도록 하였다. 만약 동시에 두 조건을 만족할 시 곧바로 경보가 발령되어 약제방제에 들어가고 실제 병 발생 시부터는 병진전도 모형으로 시스템이 넘어가게 된다.

나. 감자 역병 진전도를 예측할 수 있는 모형작성을 위하여 1975년부터 1992년까지 18년간 대관령 고랭지 감자포장에서 발생한 역병 발생정도와 기상자료를 가공한 CSV와의 관계를 조사하여 독립변수의 적합성을 검정하고, logistic 모형과 Gompertz 모형을 비교하여 적정모형을 선발하였는데 logistic 함수가 감자 역병 예측모형으로 적합한 것으로 판정되었다.

다. 약제방제 수준별 병 진전속도와 기상조건을 정량화한 수치 ACSV 사이에서 다음과 같은 관계식을 얻어 이를 병 진전속도 parameter(b_1)로 이용하였다. $b_1 = 0.0088 \text{ ACSV} - 0.033$ ($R^2 = 0.976$). 추정된 절편 및 병 진전속도 parameter를 logistic 모형에 대입하여 다음과 같은 수미품종의 역병 진전도 모형식을 작성하였다. 병 진전정도 $X = 4 / (1 + 39.0 \cdot \exp((0.0088 \text{ ACSV} - 0.033) \cdot \text{CSV}))$.

라. 모형검증결과 1998년 1999년 모두에서 모형은 병진전도를 잘 설명하고 있다. 특히 1999년 무방제시에는 90%이상의 결정계수를 보이면서 실제

제 병진전정도를 모의하고 있다. 그러나 방제시 특히 방제수준이 높을 경우에는 고도의 유의성은 인정되나 정확도가 다소 낮아지는 경향이였다.

마. 감자수량감소 허용기준은 10%로 할 경우에는 수미는 CSV 12 에서, 조풍은 CSV 13에서 약제를 살포하여야 하며, 수량감소를 5%로 할 경우에는 두품종 모두에서 CSV 11간격으로 약제를 살포하는 것이 좋은 것으로 나타났다.

바. 이상의 결과를 종합하여 볼 때 역병 발생후 본 방제모형에 의하여 약제를 살포하면(SV 12 간격) 기존 관행방법에 비하여 약제살포 횟수를 2-3회 정도 줄일 수 있을 것으로 생각된다. 즉 본 연구결과를 종합하여 보면 역병 초발생 및 방제모형을 이용하면 역병 초발생전에 3-4회, 발생후에 2-3회 줄일 수 있을 것으로 확인되어 1작기당 관행 10회정도 살포를 4-6회 즉 50% 단축시킬 수 있으면서도 정상적인 수량이 가능할 것으로 나타났다.

사. 감자에 많이 발생하는 병해를 학술적인 측면과 병증상에 의한 분류로 나누어 감자 병해 분류체계를 작성하여, 제2과제에서 수행하는 컴퓨터 병해 진단시스템의 자료로 제공하였다.

아. 감자 병해충에 대한 지식을 지식베이스로 구축하고 감자 병해충을 추론하기 위한 추론모듈을 생성하였다. 특히 지식베이스구축에 있어 실제 감자 전문가들과의 지속적인 인터뷰를 통해 감자 병해충을 효과적으로 판단할 수 있는 지식을 획득하고 이를 지식베이스로 구축하였다. 감자 병해충 전문가시스템이 인터넷을 기반으로 설계되었기 때문에 농민들이 감자에 병증상이 있을 때 쉽게 집에서 진단 전문가시스템을 이용할 수 있다. 이런 결과로 그 동안 문제가 되어 왔던 농민들과 감자 전문가들 사이의 교류의 부족함을 해소할 수 있다. 또한 농민들이 전문가시스템을 쉽게 사용할 수 있도록 정지화상과 텍스트를 함께 이용함으로써 농민들의 편의성을 고려한 전

문가 시스템을 구축하였다.

자. 개발된 시스템은 WWW과 인터넷에 기반을 두고 있어 전용 단말기가 없어도 전화와 모뎀을 갖춘 일반 PC에 넷스케이프나 마이크로소프트사의 익스플로러와 같은 웹 브라우저 만을 갖추게 된다면 그 어느 곳에서라도 사용 가능하다. 그리고 인터페이스는 멀티미디어와 하이퍼미디어에 기반을 두고 있어 사용자들이 단순한 마우스 클릭만으로 원하는 감자 정보와 역병 예찰 정보를 얻을 수 있으며, 다양한 멀티미디어 데이터를 효율적으로 전달할 수 있다. 그리고, 강우량 및 온도와 습도 등과 같은 raw-data를 자동으로 입력하고 관리할 수 있는 AWS raw-data 입력 및 관리구도 개발하였다. 본 과제를 통해서 획득한 기술들은 클라이언트/서버, 인터넷, WWW, Java, JDBC, 멀티미디어 등의 기술과 다양한 시스템을 통합하는 기술 등이다. 본 과제의 연구 결과를 활용하기 위한 방안으로는 크게 실용적인 면과 연구적인 측면으로 나눌 수 있다. 실용적인 활용 면에서는 주관연구기관에서 변경된 AWS 정보의 수정을 거쳐 강원도 일대에 대한 감자 관련 정보 서비스를 웹 환경에서 일반 사용자들에게 제공해 줄 수 있다. 그리고 기존의 농업 관련 방송, 하이텔이나 천리안에서 제공하는 농업 정보 등과 연계하여 사용할 수 있다. 연구적인 활용 면에서는 본 과제에서 획득한 인터넷, WWW, 클라이언트/서버, Java, JDBC, 멀티미디어, 인공지능과 실시간 컴퓨팅의 통합, 시스템 통합, 그리고 객체지향형 DBMS 기술을 활용하여 향후 국가적 차원에서 농업 관련 정보 인프라 구축에 적극 응용할 수 있다.

2. 활용에 대한 건의

가. 본 연구의 결과가 실용적으로 넓게 활용되기 위해서는 정확한 기상 정보입수가 중요하다. 주요 영농단지 등에 기상관측장치의 설치되어야 할

것이다. 또한 현재의 기상청 관측망과 진흥청 운용망과의 연계통합 등이 이루어진다면 하여 다양한 기상관측정보가 농업정보로 활용될 수 있을 때에 본 시스템은 진가를 발휘할 수 있을 것이다.

나. 농민 농민단체 회사 등에서 본 시스템이 실용되기 위해서는 본 연구에서 개발된 감자역병 외 타작물, 여러 병해충에 대한 연구도 속행되어 종합적인 시스템으로 통합되어야 할 것이다

다. 본 연구의 목적은 수량이 경제적 허용범위를 벗어나지 않는 범위 내에서 줄어들지 않으면서 농약을 최소한 적게 살포하도록 하는 기술인데, 실제 농민에 대하여 농약 과다살포에 대한 규제 혹은 농약을 적게 살포하는 농산물에 대한 우수품질 인정제 등이 뒤따라야 안정된 농산물 생산이 가능하고 또한 농생태계도 보전 될 것이다.

SUMMARY

I . Development of Management Model for Control of Potato Late Blight

Agricultural weather is one of the most important factors controlling the potato farming at highlands in Kangwon province with respect to the crop productivity and the disease epidemic. To model for prediction and management potato late blight based on weather information from AWS and standard weather stations, theoretical equations for potato late blight were set up to formulate a simulation model. Both forecasting systems of potato late blight, 'BLITECAST' system and the 'moving average method' were compared with forecasting accuracy of late blight. Accuracy of the prediction by the two methods was higher in 'moving average method'. To model for prediction of potato late blight progress, the relationship between severity index of potato late blight transformed by the logit transformation function and cumulative severity value(CSV) processing weather data during growing period in Taegwallyeong alpine area, 1975 to 1992 were examined. Equation of progressive rate, $b_1 = 0.0088 \text{ ACSV} - 0.033$ ($R^2 = 0.976$), was written by examining the relationship between the parameters of progressive rate of late blight and the average CSV(ACSV) quantifying weather information. By estimating parameters of logistic function, model able to describe the late blight progress of potato, cv. Superior was formulated in $Y = 4/(1 + 39.0 \cdot \exp((0.0088 \text{ ACSV} - 0.033) \cdot \text{CSV}))$. Calculated potato late blight progress from the model were in good agreement with actual progress showing highly significant correlation coefficient values about 0.9. Proper

fungicide spray level of potato late blight within economic threshold was appeared in CSV 12. The management model of potato late blight described in this study could reduce the number of fungicide sprays least by 4-6 times from 10 times sprays during cropping season under the conventional program of fungicide sprays without losing disease control efficacy.

II. The development of the Diagnosing Expert System for Potato Disease

We constructed the knowledge base of potato diseases and implemented module inferencing potato diseases. Especially, at the process of constructing knowledge base, we used knowledges which were acquired on interview with the experts continuously. So, we could get more accurate and effective knowledges. Farmers will be able to use expert system at home because potato diseases expert system has been designed and based on internet. Expert system will help user to solve the problem of the communication with the potato experts. Also, farmers will be able to use the expert system more easily, because the system represents information with the image and text for diagnosis.

III. A Study on the Realtime Potato Management Expert System Using Internet

In this project, we developed a realtime potato management expert system using Internet where users can access the system easier than

ever before. It utilizes the WWW, the Java programming language to provide the 'state-of-the-art' user interfacing techniques, and AI-based potato disease forecast algorithms to produce results within "real-time" and in a more efficient way. On the Web, through a graphical browser such as Netscape or MS-explorer, users can activate a hyperlink, read multimedia traffic informations, or download a file within a single mouse click. We also built the raw-data AWS input and management tool to construct AWS information storage system.

In the course of developing this study, we have earned the various technology such as client/server, Internet, WWW, Java, JDBC and system integration technology.

The application areas for the developed system are can be categorized into two parts; practical-oriented application and a research-oriented application. For the former case, the developed system can be used as an actual farm information Web service to the "Kangwon" area. Also it can be incorporated into the existing commercial Internet services as well as other farm related information providers such as farm guidance system. For the later case, we have earned the core technology such as Internet, WWW, client/server, Java, JDBC, integration between real-time computing and AI, multimedia, system integration technology. Such technology can be applied to build farm related information system in the future.

CONTENTS

Chapter I . Introduction -----	21
Section 1. Objective and Contents-----	22
Section 2. Materials and methods-----	27
Chapter II. Development of management model for control of potato late blight -----	28
Section 1. Introduction -----	29
Section 2. Forecasting of potato late blight -----	31
1. Materials and methods -----	31
2. Forecasting result -----	32
Section 3. Modeling of control system of potato late blight-----	35
1. Materials and methods -----	35
3. Results and discussion -----	37
Section 4. Classification of potato pests -----	63
1. Academic classification -----	63
2. Key to potato disease -----	65
References -----	71
Chapter III. Potato disease diagnosis expert system implementation -	74
Section 1. Introduction -----	75
1. Goals and Importance -----	75
1st Research Goals -----	75
2nd. Research Importance -----	75
2. Research Contents and Range-----	76

Section 2. Diagnosis Knowledge for Potato Disease-----	78
1. Expert Knowledge Acquisition-----	78
1st. Diagnosis Knowledge Extraction -----	78
2nd. Rule-based Knowledge Base -----	78
3rd. Potato Disease Symptom Classification -----	80
4th. Diagnosis Knowledge Gatherer Implementation for Potato Disease -----	81
2. Knowledge Base for Inductive Learning -----	92
1st. Case-based Diagnosis Knowledge-----	92
2nd. Knowledge Acquisition Architecture-----	93
3rd. Knowledge Acquisition with Entropy Theory-----	94
4th. Decision-tree's Generation -----	98
Section 3. Potato Disease Diagnosis System-----	101
1. Diagnosis System Architecture-----	101
1st. Expert System Architecture for Potato Disease Diagnosis-----	102
2nd. Diagnosis Process for Potato Disease-----	106
2. Inference System for Potato Disease-----	106
1st. Inference Engine Organization for Potato Disease-----	106
2nd. Blackboard-based Inference Engine-----	110
3. Knowledge Base for Potato Disease-----	111
1st. Rule-based Knowledge Base-----	111
2nd. Knowledge Base Implementation Methodology -----	112
3rd. Knowledge Base Implementation-----	114
Section 4. Expert System Interface on the internet -----	122
1. Remote Multimedia Diagnosis System -----	122

1st. Remote Diagnosis System -----	122
2nd. Multimedia Interface-----	122
2. Expert System Architecture on the internet -----	124
1st. System Architecture on the internet -----	125
2nd. User Interface-----	125
Section 5. Expert System Implementation -----	126
1. Diagnosis Module for Potato Disease -----	126
1st. User Response Module on the internet-----	126
2nd. Inference Module for Potato Disease-----	127
3rd. Question module for Information Acquisition-----	128
2. User Interface Module-----	130
1st. Multimedia User Interface Module -----	131
2nd. Inference result Generation Module -----	135
3. Implementation result-----	136
Chapter IV. A Study on the Realtime Potato Management Expert System Using Internet	141
Section 1. Introduction	
1. Background	142
2. Objectiveness and Necessity	143
3. Contents and Scope	145
4. Method for Research Achievement	147
5. Importance of Research	148
Section 2. The Contents and Research Result	151
1. User and System Interface	151

2. Servlet Using WWW Technique	165
3. User Interface Using Java Swing	168
4. Database Connection to JDBC	173
5. AWS Information Storage System	175
Section 3. Achievement and Contribution of Research	180
1. Achievement of Research Objectiveness	180
2. Achievement of Evaluation Pointer of View	181
3. Contribution for Related Technology	182
Section 4. Application Plan for Research Results	183
1. Necessity of Additional Research	183
2. Application for Other Research	183
References	184

목 차

제1장 서 론	21
제1절 연구개발의 목적과 범위	22
1. 연구개발의 목적	22
2. 연구개발의 범위	23
제2절 연구개발 방법	27
제2장 감자역병 방제모형 개발 및 시스템 관리	28
제1절 서 설	29
제2절 감자역병 예찰	31
1. 재료 및 방법	31
2. 감자역병 예찰결과	32
제3절 감자역병 방제모형 정립	35
1. 재료 및 방법	35
가. 감자 생장 및 수량 평가	35
나. 감자역병 진전도 및 방제모형 작성	35
2. 결과 및 고찰	37
가. 감자역병 진전도모형 작성	37
나. 진전도 모형의 적합성 검정	45
다. 감자역병 방제모형 작성	54
라. 감자 역병관리 시스템 운영기술 확보	60
제4절 감자병해 분류체계	63
1. 학술적 분류	63
2. 병증상에 의한 분류	65
참고문헌	71

제3장	감자병해충진단 전문가 시스템 개발	74
제1절	서 설	75
1.	연구개발의 목표와 중요성	75
가.	연구개발의 목표	75
나.	연구개발의 중요성	75
2.	연구개발 내용 및 범위	76
제2절	감자 병해충 진단 지식	78
1.	전문가 지식 획득	78
가.	진단 지식 추출	78
나.	규칙 기반 지식베이스	78
다.	감자 병해충 증상 분류	80
라.	감자 병해충 진단지식 구축	81
2.	귀납적 기계학습을 이용한 진단지식 구축	92
가.	사례를 이용한 진단지식 구축	92
나.	지식습득 구조	93
다.	엔트로피 개념을 이용한 지식습득	94
라.	결정 트리 생성	98
제3절	감자 병해충 진단 시스템	101
1.	진단 시스템 구조	101
가.	감자 병해충 진단 전문가 시스템 구조	101
나.	감자 병해충 진단 과정	102
2.	감자 병해충 추론 시스템	106

가. 감자 병해충 추론기관 구성-----	106
나. 블랙보드기반 추론 -----	110
3. 감자 병해충 지식베이스 -----	111
가. 규칙기반 지식베이스 -----	111
나. 지식베이스 구축방법-----	112
다. 지식베이스 구축-----	114
제4절 인터넷에서의 전문가 시스템 인터페이스-----	122
1. 멀티미디어 원격 진단 시스템-----	122
가. 원격 진단 시스템-----	122
나. 멀티미디어 인터페이스 -----	122
2. 인터넷에서의 전문가 시스템 구조-----	124
가. 인터넷 기반 시스템 구조-----	124
나. 사용자 인터페이스 구조-----	125
제5절 전문가 시스템 구현-----	126
1. 감자 병해충 진단 모듈-----	126
가. 인터넷을 이용한 사용자 입력 모듈-----	126
나. 감자역병 추론 모듈-----	127
다. 정보 획득을 위한 질의 모듈 -----	128
2. 사용자 인터페이스 모듈-----	130
가. 멀티미디어 기반 사용자 인터페이스 모듈-----	131
나. 추론 결과 생성 모듈-----	135
3. 구현 결과-----	136
제4장 실시간 감자 역병 관리 전문가 시스템 -----	141

제1절 서론-----	142
1. 연구 배경	142
2. 연구 개발 목적 및 필요성	143
3. 연구 내용 및 범위	145
4. 연구 수행 방법	147
5. 연구 개발의 중요성	148
제2절 연구 개발 수행 내용 및 결과	151
1. 사용자 및 시스템 인터페이스	151
2. WWW 기술을 이용한 Servlet 모듈 개발	165
3. 실시간 감자 역병 예찰 시스템의 사용자 인터페이스	168
4. JDBC를 이용한 데이터베이스와의 연동	173
5. 본 과제에서 개발한 AWS 정보 입력기의 Application Version	175
제3절 연구 개발 목표 달성도 및 대외 기여도	180
1. 연구 개발 목표 달성도	180
2. 평가의 착안점에 따른 달성도	181
3. 관련분야의 기술발전예의 기여도	182
제4절 연구 개발 결과의 활용 계획	183
1. 추가 연구의 필요성	183
2. 타 연구에의 응용	183
참 고 문 헌	184

제1장 서론

제1장 서론

제1절 연구개발의 목적과 범위

1. 연구개발의 목적

감자생산에 중대한 피해를 끼치는 병으로 감자 역병(*Phytophthora infestans* < Mont. > De Bary)을 들 수 있다. 이 병은 저온다습한 기후조건에서 다발하는데, 온·습도 등 호적인 환경이 일정기간 경과되면 병원균은 급격히 증식, 전파, 대발생하여 감자수량에 치명적인 영향을 끼친다. 역병이 생육초기 발생시 상당한 감수를 가져 오는 것으로 보고되어 있는데 이러한 경우, 이병후 병진전의 호조건이 계속되면 2주 이내에 지상부가 완전 고사하여 수확이 불가능해 지는 경우도 있다.

강원고랭지에서는 감자역병 방제를 위하여 관행적으로 1작기에 10회 정도 약제를 살포하여 왔는데, 현재 병 초발생 이전에는 예찰시스템의 실용화에 의하여 3-4회 정도의 약제살포 횟수의 감소가 가능하게 되었으나 발병후에는 여전히 관행방제에 의하여 7-10일 간격으로 약제를 살포하고 있다.

경제적으로 손실이 적은 범위내에서 약제살포를 최소한으로 줄이는 종합적인 병해충 관리시스템의 작성을 위해서는 해당지역의 환경자원량의 정밀한 분석자료를 토대로 한 병해충의 정확한 조기에찰과 합리적인 방제대책 기술개발로 최적 방제스케줄을 제공할 때에 가능할 것이다. 이를 위해서는 기상 등 환경자료의 정보화에 이어 작물생장 및 병해충 각각의 모형 개발 및 이들의 통합시스템 작성이 필요한 실정이다.

현재 국내에서 가동중인 감자역병 예찰정보체제는 대관령 주변의 일부 농

장을 대상으로 하고 있으며 또한 병 초발생일 예찰에 한하여 수행되고 있어 종합적인 관리체제로서는 미흡한 실정이다. 해당지역 농업생태계를 고려한 최적 방제계획(spray schedule) 수립을 위해서는 지역환경에 정확한 예찰시스템에 의하여 약제살포를 최대한 늦추고, 감자에 발생하는 여러 병중에서 역병을 정확히 조기 진단하고, 병발생 확인 후에는 합리적인 병방제 스케줄에 의하여 방제를 실시하여야 할 것이다.

이러한 병방제 시스템을 위해서는 병 진단, 예찰 및 방제 고도의 의사결정능력을 토대로 한 각종 모형 및 진단 시스템이 개발되고 이어서 실제 농업현장에서 필요로 하는 양질의 정보가 실시간(real-time)에 전달되어야 한다. 이러한 정보를 농민들에게 시각적이고 실시간으로 전달하기 위한 수단 가운데 최근에 급속하게 확산되고 있는 인터넷 (특히, 웹)의 활용이 유망하다.

따라서, 본 연구과제에서 개발하고자 하는 '인터넷 기반 감자역병 전문가 시스템'은 ①농민수준에서 병진단이 가능한 진단시스템을 작성하고, ②기상 정보를 이용하여 감자재배 농장의 감자역병 초기발생 및 진전정도를 예측하고, 약제 살포시기, 약제종류 등 최적 방제수단을 컴퓨터에 의해서 결정하며 ③ 여기에서 얻어진 지식정보를 인터넷을 통하여 영농현장에 전달가능한 시스템 작성을 목적으로 연구를 수행되었다. 즉 감자역병의 정확한 조기에찰과 방제대책 제공으로 약제살포를 최소한으로 줄이면서 경제적으로 손실이 없는 수준에서 역병을 관리하는 시스템으로, 이 연구결과가 여타의 병해충 관리 시스템과 성공리에 통합되면 합리적인 수준에서 병해충을 관리하는 종합적인 병해충관리(IPM : Integrated Pest Management)체제의 일부분으로 자리매김 할 것이다.

2. 연구개발의 범위

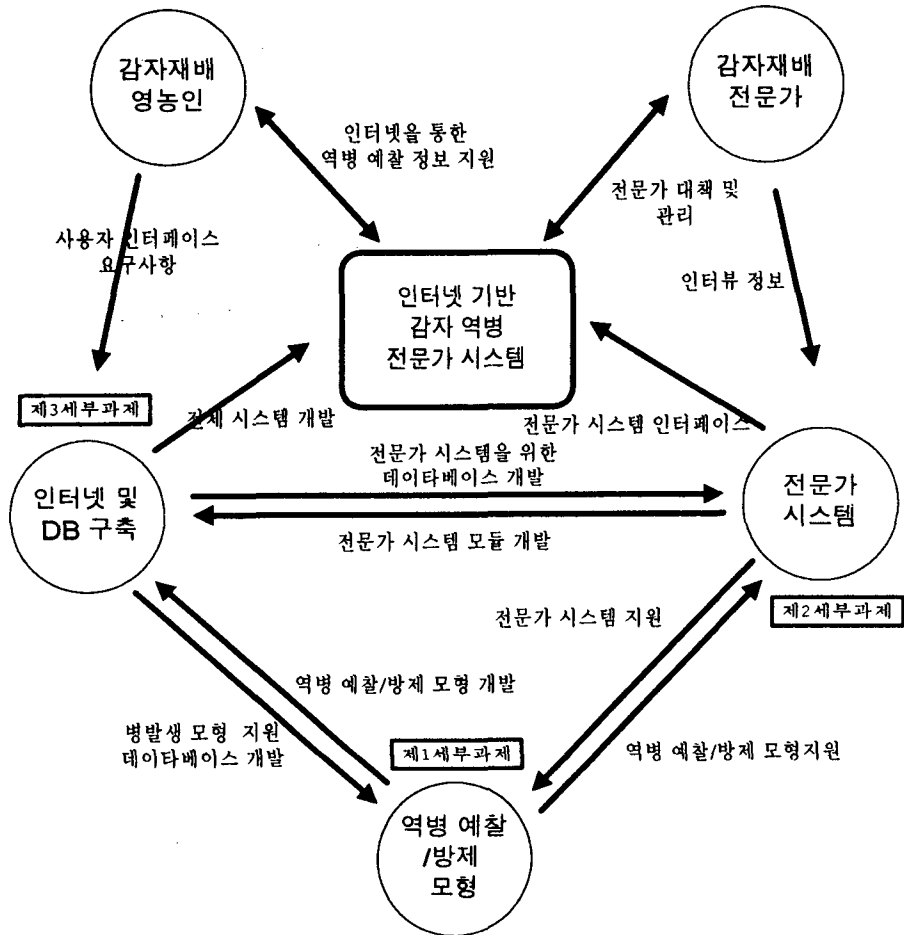
본 연구는 1997년 10월 27일부터 1999년 10월 26일 까지 2개년에 걸쳐서 수행되었으며, 다음과 같은 내용과 범위로 연차별 연구가 진행되었다.

구분	연구개발목표	연구개발 내용 및 범위
1차년도 (1997)	인터넷기반 감자 역병 예찰 및 방제 모형 전문가 시스템 설계 및 기초연구	<ul style="list-style-type: none"> - 역병 예찰모형 실용화 - 감자역병 예찰 및 방제모형 설계 - 감자병해충 분류체계 작성 - 감자역병 예찰을 위한 지식습득 방법론 연구 - 감자 역병 예찰 전문가시스템 설계 - 사용이 편리한 인터넷 사용자 및 전문가시스템 인터페이스 설계 - AWS로부터 raw-data 획득 - 감자 역병예찰 모형 및 전문가 시스템을 위한 멀티미디어 데이터베이스 구축
2차년도 (1998)	인터넷기반 감자 역병 예찰 및 방제 전문가 시스템 구현 및 시스템 통합	<ul style="list-style-type: none"> - 역병 방제모형 최적화 - 감자 역병 예찰 및 방제 전문가시스템 개발 - 인터넷과 전문가시스템 모듈 인터페이스 구축 - 인터넷상에서 통합 시스템 구축

또한 본 과제는 3 개의 세부과제로 구성되어 있으며 각 과제의 핵심 연구개발 내용 및 범위는 다음과 같다.

구 분	연구개발 내용 및 범위
제1세부과제 (감자역병 방제모형 개발 및 시스템 관리)	<ul style="list-style-type: none"> - 감자 생장 및 수량 평가 - 기존 역병예찰체계 평가 및 보완 - 종합적 감자역병 방제모형 정립 - 감자병해충 분류체계 작성 - 감자 역병관리 시스템 운영기술 확보
협동과제 (감자병해충진단 전문가 시스템 개발)	<ul style="list-style-type: none"> - 감자병해충진단을 위한 지식습득 방법론 연구 및 개발 - 고시 20년 실험자료 및 지식에 기반을 둔 전문가시스템 구축 - 감자역병 전문가 시스템을 위한 멀티미디어 인터페이스 개발 - 이병감자 화상DB 구축 및 활용 - 병 방제정보DB 구축 및 활용 - 적정 방제시기 결정 등에 전문가시스템 접목
협동과제 (인터넷기반 사용자 및 시스템인터페이스구축)	<ul style="list-style-type: none"> - 기상자료의 실시간 수집체계 구축 - 인터넷 기반 사용자 및 시스템 인터페이스 구축 - 감자역병예찰 전문가 시스템을 위한 데이터베이스 구축 - 인터넷상에서 역병 예찰 시스템 및 전문가시스템 통합 - 감자역병 예찰모델 구현(프로그래밍)

한편, 각 세부과제간의 연관 관계는 다음과 같다.



제2절 연구개발 방법

- ▶ 개발업무의 전문성에 따라 감자역병 예찰 및 방제모형 작성 등 역병생태 연구, 병해충 진단전문가시스템 연구, 인터넷 구축 연구 등 3개 세부과제로 나누어 추진하되 최종적인 결합 및 시작품 운영은 인터넷 구축과제에서 담당
- ▶ 시작품 개발에 허용된 기간내에 전체시스템을 고령지 농업시험장에 설치하여 감자재배기간 중 실제 운영을 통해 미비점 보완 등, 그 실용성을 확보

제2장 감자역병 방제모형 개발 및 시스템 관리

제2장 감자역병 방제모형 개발 및 시스템 관리

제1절 서 설

감자생산에 있어서 중요한 제한요인은 바이러스와 역병이다. 이중 감자역병은 일정한 기상환경 조건에 의하여 규칙적으로 발생 및 진전이 이루어진다. 이러한 규칙성은 감자역병의 초발생 및 진전도를 예측할 수 있는 수리적 모형작성을 가능케하였고 현재 선진외국에서는 컴퓨터시스템에 의하여 초발생 예찰 및 병방제 스케줄이 작성되어 병 방제에 실용되고 있으며 이는 관행적인 약제살포에 비하여 안전하게 약제살포 횟수를 수회 경감시킬 수 있는 것으로 보고되고 있다.

이러한 감자역병 예찰을 위하여 고령지농업시험장에서는 1975부터 '이동그래프법'에 의하여 감자역병 초발생일 예측을 실시하여 왔다. 매일매일의 온도와 습도를 이용하여 초발생일 계산하는데 대부분의 해에서 병발생 전제조건인 상대습도 조건을 만족하지 못하여 예보정확도가 낮았다. 1994년부터는 이동그래프법에 비하여 예측정도가 높은 'BLITECAST'시스템에 의한 예찰도 함께 실시하여 농민 및 감자 유관기관에 통보해 오고 있다. 그러나 'BLITECAST'시스템은 비교적 정확한 예찰은 가능하나 필요로 하는 기상정보의 입수에 다소의 어려움이 있었다. 이러한 문제를 해결하기 위하여 1998년 고령지농업시험장에서는 기존에 실시하여온 감자역병 발생예찰방법인 '이동그래프법'과 'BLITECAST'시스템을 조합 변형시킨 '이동평균법'을 개발하였다. '이동평균법'은 예찰방법이 간단하면서 정확도가 높아 기상관측소의 일반 기상정보로도 누구나 손쉽게 감자역병 예찰이 가능하다.

병의 초발생일 예찰에 이어서 방제스케줄 작성시 중요한 전제조건은 병진도를 예측할 수 있는 모형을 작성하여야 한다. 감자 역병의 이상적인 진전

도 곡선은 sigmoid 형태를 나타내는데 병의 진전도를 예측할 수 있는 모형식 작성을 위해서는 일반적으로 병의 진전정도에 영향을 미치는 환경조건을 독립변수로 하여 병의 진전도를 설명할 수 있는 적합한 모형선발에 이어 품종(품종)저항성, 약제방제 수준 등에 따라 변하는 진전속도를 표현할 수 있는 parameter를 추정하게 된다.

본 연구는 감자 역병관리 시스템 작성을 위하여, 우선 감자역병을 쉽게 동정할 수 있는 컴퓨터 시스템 개발을 위하여 수행되고 있는 협동연구과제 '병해충 진단 전문가 시스템 개발'에 필요한 감자병해 분류체계를 작성하고, 감자역병 초발생일 예측시스템인 'BLITECAST'시스템과 '이동평균법'을 검증하고, 또한 병진전도 모형을 활용하여 적정 병방제스케줄을 작성하여 최종적으로 인터넷을 기반으로 하는 감자역병 예찰 및 방제 전문가시스템 구축에 필요한 수리적인 모형을 작성하는데 목적이 있다.

제2절 감자역병 예찰

1. 재료 및 방법

대관령에 위치한 고령지농업시험장(대관령, 해발 800 m)에서 기존에 실시하여온 감자역병 발생 예찰방법인 'BLITECAST'시스템과 '이동평균법'을 함께 이용하여 1998년도와 1999년도 2개년의 역병 초발생을 예찰하였다. 예찰방법은 표1에 나타나 있는 바와 같다.

공시품종은 현재 전국적으로 재배면적이 가장 많고 역병에 감수성인 수미를 주 품종으로 하고 역병저항성인 조풍을 추가하여 강릉과 대관령 지역의 감자포장의 역병 초발생일을 조사하였다.

표1. 감자역병 예찰방법

구 분	BLITECAST시스템(기존)	이동평균법(개선)
예찰조건	- 일일 상대습도 90%이상 지속 시간과 평균기온과의 관계를 등급화한 임의의 수치 severity value(SV) (1~4) 이용	1. 일일 평균기온의 7일이동 평균치가 12 ℃ 이상 2. 상대습도 5일 이동평균치 80% 이상
방법	- 출아후 부터 일일단위로 합산, 적산SV가 18이상인 되는 날 이후 7-14일 사이에 발병할 것으로 예보	- 상기 2조건 7일간 동시·연속 만족시 예보 - 예보일 7-14일 후에 병발생할 것으로 함

2. 감자역병 예찰결과

1998년은 'BLITECAST'시스템에서는 6월 12일 적산SV가 19가 되어 6월 19일부터 6월 25일 사이에 역병이 발생할 것으로 예측하였고 '이동평균법'에서는 상대습도를 5일간 이동평균한 수치가 80%이상, 7일이동 평균온도 12℃이상 두 조건을 동시에 7일 연속조건을 만족하는 날이 6월 18일이어서 6월 25일 부터 7월 1일 사이에 초발생할 것으로 예측하였는데 실제 6월 27일에 역병이 초발생하여 되어 'BLITECAST'시스템에서는 2일 조기에 예측되었고 '이동평균법'에서는 적중하였다(표2). 강릉 영동지방에서는 1999에는 'BLITECAST'시스템에서는 6월 29일 부터 7월 5일 사이에 역병이 발생할 것으로 예측하였고 '이동평균법'에서는 7월 3일 부터 7월 9일 사이에 초발생할 것으로 예측하였는데 실제 7월 5일에 역병이 초발생하여 두 시스템 모두에서 적중하였다(표3).

이상의 두가지 방법중 'BLITECAST'시스템에 비하여 '이동평균법'에서 적중율이 높은 것으로 나타났는데 'BLITECAST'시스템에서는 다소 조기에 예측되는 결과를 볼 수 있었다. 실제시스템 작성 시 두 시스템 모두를 이용하였는데 'BLITECAST'시스템에서 먼저 조건을 만족할 시 주의보가 발령되고 그 이후 '이동평균법'에서 병발생 호조건으로 경과될 시 주의보를 발령하도록 하였다. 만약 동시에 두 조건을 만족할 시 곧바로 경보가 발령되어 약제방제에 들어가고 실제 병 발생 시부터는 병진전도 모형으로 시스템이 넘어가게 된다.

표2. 1998년 예찰결과

(대관령 : 수미, 남작품종)

날 짜 (월.일)	BLITECAST system					이동평균법			
	평균 기온 (℃)	상대습도 90%이상 지속시간	SV value (0-4)	적산 SV	예찰	7일이동 평균기온 (℃)	상대 습도 (%)	상대습도 5일이동 평균(%)	예찰
6. 1	15.8	0	0				58		
6. 2	11.2	18					94		
6. 3	9.4	22					95		
6. 4	7.3	24					95		
6. 5	7.8	24	4	4			95	87.4	
6. 6	8.0	24	4	8			94	94.6	
6. 7	11.2	18.5	1	9		10.1	81	92.0	
6. 8	13.6	13	1	10		9.8	82	89.4	
6. 9	13.9	14.5	1	11		10.2	87	87.8	
6.10	12.4	24	4	15		10.6	94	87.6	
6.11	14.5	11	0	15		11.6	87	86.2	
6.12	13.7	24	4	19		12.5	94	88.8	
6.13	12.3	24	4	23	예보	13.1	95	91.4	
6.14	10.4	24	4	27		13.0	94	92.8	
6.15	12.7					12.8	89	91.8	
6.16	16.2					13.2	80	90.4	
6.17	14.4					13.5	94	90.4	
6.18	15.3					13.6	93	90.0	
6.19	16.2				초	13.9	87	88.6	예보
6.20	15.8				발	14.4	91	89.0	
6.21	13.8				생	14.9	96	92.2	
6.22	12.0				예	14.8	94	92.2	
6.23	12.2				측	14.2	92	92.0	
6.24	13.3				기	14.1	95	93.6	
6.25	16.6				간	14.3	95	94.4	
6.26	18.4					14.6	89	93.0	초
6.27	18.9	병 초발생				15.0	77	89.6	발
6.28	20.0					15.9	84	88.0	생
6.29	21.2					17.2	84	85.8	예
6.30	21.0					18.5	87	84.2	측
7. 1									기

표3. 1999년 예찰결과

(대관령 : 수미, 남작)

날 짜 (월.일)	BLITECAST system					이동평균법			
	평균 기온 (℃)	상대습도 90%이상 지속시간	SV value (0-4)	적산 SV	예찰	7일이동 평균기온 (℃)	상대 습도 (%)	상대습도 5일이동 평균(%)	예찰
6. 1	15.5	9	0	0			83		
2	15.2	18	2	2			82		
3	14.9	13	1	3			80		
4	15.9	10	1	4			77		
5	14.9	22	4	8			92	82.8	
6	17.2	14	2	10			84	83.0	
7	14.6	6	0	10		15.5	76	81.8	
8	16.7	8	0	10		15.6	59	77.6	
9	13.6	7	0	10		15.4	78	77.8	
10	17.5	4	0	10		15.8	69	73.2	
11	18.3	0	0	10		16.1	60	68.4	
12	19.6	0	0	10		16.8	58	64.8	
13	20.9	0	0	10		17.3	59	64.8	
14	21.2	9	0	10		18.3	60	61.2	
15	20.7	0	0	10		18.8	44	56.2	
16	14.2	3	0	10		18.9	84	61.0	
17	13.9	21	3	13		18.4	89	67.2	
18	16.5	12	1	14		18.1	70	69.4	
19	15.1	8	0	14		17.5	82	73.8	
20	15.1	12	1	15		16.7	75	80.0	
21	14.2	12	0	15		15.7	92	81.6	
22	13.8	20	3	18		14.7	88	81.4	
23	12.8	24	4	22	예보	14.5	94	86.2	
24	16.6	22	4	26		14.9	78	85.4	
25	18.5	6	0	26		15.2	70	84.4	
26	19.8	9	0	26		15.8	72	80.4	
27	21.0	13	2	28		16.7	66	76.0	예보
28	19.8	3	0	28		17.5	73	71.8	
29	15.4	9	0	28	병발생	17.7	93	74.8	
30	14.7	24	4	32	예측기간	18.0	94	79.6	
7. 1	16.0	23					92		
2	14.8	24					94		
3	14.9	24					93		
4	15.1	21					91		
5	14.8	14	병발생				89		
6	15.1	9					83		
7	14.5	19					85		
8	15.5	20					93		
9	15.5	24					94		

제3절 감자역병 방제모형 정립

1. 재료 및 방법

가. 감자생장 및 수량 평가

역병예찰에 기본이 되는 생장 및 수량을 조사하기 위하여 포장실증실험을 실시하였다. 공시품종은 수미와 조풍으로 하고, 파종기는 1998년 강릉 3월 중, 4월상, 4월중순경에, 대관령에서는 4월하, 5월상, 5월중하순경으로, 품종별, 재배지역별 난괴법 3반복으로 시험구를 배치하였다. 파종후 생육 및 수량, 괴경형성기 등을 조사하여 생체정보로 하였다. 재식거리는 휴간 75cm, 주간 25cm로 하였고, 시비는 N - P₂O₅ - K₂O = 15-18-12 kg/10a 를 전량 기비로, 퇴비는 1.5ton/10a을 사용하였다. 약제방제는 관행방제에 준하였는데 감자 출아 후부터 병발생 전에는 보호살균제 다이센 M45와 타코닐을, 병 발생 후에는 치료살균제(커지엠, 포름D)를 7-10간격으로 교대로 살포하였다.

나. 감자역병 진전도 및 방제모형 작성

1998년과 1999년 2개년에 걸쳐서 수미와 조풍을 공시하여 강릉에서는 3월 중하순에, 대관령고냉지는 5월상순에 파종하여 시험을 실시하였다. 처리내용은 표 4에 나타나 있는 것과 같이 약제살포 간격을 달리 하여 품종별, 재배지역별 난괴법 3반복으로 하였다. 감자 생육 및 수량과 병 진전상황을 조사하여 생체정보로 하고 기상환경을 측정하여 환경정보로 하여 모형을 작성한다. 재식거리는 휴간 75cm, 주간 25cm로 하였고, 시비는 N - P₂O₅ - K₂O = 15-18-12 kg/10a 를 전량기비로, 퇴비는 1.5ton/10a을 사용하였다. 시험구 배치의 지역, 품종별 난괴법 3반복으로 하였다. 약제방제는 감자 출아 후부

터 병발생 전에는 보호살균제 다이센 M45와 타코닐을, 병 발생 후에는 치료살균제(커지엠, 포룸D)를 교대로 살포하였다. 기상자료는 포장에 설치된 AWS(CR10)자료를 바탕으로 하고, 실험포장과 1km 정도 떨어진 대관령관측소의 자료를 함께 이용하였는데 상대습도는 자기습도계로 측정하여 상대습도 90%이상 지속시간을 1일단위로 합산하였다. 병 진전정도는 품종별 방제구 및 무방제구의 역병진전정도를 조사하는데 역병조사기준(James, C. W. A manual of assessment for plant disease. Canada Department of Agriculture Publication No.1458)에 의하였다.

표4. 농약살포 간격 (Severity value : SV)과 방법

약제살포 수준 Severity value(SV)간격	비 고
무방제(대조구) SV 4 SV 8 SV 12 SV 16	- 병발생전 보호제(다이센, 타코닐) - 병발생후 치료제(커지엠, 포룸D) 교대살포

모형화 추진체계는 우선 이론식을 작성하여 모형형태 등 골격을 결정하고 병 방제 수준에 따른 병진전도를 조사하여 이들과 기상요인 간의 관계해석을 거쳐서 파라메타를 추정하여 병 발생 및 진전정도 모형식을 작성하였다. 병 발생수준에 따른 수량변화 예측모형을 작성하여 피해허용수준을 결정하고 생육시기별 병진전정도를 정량화하여 이들이 수량감소에 미치는 영향을 조사, 모형화하고 최종적으로 이들을 통합하여 최적 방제스케줄을 작성하였다.

2. 결과 및 고찰

가. 감자역병 진전도모형 작성

(1) 감자역병 진전도 모형

(가) 이론식

- logistic 모형과 Gompertz 모형을 비교
- 전제조건은 역병 전염원은 감자포장에 항상 분포되어 있고, 역병 진전은 기상환경 조건에 의하여 달라짐

logistic 모형에서 시간 t 에 따라서 변화하는 병진전정도 X 는

$$X = X_f / (1 + \exp(-(b_0 + b_1 \cdot t))) \text{----- (1)}$$

(2)식을 변환하면 다음과 같이 된다.

$$(X_f - X) / X = \exp(-(b_0 + b_1 \cdot t)) \text{----- (2)}$$

(3)식에 자연대수를 취하여 logit 변환함수 $\ln(X / (X_f - X))$ 로 바꾸면

$$\ln(X / (X_f - X)) = b_0 + b_1 \cdot t \text{----- (3)}$$

Gompertz 모형에서 시간 t 에 따라서 변화하는 병진전정도 X 는

$$X = \exp(-b_0 (\exp(b_1 \cdot t))) \text{----- (4)}$$

b_0 와 b_1 는 파라메타이다. (4)식은 다음과 같이 1차함수로 변환될 수 있다.

$$-\ln(-\ln(X)) = -\ln(b_0) + b_1 \cdot t \text{----- (5)}$$

(나) 적정 모형선발

㉠ 일반적인 병 진전도

감자 역병의 이상적인 진전도 곡선은 sigmoid 형태를 나타낸다(그림 1) 그러나 일반적으로 품종, 약제방제 수준, 재배방법 등에 다르게 나타나는데 병의 진전도를 예측할 수 있는 모형식 작성을 위해서는 일반적으로 병의 진전정도에 영향을 미치는 환경조건을 독립변수로 하여 병의 진전도를 설명할 수 있는 적합한 모형선발을 하게 된다.

환경-병 진전도 모형작성시 연차간의 변이를 줄이기 위하여 누적일수 대신에 병 진전에 영향을 미치는 환경요소를 가공한 수치를 설명변수로 이용하는 경우가 많다. 본 연구에서는 출현 후부터의 누적일수 대신에 Severity value(표5)를 적산한 CSV를 설명변수로 이용하였는데 그때의 일반적인 병진전도 곡선은 그림1과 같다.

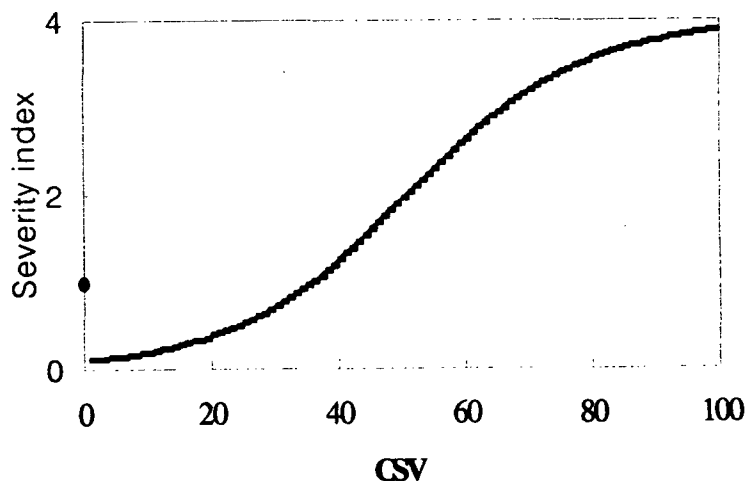


그림1. 적산 SV(CSV)에 따른 감자역병의 이상적인 진전도의 곡선

표5. Severity value(SV) 계산

평균기온 (°C)	Severity value			
	1	2	3	4
	상대습도 90%이상 지속시간			
7.2-11.6	16-18	19-21	22-24	24
11.7-15.0	13-15	16-18	19-21	22+
15.1-26.6	10-12	13-15	16-18	19+

㉠ 감자역병 진전도

고령지농업시험장의 과거 연구자료 1975년부터 1992년까지 18년간의 대관령 고랭지 감자포장에서 발생한 역병 발생정도를 그림2에 나타내었다. 심한 변이를 보이고 있다.

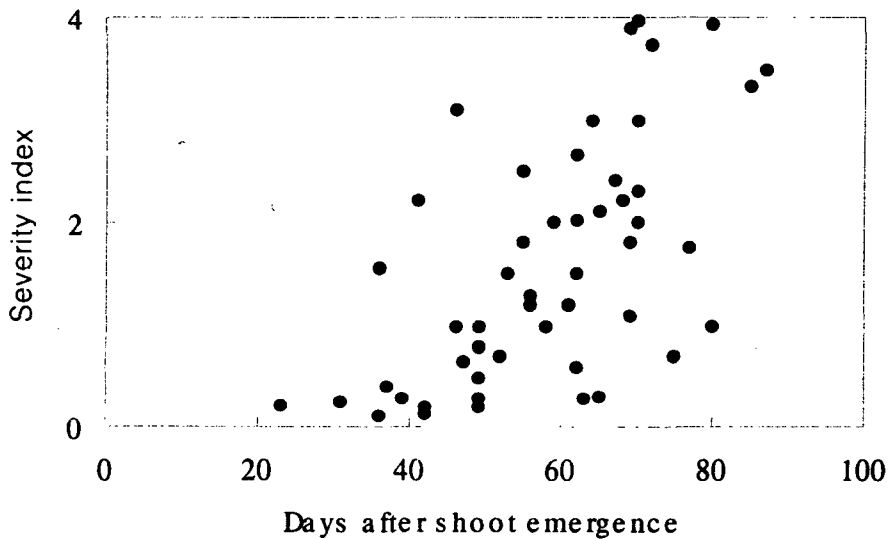


그림2. 감자역병 발생정도(1975년~1992년, 대관령고랭지)

㉔ CSV의 적합성 검토

출현후부터의 누적일수 대신에 CSV를 설명변수로 이용하였을 때 그림 2 에서 보이는 연차간의 변이를 어느 정도 좁히면서 초기 완만한 증가를 보 이다가 CSV 40 정도 이후부터 급격히 증가하는 sigmoid 형태를 나타내고 있다(그림3).

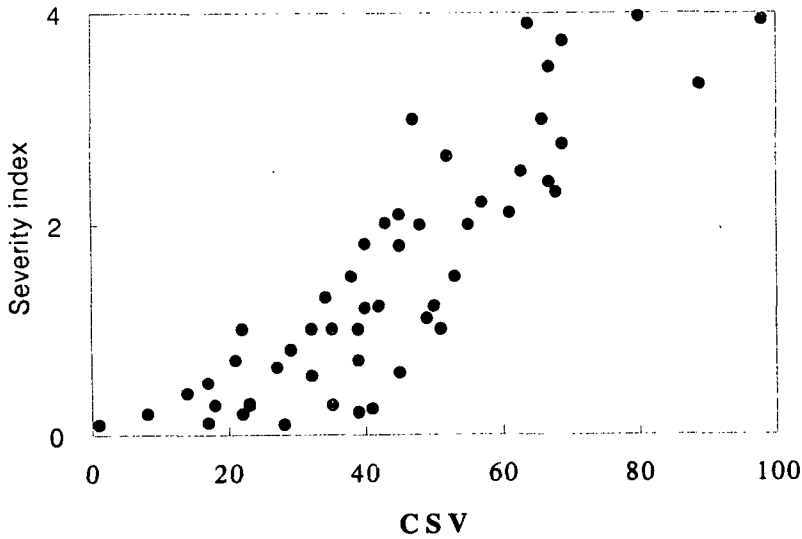


그림3. CSV와 병발생량과의 관계

㉕ 모형선발

그림1의 병진전도를 logit 변환함수 $\ln(X/X_r - X)$ (3식)와 Gompit 변환함수 $-\ln(-\ln(X))$ (5식)를 이용하여 변환시켜서 CSV와의 관계를 조사하여 보았다. 두 변환함수 모두에서 고도의 상관관을 보였으나 결정계수는 Gompit변환함수 0.680 보다 logit변환함수에서 0.742로 높게 나타났다. 역병 진전도 예측을 위해서는 Logistic 모형이 적합한 것으로 판정되었다.

그림4의 logit 변환함수의 회귀식에 나타난 기울기와 절편을 (1)식에 대입하여 다음과 같은 관계식을 얻을 수 있었다.

$$\text{병 발생량 } X = 4 / (1 + 39.0 \cdot \exp(-0.079 \cdot \text{CSV})) \text{ ----- (6)}$$

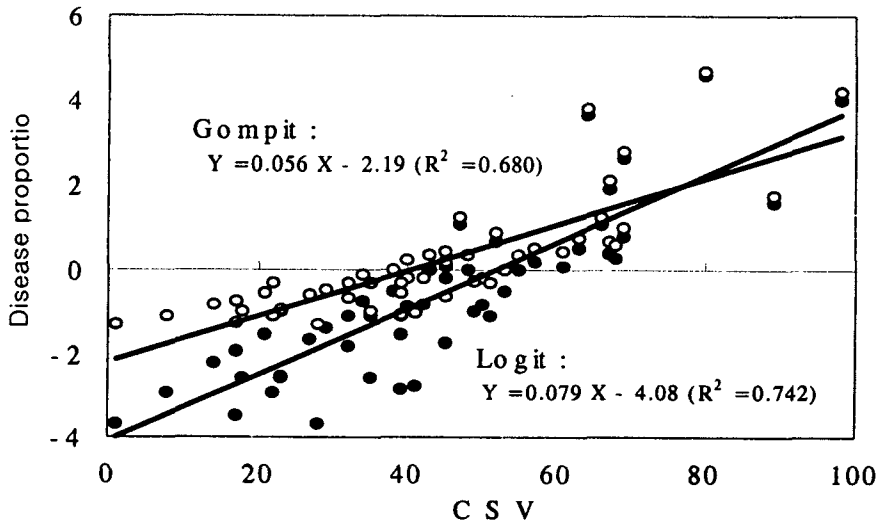


그림4. Logistic 모형(3식:Gompit)과 Gompertz 모형(5식:logit)을 병발생량에 fitting한 결과

㊸ Parameter 추정

병진전도에 영향을 미치는 기주(품종)저항성, 약제방제 수준, 재배방법 등을 설명할 수 있는 적절한 parameter를 추정하는 것이 중요하다. 1993년도 감자 재배기간 중의 CSV를 독립변수로 하여 무방제시의 병진전속도를 조사한 결과가 그림 5에 나타나 있다. 병진전속도를 나타내는 곡선의 기울기는 남작과 수미는 각각 0.137, 0.136로 거의 비슷하였는데 조풍은 0.070으로 타 품종에 비하여 기울기가 완만한 것으로 나타났다.

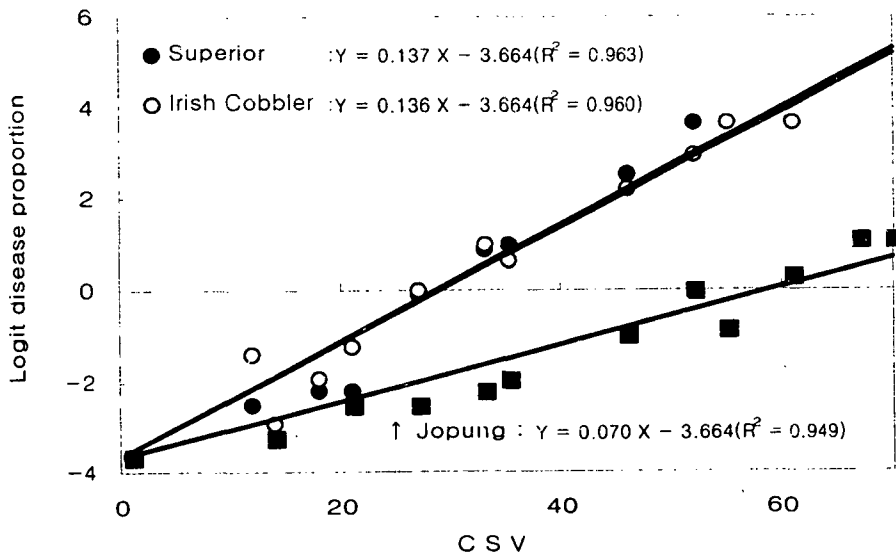


그림5. CSV 독립변수에 대한 무방제시의 감자품종별 병 진전속도

(2) 역병 방제모형 작성

방제수준이 다를 때 병 진전속도의 변화를 알아보았다(표6). 본 연구에서는 약제 살포간격을 나타낼 때 흔히 사용되는 일수대신에 그 기간동안의 기상상태를 수치화한 CSV를 기준으로 하여 병 진전속도를 조사하였다. 즉 재배기간 동안의 CSV를 총 약제살포 횟수로 나눈 수치인 평균 CSV(ACSV)를 이용하여 병 진전속도와의 관계를 조사하였다. 병 진전속도(b_1)와 ACSV와의 관계를 조사하여 기상조건에 의하여 병 진전속도를 예측할 수 있는 다음과 같은 관계식을 얻었다.

$$b_1 = 0.0088 \text{ ACSV} - 0.033 \quad (R^2 = 0.976) \quad \text{-----} \quad (7)$$

표6. 품종과 약제살포 수준에 따른 병진전 속도 (1993, 1997년)

Cultivar	Fungicide spray interval	No of Application times	ACSV ^z	Progressive rate	Coefficients of determination (R ²)
Superior	Unsprayed	0	-	0.137	0.963
	14-days	5	13.4	0.083	0.860
	SAP1 ^y	8	8.8	0.049	0.907
	7-days	9	7.4	0.034	0.774
	SAP2 ^x	10	6.7	0.020	0.998
Jopung	Unsprayed	0	-	0.070	0.949
	14-days	5	13.4	-	-
	SAP1	8	8.8	0.023	0.810
	7-days	9	7.4	-	-
	SAP2	10	6.7	-	-

z : Average CSV per application interval(total/number of application times).

y : Sprayed in June 14, June 20, June 26, July 1, July 9, July 15, July 20, July 26, 1993.

x : Sprayed in June 12, June 21, June 27, July 2, July 7, July 14, July 19, July 24, Aug. 5, Aug. 14, 1997.

Logistic 함수의 이론모형식 (1)의 절편 parameter b_0 는 (3)식의 $\ln(X_i/(X_r - X_i))$ 로 구할 수 있는데, 모형의 반응변수의 두 범주 즉, X의 최고치 X_r 가 '4', X의 초기치 X_i 가 '0.1'로 정해져 있으므로 b_0 는 -3.664로 계산되었다. 구하여진 절편상수와 병진전속도를 나타내는 (7)식을 (6)식에 대입하여 감자 수미품종의 역병진전도를 예측할 수 있는 모형식을 작성하였다.

$$\text{병진전도 } X = 4 / (1 + 39.0 \cdot \exp(-(0.0088 \text{ ACSV} - 0.033) \cdot \text{CSV})) \quad \text{--- (8)}$$

이상의 연구결과로 감자 역병진전도를 예측할 수 있는 모형식이 작성되어 합리적인 계획방제 스케줄 작성이 가능하게 되었다. 즉 예측시스템에 의한 초발생예찰에 이어, 초발생시에는 작물의 생육단계, 경제적인 피해허용

범위 등을 고려하여 사전에 계획된 병 방제수준에 따라서 ACSV가 결정되고 자동기상관측장치에 의하여 측정되는 실시간 기상자료를 분석하여 CSV를 계산, 적절한 방제시기를 결정하면 될 것으로 생각된다.

방제모형식을 바탕으로 방제수준에 따른 병진전도를 예상할 수 있는 결과를 모의 하여본 결과 다음과 같은 결과를 얻을 수 있었다. 즉 이어지는 연구인 피해허용수준 결정에 따라서 정해지겠지만 수량에 큰 피해가 없는 범위 내에서의 적정 약제살포 수준결정이 가능하게 되었다. 가령 25~50 % 정도까지 역병에 이병되어도 수량에 큰 피해가 없다면 SV 10정도를 기준으로 약제를 살포하면 될 것이고, 10 ~ 25%가 허용수준이라면 초기에는 SV 10 후기에 SV 6으로 역병을 관리하면 될 것이다.

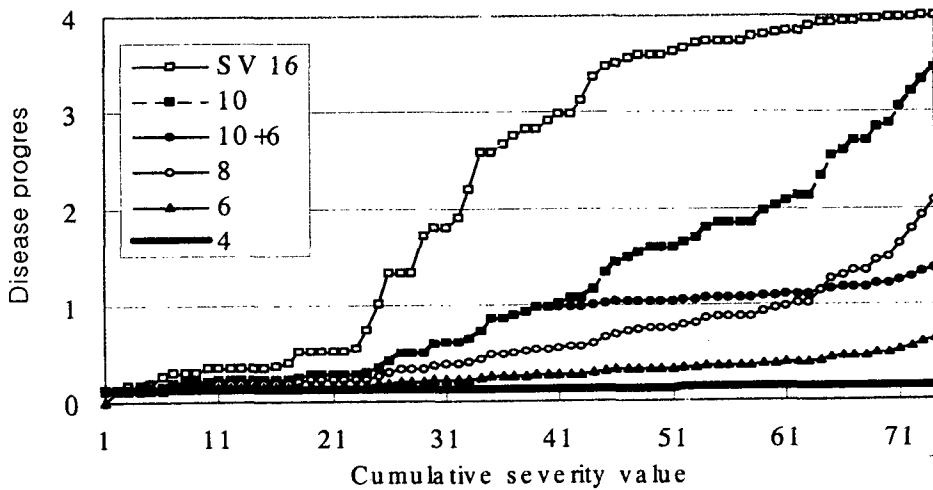


그림6. 약제살포 간격(SV의 차이)에 따른 병진전도 시뮬레이션

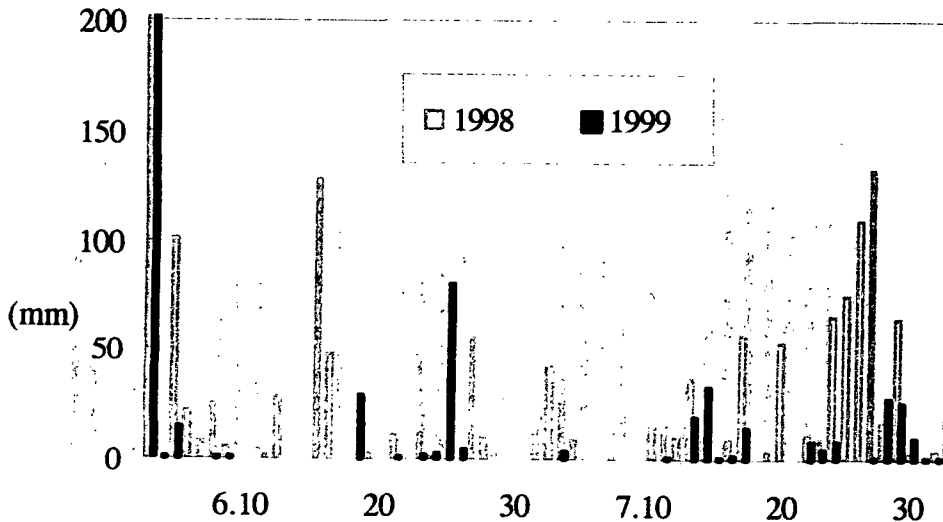
나. 진전도 모형의 적합성 검정

기 작성된 병방제모형의 적합성을 검정하기 위하여 1998년과 1999년 2개년에 걸쳐서 수미와 조풍을 공시하고 파종기를 달리하여 시험을 실시하였던 결과는 다음과 같다. 당해연도의 기상조건을 가공한 severity value(SV) 변화와 병진전도를 모형 대입하여 적합성을 검증하였다.

(1) 환경조건과 병진전 정도

(가) 강수량

병진전도 모형의 설명변수로 이용된 severity value(SV)에 중요한 영향을 미치는 강수량은 1998년은 감자생육 중후기에 집중적으로 내려었는데 1999년은 생육기 전반에 걸쳐서 고루 내렸다.



(나) Severity value(SV) 변화

감자역병 초발생 후부터의 SV의 변화는 1998년도와 1999년도는 상당한 차이점을 보이고 있다. 1998년도는 후기에 집중적이고 연속적으로 높은 상태를 보였는데 1999년도는 병 발생 직후 SV가 집중되었던 반면 그 이후에는 고루 분포되어 있다. 이러한 현상은 병 진단도와 방제스케줄 작성에 중요한 요인으로 작용하게 된다.

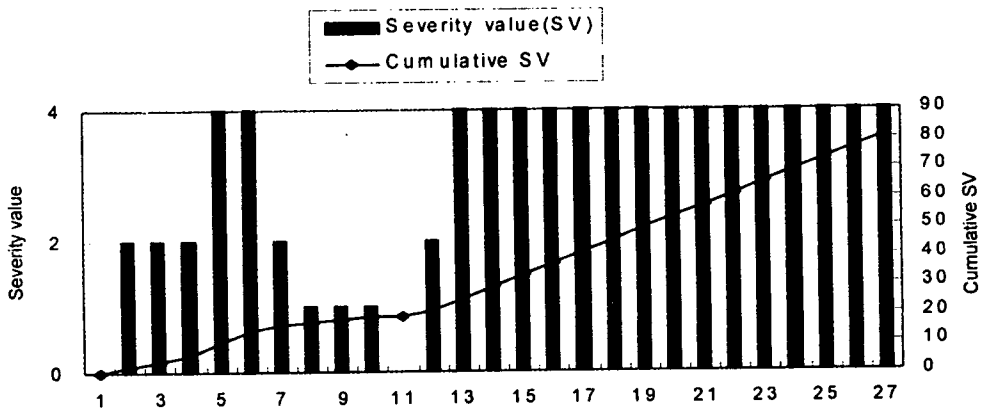


그림 7. Severity value 변화 (1998)

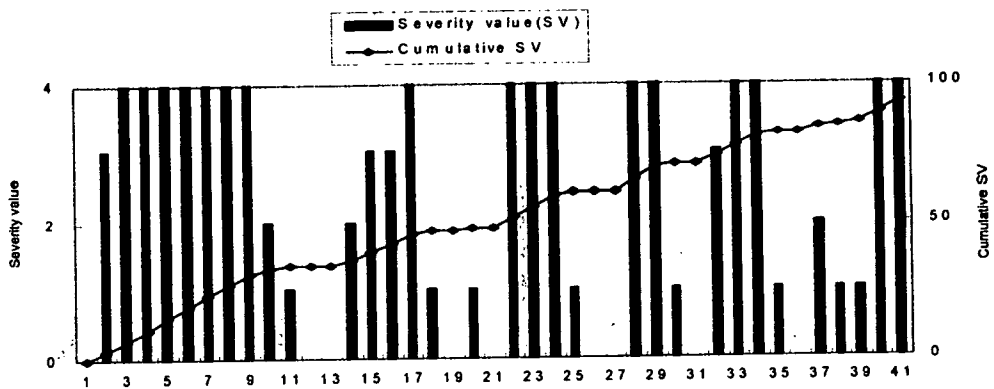


그림 8. Severity value 변화 (1999)

(다) 역병 초발생 및 진전도

표7. 약제살포 간격에 따른 역병 초발생 및 진전도(1998년)

약제살포간격 (적산 SV)	초발생일 (월.일)	진전정도(0-4)			
		7월6일	7월11일	7월18일	7월25일
4	-	0	0	0	0
8	-	0	0	0	0
12	7.5	0.1	0.25	0.25	0.25
16	6.27	0.45	0.5	0.9	1.25
무방제(대조)	6.27	0.8	3.1	3.5	4.0

표8. 약제살포 간격에 따른 역병 초발생 및 진전도(1999년)

품종	약제살포 간격 (적산SV)	초발생 (월.일)	진전정도(0-4)								
			7.8	7.10	7.14	7.19	7.23	7.26	7.30	8.9	8.12
수미	4	7.10	0.0	0.0	0.5	0.5	0.5	0.6	0.7	0.7	1.8
	8	7.5	0.1	0.3	0.5	0.7	1.0	1.3	1.5	2.0	2.8
	12	7.5	0.1	0.1	1.3	1.3	1.5	1.5	2.0	3.9	4.0
	16	7.5	0.0	0.4	1.0	2.0	2.8	2.9	3.3	4.0	4.0
	무방제 (대조)	7.5	0.2	0.7	2.3	3.3	3.6	3.9	4.0	4.0	4.0
조풍	4	7.28	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.2
	8	7.20	0.0	0.0	0.0	0.0	0.2	0.0	0.0	1.5	2.5
	12	7.12	0.0	0.0	0.0	0.2	0.2	0.5	0.5	2.2	3.5
	16	7.12	0.0	0.0	0.0	0.1	0.3	0.7	0.7	2.5	4.0
	무방제 (대조)	7.12	0.0	0.0	0.0	0.3	1.1	2.3	2.5	4.0	4.0

(2) 역병 약제살포 횟수

역병약제살포 횟수는 살포간격에 의해서 결정되는데 적산 SV에 좌우되지
만 강수가 연속될 경우 약제살포가 불가능하므로 강수와 관련되어 실제에는
적산 SV와는 다소 다르게 된다. 1998년의 경우 1998년도의 3~9회에 비하
여 다소 많은 5~10회정도 약제살포를 하였다(표9).

표9. 역병 약제살포 횟수

연도	약제살포간격 (적산 SV)	보호제	치료제	계
1998	4	2	8	10
	8	2	6	8
	12	1	5	6
	16	1	4	5
	무방제(대조)	-	-	-
1999	4	2	7	9
	8	2	5	7
	12	1	3	4
	16	1	2	3
	무방제(대조)	-	-	-

(3) 모형의 검증

작성된 병진전도모형(8식)에서 계산된 결과와 1998년과 1999년 2개년에 걸쳐서 약제살포 수준을 달리하여 시험을 실시하여 조사된 병진전도를 비교한 결과는 다음과 같다.

1998년 1999년 모두에서 모형은 병진전도를 잘 설명하고 있다. 특히 1999년 무방제시에는 90%이상의 결정계수를 보이면서 실제 병진전정도를 모의하고 있다(그림12, 14). 그러나 방제시 특히 방제수준이 높을 경우에는 고도의 유의성은 인정되나 정확도가 다소 낮아지는데(그림11, 13), 이는 살포기술, 약제의 문제 등 기상 외적인 여러 기술적인 문제가 복합적으로 작용한 것으로 생각된다.

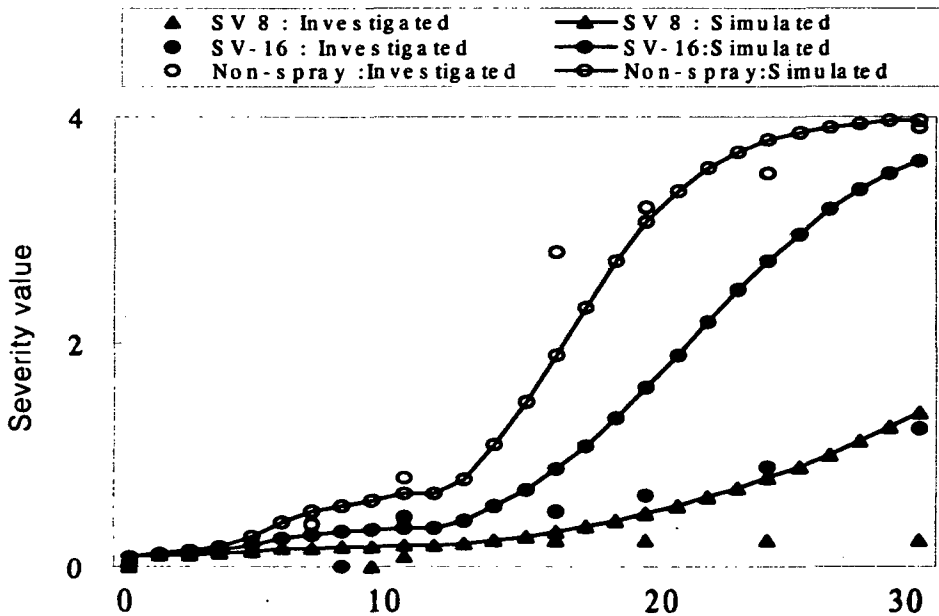


그림10. 역병진전도의 예측치와 조사치 (1998년)

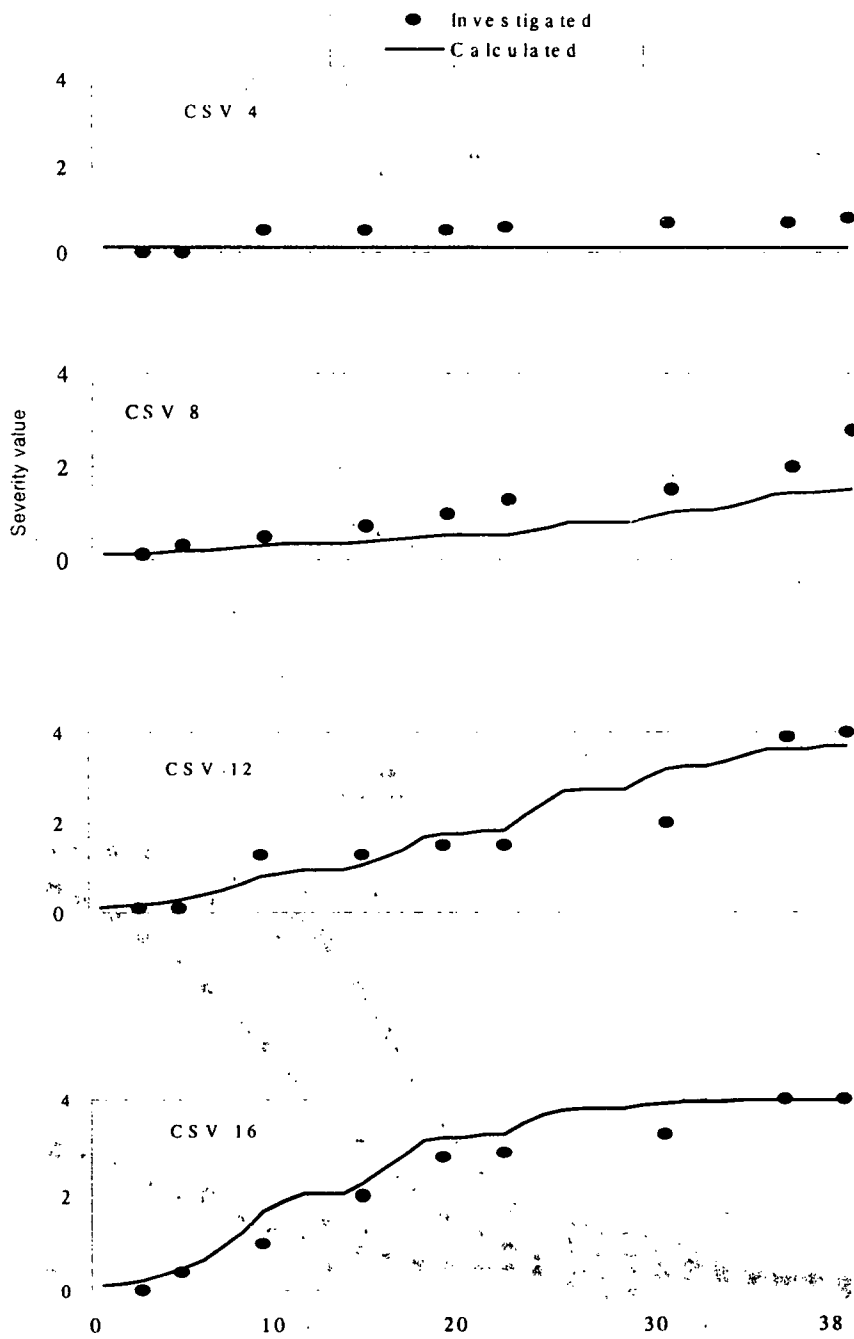


그림11. 역병진전도의 예측치와 조사치 (1999년)

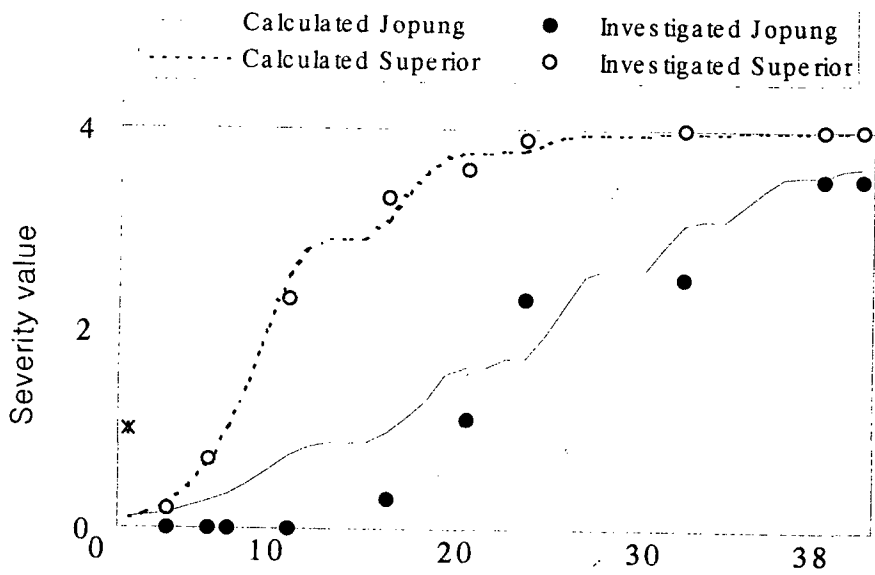


그림12. 무방제시 역병진전도의 예측치와 조사치 (1998년)

병진전도(수미, 약제살포)

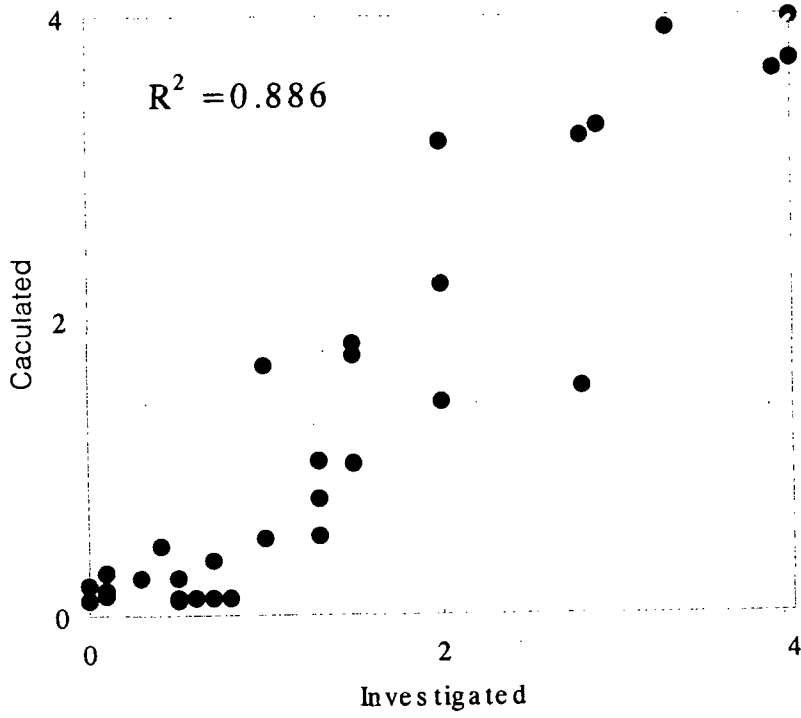
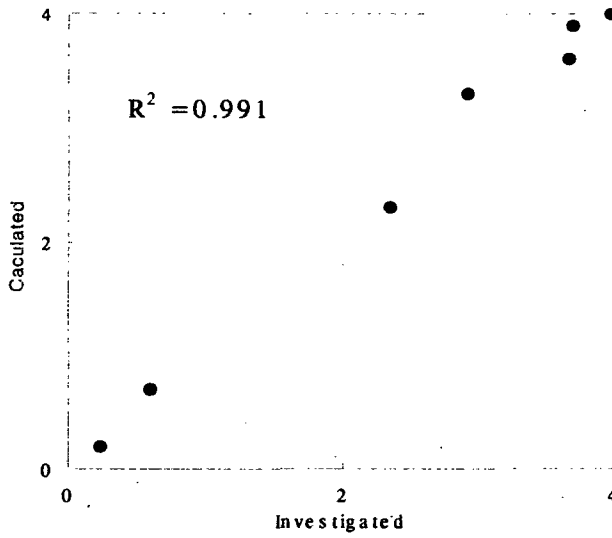


그림 13. 역병진전도의 예측치와 조사치 비교(1998년)

병진전도(수미, 무방제)



병진전도(조풍, 무방제)

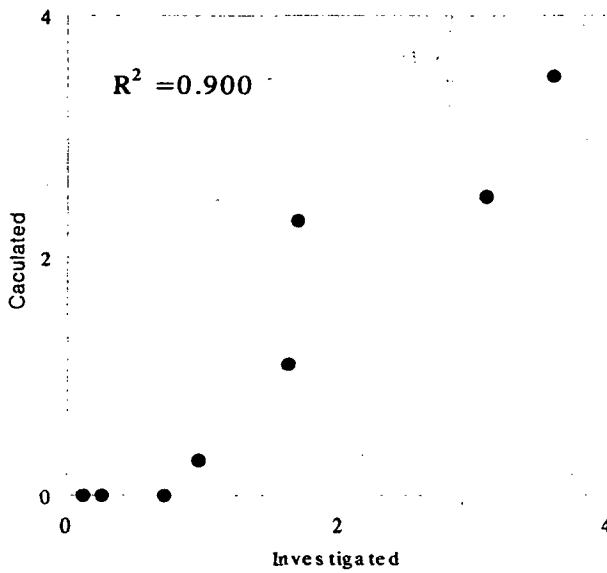


그림14. 역병진전도의 계산치와 조치와의 비교 (1999년)

다. 감자역병 방제모형 작성

(1) 피해허용범위 결정

상기 작성된 병 방제모형을 기준으로 당해연도의 병진전도를 조사하여 발생수준에 따른 수량변화를 검토하여 피해허용수준을 결정하고 이들이 수량감소에 미치는 영향을 조사, 모형화하여 최적 방제스케줄을 작성하였다.

(가) 감자 생장 및 수량

표10. 감자 발육단계

(’98, 대관령)

품종	파종기 (월.일)	출아기 (월.일)	출아율 (%)	출아기간 (일)	괴경형성기 (월.일)
남작	4. 21	5. 23	97.3	32	6.10
	5. 1	5. 25	100.	31	6.13
	5. 11	6. 2	95.0	25	6.15
수미	4. 21	5. 20	98.3	30	6. 9
	5. 1	5. 24	99.3	29	6.10
	5. 11	6. 2	96.0	25	6.14
조풍	4. 21	5. 23	96.7	31	6. 9
	5. 1	5. 28	93.7	30	6.10
	5. 11	6. 3	91.3	25	6.14

표11. 괴경중 변화

(kg/주)

품종	파종기 (월.일)	조사시시(월.일)		
		6. 9	6. 24	7. 18
수미	3. 20	225.7	593.6	1047.5
	3. 31	406.3	745.2	1066.0
	4. 10	142.3	398.4	769.3
조풍	3. 20	569.3	937.0	1268.7
	3. 31	314.4	835.1	1037.0
	4. 10	234.5	715.5	1117.0

표12. 수량

연도	품종	파종기 (월.일)	총서중 (g/주)	규격서율 (%)	규격서중 (g/주)	수량 (kg/10a)
1998	남작	4. 21	350.3	88.6	310.2	1654
		5. 1	363.0	88.8	322.5	1720
		5. 11	200.8	89.6	179.8	959
	수미	4. 21	550.0	89.3	491.2	2620
		5. 1	387.0	92.2	357.0	1904
		5. 11	298.2	89.1	265.8	1418
	조풍	4. 21	390.0	99.4	387.8	2068
		5. 1	342.5	98.1	336.0	1792
		5. 11	300.2	95.2	285.7	1524
1999	남작	5. 1	332.0	90.7	301.5	1605
	수미	5. 1	430.3	92.9	400.0	2133
	조풍	5. 1	491.3	98.7	485.0	2587

표13. 약제살포 간격에 따른 감자수량

연도	약제살포 간격 (적산 SV)	수미		조풍	
		수량(kg/10a)	지수	수량(kg/10a)	지수
1998	4	1,530	100	2,412	100
	8	1,613	105.4	2,469	102.4
	12	1,532	100.1	2,173	90.1
	16	1,170	76.5	1,826	75.7
1999	4	1,991	100	2,587	100
	8	1,934	97.1	2,502	96.7
	12	1,533	77.0	2,455	94.9
	16	1,626	84.1	2,026	78.3

1998년과 1999년 2개년, 약제살포 간격을 다르게 하였을 때의 감자수량은 다음과 같다(표13). 2개년 모두 약제살포 간격이 넓으면 수량이 적어 졌는데 SV 8 까지는 수량에 영향이 없었다. 이러한 현상은 표7과 8에 나타나 있는 약제살포 간격에 따라 변하는 병초발생 및 진전도에 따라서 수량이 결정되었던 것으로 추정된다.

(나) 최적 방제수준(economic threshold) 결정

약제 살포수준에 따른 감자수량 변화를 알아보기 위하여 수량과 약제살포수준과의 관계를 조사하였다. 수미는 약제 살포수준에 따라서 초기감소가 적으며 완만하게 감소하여 SV 12부터 수량이 감소하였다.

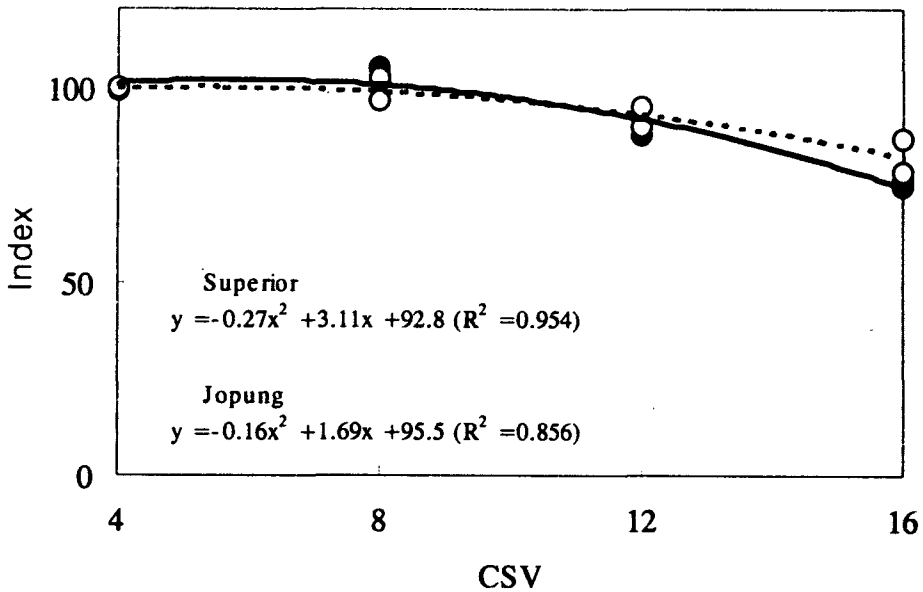


그림 15. 약제살포 간격(CSV)과 감자 수량지수와와의 관계
(수미 : 실선, 조풍 : 점선)

그림15의 결과를 바탕으로 피해허용 수준을 얼마로 둘 것인가에 의해서 다음과 같은 약제살포 기준을 작성할 수 있었다. CSV 4~16까지는 본 연구에서의 실험결과이고 그밖의 수치는 외삽용 참고자료이다. 수량감소 허용기준을 얼마로하느냐에 따라서 경제적인 피해허용수준이 결정되겠는데 10%로 할 경우에는 수미는 CSV 12 에서, 조풍은 CSV 13에서 약제를 살포하여야 하며, 수량감소를 5%로 할 경우에는 두품종 모두에서 CSV 11간격으로 약제를 살포하는 것이 좋은 것으로 나타났다(표14).

표14. 약제살포 간격(CSV)과 감자 수량지수

CSV	수량지수	
	수미	조풍
1	95.6	98.2
2	97.9	98.2
3	99.7	99.1
4	100.9	99.7
5	101.6	100.0
6	101.7	99.9
7	101.3	99.5
8	100.4	98.8
9	98.9	97.8
10	96.9	96.4
11	94.3	94.7
12	91.2	92.7
13	87.6	90.4
14	83.4	87.8
15	78.7	84.9
16	73.4	81.6
17	67.6	78.0
18	61.3	74.1
19	54.4	69.9
20	47.0	65.3

이상의 결과를 종합하여 볼 때 역병 발생후 본 방제모형에 의하여 약제를 살포하면(SV 12 간격) 기존 관행방법에 비하여 약제살포 횟수를 2-3회 정도 줄일 수 있을 것으로 생각된다. 이러한 결과와 표12의 역병 약제살포 횟수를 정리하여 볼 때 본 모형식을 이용하면 역병 초발생전에 3-4회, 발생후

에 2-3회 줄일 수 있을 것으로 확인되어 1작기당 관행 10회정도 살포를 4-6회 즉 50% 단축시킬 수 있으면서도 정상적인 수량이 가능할 것으로 나타났다.

라. 감자 역병관리 시스템 운영기술 확보

(1) 강원고랭지 및 영동지역 국지 기상환경 해석을 위한 AWS 설치

기 작성된 감자병 시스템의 현업기술 확보와 적합성 검정을 위하여 강릉, 대관령, 왕산 외에 홍천(1997년), 태백지역(1998년)에 AWS를 추가 설치하였다. 설치지역은 홍천 씨감자 채종단지과 태백시 귀네미골 (해발 1,000m)이며 대기 및 지중온도, 강수량, 일사량 및 일조시수, 습도 등을 측정중이다. 본 과제에서 작성된 예찰시스템이 감자농장에서 실제 작동되기 시작되는 내년에 실용 예찰정보를 생산할 것이다.

(2) 감자 역병예찰용 컴퓨터시스템 구조 작성

역병 초발생일 예찰은 '이동평균법'과 'BLITECAST' 시스템 두가지 방법을 동시 운용하게되는데 상기 두가지 예찰법중 어느한 방법 만족시 1차 주의보, 나머지에 2차경보를, 두예찰법 동시에 병발생 호조건시 1차에 경보(1-2주일 사이에 병발생)를 발생하도록 시스템을 작성한다.

또한, 기상관측창치가 설치되어 있어서 매시 기상정보 입수가 가능한 기관이나 영농현장에서는 기존의 'BLITECAST' 시스템과 '이동평균법'을 공용하고, 별도의 장비가 없으면 '이동평균법' 만으로 역병예찰이 가능하도록 시스템을 짠다. 프로그래밍은 제3과제에서 수행하였다.

(3) 적정 방제스케줄 작성

병발생전에는 예찰시스템에 의하고 발병후에는 위에서 작성된 피해 허용 범위에 따른 최적 방제수준이 결정된다. 우선은 첫번째 방법은 지금까지 검토된 CSV에 의하여 스케줄을 작성하는 방법인데, 예를 들면 CSV 12가 되면

언제나 약제를 살포한다. 역병 방제약제로는 발병 전에는 ‘만코지’ 계통의 보호약제가, 발병 후에는 ‘디메쏘모르프’ 계통의 치료약제가 추천되어 있는데 다른 계통의 약제를 교대로 살포하는 것이 좋다. 두 번째 방법으로 CSV와 number metrix에 의한 역병방제 정보스케줄(BLITECAST법)을 작성하는데 지난 7일간의 기상조건에 의해서 방제정보를 생산한다(표15, 16).

표15. 약제살포 정보 메시지 number metrix

7일중 강수일수	CSV during last seven days					
	<3	3	4	5	6	>6
<5일	-1	-1	0	1	1	2
≥5일	-1	0	1	2	2	2

표16. Number metrix와 약제살포 방법

메시지 등급	방 제 방 법
-1	무방제
0	경보(2-3일간의 추가 자료를 검토하여 정보생산)
1	7일 간격 살포
2	5일 “

세부방법으로 병이 초발생할 것으로 예찰된 날 이후 7일전에 처음으로 약제를 살포, 살포일 부터 역산하여 1주일 단위로 다음 정보를 생산한다. 즉 7월 1일 예찰하였다면 7월 7일 약제살포를 하고, 7월 1일 부터 7월 7일 까지

의 기상정보를 분석하여 다음 약제살포시기를 결정한다. 다음은 7월14일에도 7월 8일 부터 7월14일의 기상자료를 분석하여 다음시기를 결정한다. 만약 7월14일에 “0”의 경보가 나왔다면 7월 17일에 그날 이전의 일주일 자료를 이용한다. 5일살포에 해당되는 “2”메시지가 나오면 5일후에 살포하고 역시 그날이전의 일주일간의 자료를 이용. 즉 2일간의 자료는 전주 예찰자료와 중복 이용된다.

제4절 감자 병해 분류체계

1. 학술적 분류

번호	분류	병명			학명
		한글	한자	영어(약어)	
1	바이러스 병	바이러스 비루스		virus disease	
1-1		감자잎말림	葉卷病	Potato Leafroll Virus(PLRV)	
1-2		감자Y바이러스 (Y모자이크병)		Potato Virus Y (PVY)	
1-3		감자Y바이러스 (X모자이크병)		Potato Virus X (PVX)	
2	파이코플라스마 (마이코플라스마)			Phycoplasma (Mycoplasma)	
3	세균병			Bacterial disease	
3-1		무름병	軟腐病	Bacterial soft rot Soft rot	<i>Erwinia arotovora subsp.carotovora</i>
3-2		줄기검은병	黑脚病	Blackleg	<i>Erwinia carotovora subsp.atroseptica</i>
3-3		더덩이병	瘡痂病	Common scab, Scab	<i>Streptomyces scabies</i>
3-4		둘레썩음병	輪腐病	Ring rot, Bacterial ring rot	<i>Corynebacterium michiganense</i>
4	곰팡이병			Fungal disease	
4-1		역병, 돌립병	疫病	Late blight	<i>Phytophthora infestans</i>
4-2		겹등근무늬병	夏疫病	Early blight	<i>Alternaria solani</i>
4-3		검은무늬병 잘록병	黑痣病	Rhizoctonia canker, Black scurf, Dampng off, Stem canker	<i>Rhizoctonia solani</i>
4-4			切片腐敗病	Seed piece decay	<i>Fusarium solani</i>
4-5		마른썩음병	乾腐病	Dry rot, Fusarium dry rot	<i>Fusarium solani</i>
4-6		시들음병	萎凋病	Fusarium wilt, Wilt	<i>Fusarium oxysporum</i> , <i>F. solani</i>
4-7		반쪽시들음병	半身萎凋病	Verticillium wilt	<i>Verticillium dahliae</i> , <i>V. albo-atrum</i>

4-8		가루더렁이병	粉狀瘡痂病	powdery scab, Corky scab	<i>Spongospora subterranea</i>
4-9		갯빛곰팡이병	灰色黴病	Gray mold	<i>Botrytis cinerea</i>
4-10		은무늬병	銀痂病 銀斑病	Silver scurf	<i>Helminthosporium solani</i>
4-11			菌核病	White mold Stalk break	<i>Sclerotinia Sclerotiorum</i>
4-12				Stem rot, Southern blight	<i>Sclerotium rolfsii</i>
4-13			炭疽病	Black dot, Antracnose	<i>Collectotrichum atramentarium</i>
5	생리적 병해			Physiological disease	
5-1			塊莖內部 褐斑點	Internalbrown spot, Internalheat necrosis	
5-2			黑色心腐病	Black heart	
5-3			中心空洞	Hollow heart	
5-4			維管束變色	Vascular ring, Vascular discoloration	
5-5			皮目肥大	Lenticel enlargement	
5-6			除草劑被害	Herbicide injury, Chemical injury	
5-7			綠化	Greening	
5-8			日燒	Suncald, Sunburn	
5-9			低溫(霜害)	Frost injury	
5-10			2次生長	Second growth, Growth abnormality	
6	미량요소			Mineral deficiency	
6-1			N 缺乏	Nitrogen(N) deficiency	
6-2			P 缺乏	Phosphorus(P) deficiency	
6-3			K 缺乏	Potassium(K) deficiency	
6-4			Ca 缺乏	Calcium(Ca) deficiency	
6-5			Mg 缺乏	Magnesium(Mg) deficiency	
6-6			Mn 過多	Manganese toxicity	
6-7			B 缺乏	Boron deficiency	

2. 병증상에 의한 분류(KEY TO POTATO DISEASE)

1. 식물체(entire plant)

1-1. 矮化, 생육불량(stunting and/or poor growth) : 바이러스, K결핍, N결핍

1-2. 시들음(Wilt)

1-2-1. 한해(旱害, drought)

1-2-2. 황화(Chlorosis), 萎凋(wilt), 早期 枯死 : 선충(시스트선충, 뿌리혹선충) 피해

1-2-3. 일부 시들음, 황화(wilt and chlorosis)

1-3. 괴저(Necrosis)

1-4. 기형(Deformation, Malformation)

1-4-1. 短縮 莖(마디단축, Internodes shortened)

1-4-2. 斑족 縷葉(방추형 주름엽, Spindly, crinkled leaves) : 약해(chemical injury)

1-4-3. 多莖叢生(Numerous stems, bushy): 유전적기형(genetic abnormalities), mycoplasmas

1-4-4. 총생(Rosette)

1-4-5. 잎자색화, 황화(정부), 왜화(stunting), chlorosis, 마디비대(thickening of nodes), 기중괴경(aerial tuber), necrosis(유관속 및 줄기 괴저) : blackleg, Rhizoctonia, mycoplasma, Fusarium wilt

2. 잎(leaves and leaflets)

2-1. 모자이크(Mosaic mottles) :

2-2. 퇴록, 황화(Chlorosis)

2-2-1. 일반증상 : 시들음병(Fusarium wilt), 반쪽시들음병(Verticillium wilt), 잎 말림바이러스, 질소결핍, 흑지병(정부)

- 2-2-1. 맥간퇴록 혹은 괴저(Interveinal chlorosis or necrosis)
- 2-3. 괴저반점(Necrotic lesions)
- 2-3-1. 壞疽輪點, 괴저반점(Necrotic spots)-모자이크가 진전된 상태 : PVY
- 2-3-2. 엽병 및 소엽괴저(Necrosis of petioles and leaflet) : PVY, TSWY
- 2-3-3. 無윤문(without concentric zonation)
- 2-3-4 윤문반점(with concentric zonation)
- 2-4. 말림(upward rolling)
- 2-5. 변형(deformed)

3. 줄기(Stem)

- 3-1. 총생(coiled sprout and thin-genetic abnormalities, hair sprout) :
mycoplasma, TMV
- 3-2. 왜화(Stunting) and 節間단축(internodes shortened) : PVY, PYDY
- 3-3. 地中病斑(Below Ground)
- 3-4. 지표병반(Near Soil Surface)
- 3-5. 지상병반(above Soil Surface)
- 3-5-1. 표피백화(Epidermis white) : 日燒(sunscald), 우박피해(hail injury)
- 3-5-2. 흑갈색 괴저(Necrosis)
- 3-5-3. 줄기, 엽병에 갈색줄무늬 : PVY
- 3-6. 유관속 변색(Vascular Discoloration)
- 3-7. 隨조직 괴저(Pith Necrosis)
- 3-8. 氣中塊莖 and/or 절간신장(Aerial Tubers and/or Enlarged Nodes) :
pink rot, Fusarium wilt, mycoplasma, Rhizoctonia

4. 괴경 : Tuber

- 4-1. 小塊莖(小薯, 작은감자: Small Size):각종 바이러스, 양분 불균형

- (nutrient Imbalances)
- 4-2. 소서 多着生(Many tubers) : witches' broom, psyllid yellows, 2차생장
(second growth), 흑지병
- 4-3. 연약, 시들음(Flaccid or Wilted) : mycoplasma
- 4-4. 2차 괴경 형성 혹은 조숙(Forming Secondary Tuber or Plants
Prematurely) : second growth, 이차괴경(secondary tubers)
- 4-5. 혹(Galls)
- 4-6. 기형(Deformed)
- 4-7. 외견상 정상괴경, 표면정상(Normal Tuber Shape, Surface Unblemished)-
즉 외형상으로는 정상모양, 무병징이나 내부가 비정상 감자
- 4-7-1 괴경표면에 흑색균핵(Black sclerotia on tuber surface) : Rhizoctonia
- 4-7-2. 괴경기부 투명 : 2차생장, 霜害
- 4-7-3. 괴경조직 당축적 : 저온저장
- 4-7-4. 표면(Surface) and/or 내부(interior)
- 4-7-5. 피하조직 (subsurface tissue)
- 4-7-6. 중심부 공동 or 열개 : hollow heart
- 4-7-7. 내부경화(Interior firm)
- 4-7-8. 내부경화(Interior firm to soft)
- 4-7-9. 유관속 변색(vascular discoloration)
- 4-8. 외견상 정상괴경, 표면병징(Normal Tuber Shape, Surface Blemished
Without Active Rot)
- 4-8-1. 표피 직하 회백색 반점(chalky white spots below skin, later dry
granular, becoming darker) : rot nematode
- 4-8-2. 표시 갈색윤문, 괴저(Brown rings to necrotic areas : PVY, PMTV
- 4-8-3. 피목병징(Lenticels affected) : 피목비대(enlarged lenticels), bacterial
soft rot, powdery scab, stem rot, charcoal rot

- 4-8-4. 복지 부착면 변색(Stolon attachment discolored) : chemical injury, jelly end rot, stem-end browning, brown rot, ring rot, stem rot, Verticillium wilt, Fusarium wilts, boron deficiency, charcoal rot, pink rot
- 4-8-5. 눈변색(Eyes discolored)
- 4-8-6. 상처 이병(Wounds infected) : bacterial soft rot, powdery scab, leak, Fusarium tuber rots
- 4-8-7. 표피직하(얕은) 병반(Shallow lesions anywhere)
- 4-8-8. 암갈색 陷沒(Brown to dark sunken pits), 돌출병반, 암갈색 변색 (russet or raised lesions or russet discoloration)
- 4-8-9. 硬병반(Firm lesions), 얕은 함몰(slightly sunken, relatively shallow)
- 4-9. 부패(Active Rot)
- 4-9-1. 표피균열, 변색(Skin cracks, discolored) : ring rot
- 4-9-2. 연질 空洞(Cavities spongy, shrunken) : Fusarium tuber rot
- 4-9-3. 회색변색(Gray smoky discoloration)
- 4-9-4. 수침상(Watery)
- 4-9-5. 반 수침상(Smei watery)
- 4-9-6. 병징경계 명확(Advancing margin clearly delimited) : white mold, Fusarium tuber rot
- 4-9-7. 조직경화, 건조, 공동(Tissue firm to dry, punky with cavities)
- 4-9-8. 유관속 흑갈색 변색(Vascular tissue discolored reddish brown to black) : brown rot, ring rot
- 4-10. 이차적 부패(Secondary Rots)----- brown rot, ring rot, low or high temperature injury, blackheart, late blight, stem rot, charcoal rot, Rhizopus rot, and rot nematode 등의 일차 이병후 나타나는 증상

5. 종서=씨감자(Seed tuber)

- 5-1. 부패(Decay) : 산소관련(oxygen relations), blackheart, low or high temperature, Fusarium tuber rot, leak, bacterial soft rot, stem rot
- 5-2. 휴면, 발아지연(Dormant, or delayed in germinating) : mycoplasma
- 5-3. 휴면부족(Dormancy lacking) : psyllid yellows, witches' broom, second growth
- 5-4. 비정상왜화(Abnormallythin) : genetic abnormalities, hair sprout, mycoplasma
- 5-5. 피경안으로 성장(Growing into the tuber) : 피경내부 발아(internal sprouting)
- 5-6. 모서 피경형성(Forming tuber directly) : 이차성장(second growth), 이중피경(secondary tubers), hair sprout, calcium deficiency
- 5-7. 괴저(Necrotic)
- 5-8. 갈색반점(Brown lesions)

6. 복지(Stolons)

- 6-1. 단축(Short) : potassium deficiency
- 6-2. 다수, 연장(Numerous, long) : genetic abnormalities
- 6-3. 괴저(Necrosis) : pink rot
- 6-4. 갈색병반(Brown lesions) : skin spot
- 6-5. 혹(Galls)

7. 뿌리(Roots)

- 7-1. 혹(Galls)
- 7-2. 씨스트(포낭), 백갈색(Cysts, white to brown) : cyst nematodes
- 7-3. 갈색병반(Brown lesions) : skin spot

- 7-4. 피층부패(Cortical decay)
- 7-5. 피층장애(Cortical injury) : lesion nematodes
- 7-6. 부패(General rot) : Fusarium wilts
- 7-7. 괴저(Necrosis)
- 7-8. 왜화 (Stunted) : magnesium deficiency ,boron deficiency ,aluminum toxicity, cyst nematodes
- 7-9. 측근총생(Proliferation of lateral root) : cyst nematodes, root-knot nematode.

참 고 문 헌

안 재 훈, 함 영 일, 박 천 수, 김 병 현, 김 정 간. 1994. 기상정보에 의한 감자역병 예찰시스템 작성. 농업논문집 36(1) : 320-325.

안 재 훈, 함 영 일, 박 은 우. 1998. 감자 역병 초발생일 예찰을 위한 '이동 평균법' 개발. 한국식물병리학회지. 14(1) : 34-40.

Berger, R. D. 1981. Comparison of Gompertz and logistic equations to describe plant disease progress. *Phytopathology* 71 : 716-719.

Bruhn, J. A. and Fry, W. E. 1981. Analysis of potato late blight epidemiology by simulation modeling. *Phytopathology* 71:612-616.

France, J. and Thornley, J. H. M. 1984. Mathematical model in agriculture. Butterworth, London, 80-81.

Friesbie, R. E. and Adkisson, P. L. 1985. IPM : Definitions and current status in U. S. agriculture. In Biological Control in Agricultural IPM System. p.41-51.

James, C. W. 1971. A manual of assessment for plant disease. Canada Department of Agriculture Publication No.1458.

Kim, C. H. and Park, K. S. 1988. A Predictive Model of Disease Progression of Red-pepper Anthracnose. *Korean J. Plant Pathol* 4(4) : 325-331.

김 관 수, 김 현 준, 김 화 영, 김 정 간, 함 영 일, 한 병 희. 1990. 감자 조기내병다수성 신품종 "부豊"육성. 농사시험연구 논문집(원예) 32(2):50-54.

고령지시험장. 1975-1992. 종서단계별 병해검정 및 주요작물 병해충조사. 시험연구보고서.

Krause, R. A., Massie, L. B. and Hyre, R. A. 1975. BLITECAST : A computerized forecast of potato late blight. *Plant Dis. Rept.* 59(2):95-98.

MacKenzie, D. R. 1981. Scheduling fungicide applications for potato late blight with BLITECAST. *Plant Dis.* 65:394-399.

Shields, E. J., Hygnstorm, J. R., Cürwen, D., Stevenson, W. R., Wyman, J. A. and Binning, L. K. 1989. Pest management for potatoes in Wisconsin - a pilot program. *Am. Potato J.* 66:227-233.

Shtienberg, D., and Fry, W. E. 1990. Quantitative analysis of host resistance, fungicide and weather effects on potato early and late blight using computer simulation models. *Am. Potato J.* 67:277-286.

Stevenson, W. R. 1983. An integrated program for managing potato late blight. *Plant Dis.* 67:1047-1048.

Stevenson, W. R. 1993. IPM for potatoes : A multifaceted approach to disease management and information delivery. *Plant Dis.* 77:309-311.

Thornley, J. H. M. 1976. Mathematical models in plant physiology. Academic Press, London. pp. 8-11.

Tooley, P. W., Sweigard, J. A. and Fry, W. E. 1986. Fitness and virulence of *Phytophthora infestans* isolates sexual and asexual populations. *Phytopathology* 76:1209-1212.

Vanderplank, J. E. 1963. Plant disease : Epidemics and control. Academic Press, N.Y. pp. 349.

Zadoks, J. C. 1981. EPIPARE : a disease and pest management system for winter wheat developed in the Netherlands, EPPO Bull. 11:365-36.

제3장 감자병해충진단 전문가 시스템 개발

제3장 감자병해충진단 전문가 시스템 개발

제1절 서론

1. 연구개발의 목표와 중요성

가. 연구개발의 목표

본 연구는 감자 병해충을 예찰하고 방제하기 위한 전문가 시스템을 구현하고 이를 인터넷상에서 일반 사용자가 활용할 수 있는 원격 진단 전문가시스템을 구축하고 것을 최종 연구 목표로 한다. 즉 사용자가 감자 재배에 있어서 발생할 수 있는 병에 대한 다양한 전문적인 진단 지식을 갖춘 전문가 시스템을 구축하는 것이다.

감자 병해충을 진단하기 위해서 감자 전문가로부터 감자 병해충 진단 지식을 추출하고 추출된 지식으로부터 감자 병해충을 추론할 수 있는 시스템을 구축한다. 그리고 전문가의 지식을 인터뷰를 통하여도 추출할 수 있지만 과거에 축적된 정보를 가지고 감자 병해충 진단 지식을 습득할 수 있는 기술을 연구하고 사용자가 인터넷을 이용해서 원격으로 전문가시스템을 사용할 수 있도록 인터넷상의 원격 진단 시스템을 구축한다.

나. 연구개발의 중요성

감자 병해충 진단 지식은 고령지 시험장에 있는 감자 전문가가 수십년 동안 전문적이고 경험적인 방식으로 보유하고 있는 지식을 소프트웨어화하는 작업이다. 이와 같은 연구는 무형적으로 보유하고 있던 지식을 유형적으로 표현하므로써, 지속적으로 새로운 진단 지식을 추가하고 보완하는 용도로 활

용될 수 있도록 구축하고 있다. 즉 감자 병해충이 발생했을 때 많은 사용자는 적은 인력의 감자 전문가와 접촉하기가 어렵다. 그러나 감자 전문가시스템을 소프트웨어화하면 얼마든지 전문가지식을 수적인 제약 없이 사용할 수 있는 것이다. 또한 전문가시스템을 소프트웨어화하면 감자 병해충을 진단하고 예찰하는 진단 지식을 얼마든지 새로 생각하여 추가할 수 있으며, 일단 표현된 진단 지식도 활용과정에서 보면 불충분하여 이를 보완을 할 수 있다.

감자 병해충 진단 전문가시스템은 감자 재배 농민이 감자 재배 중에 병해충이 발생하였는데, 구체적인 병해충 이름을 몰라서 이를 효과적으로 방제할 수 없는 경우에 도움을 주기 위한 시스템이다. 또한 위에서 언급했듯이 감자 병해충 진단 전문가의 입장에서 보면, 감자 병해충을 잘 진단할 수 있는 전문가의 수가 많지 않으므로 많은 농민에게 양질의 정보를 제공하는 데는 한계가 있으므로 일상적인 병해충에 대한 예찰 및 방제는 전문가시스템으로 하여금 농민에게 도움을 주도록 하고 보다 복잡하고 delicate한 병해충을 처리하는 데 감자 전문가가 많은 시간을 투자할 수 있도록 해준다.

2. 연구개발 내용 및 범위

감자 병해충 진단 전문가시스템을 구축하기 위해 지식습득 방식에 대해 연구를 하였으며 감자 병해충 진단 지식을 추출한다. 추출된 진단 지식을 규칙기반의 지식으로 표현하고 병충해 진단 지식을 주요 병해충 18가지의 잎, 줄기, 괴경에 대한 주 증상, 부 증상을 전문가로부터 인터뷰하여 추출하였고 이를 소프트웨어로 구축하였다. 현재, 이와 같은 진단 지식을 이용하여 구축한 지식베이스는 규칙으로 표현되고 사용되고 있다.

감자의 병해충을 진단하고 적절한 방제 지식을 제공하기 위한 전문가시스템이 블랙보드 방식의 추론 기관을 이용하여 설계하였다. 또한 인터넷상에 연동이 가능하고 감자 병해충 진단에 사용자가 쉽게 참여 할 수 있도록 사용자 인터페이스를 메뉴 기반으로 하였으며 병해충에 걸린 감자의 화상 정보를 제공할 수 있는 구조를 구현하였다.

감자 병해충 지식을 인터뷰기반으로 구하는 방식이 가장 효율적이다. 그러나, 경우에 따라서는 인터뷰 기반 방식에 소요되는 시간이 많을 수가 있으므로, 이를 보완하기 위한 기계학습 기반의 지식습득 연구를 하였다.

제2절 감자 병해충 진단 지식

1. 전문가 지식 획득

가. 진단 지식 추출

감자 병해충 진단 지식은 연구의 가장 중요한 분야 중의 하나이다. 양질의 지식베이스는 전문가시스템의 성능을 결정하므로, 고랭지 농업시험장 연구팀의 도움을 받아 감자 병해충을 진단할 수 있는 지식을 인터뷰 방식으로 추출하여 병해충 진단 지식을 획득하였다.

주요 감자 병해충을 분류하고 각 병해충에 대한 잎, 줄기, 괴경에 대한 주요 증상을 표현하였다. 또한, 품종별, 생육 단계별로 진단이 가능하도록 주요 증상 부분에 이를 입력하였다. 고랭지 농업시험장의 감자 전문가로부터 습득한, 감자 병해충 분류 표, 감자 주요 병해충 증상 및 방제 관련 자료를 개인 인터뷰를 통해서 감자 병해충 진단 지식을 추출하였다.

나. 규칙 기반 지식베이스

감자 병해충 진단 지식을 활용하여 전문가시스템에 활용이 가능한 지식베이스를 구축하였다. 방식은 규칙 기반의 지식베이스로서 다음과 같은 기본 구조를 가지고 있다.

RuleNumber : If Symptom₁ and Symptom₂ and . . . and Symptom_n Then Conclusion.

예를 들면, “감자의 잎이 말리고, 황화되며 줄기가 직립성장하면 감자잎 말림 바이러스일 가능성이 높다”와 같은 지식은 위의 기본 구조를 이용하여 다음과 같은 규칙으로 표현되어 전문가시스템에 의해서 활용된다.

Rule1000 : If symptom(잎, 형태, 말린다) and symptom(잎, 색깔, 황화) and symptom(줄기, 성장, 직립)
Then 병명(감자잎 말림 바이러스)

이와 같은 구조의 규칙을 표현하기 위해서는 감자 병해충 증상을 전문가시스템이 활용하기 위한 규칙의 형태로 표현하기 위해서 규칙의 Condition 부분을 보다 체계적으로 다음과 같이 표현하였다.

symptom(Type, Attribute, Value)

symptom은 각 감자 병해충의 증상을 표현하는 용어이고 3개의 argument로 모든 병해충의 증상을 표현하였다. 일반적으로 진단을 위한 전문가시스템을 구축하기 위해서는 각 증상의 Type, 각각의 Type이 가지는 Attribute, 각 Attribute가 가지는 Value등을 지정하게 된다.

감자 병해충 증상을 효과적으로 표현하기 위해서 다음과 같은 형태로 세 가지 argument를 표현하고 있다.

Type : 감자 병해충과 관련되는 주요 부위
Attribute : 감자의 각 부위가 가지는 증상의 종류

Value : 증상의 종류 중의 특정한 값

이와 같은 구조를 가지기 위해서, Type은 잎/줄기/괴경이 되고, Attribute는 각각의 잎/줄기/괴경이 가지는 주요 증상 속성을 나타낸다. 예를 들면, 잎인 경우에 “형태”, “색깔”, “모자이크”, “시들음”과 같은 속성을 정의해야한다. 따라서, 본 연구에서는 감자가 병해충이 들었을 때 감자의 잎, 줄기, 괴경의 상태를 표현하기 위한 용어를 정의하고 이 용어들이 Attribute의 값이 된다. 이와 같은 용어는 고랭지 시험장의 전문가가 제공한 자료를 분석하여 최적의 용어 set를 정의하고 이를 다시 고랭지 시험장 전문가의 확인을 받아서 구체화였다.

Type과 Attribute를 결장하고 각 Type, Attribute가 가지는 값을 이용하여 증상을 표현하였다. 예를 들면, “감자의 잎이 황화 되었다”는 증상은 symptom(잎, 색깔, 황화)와 같은 형태로 표현된다.

다. 감자 병해충 증상 분류

감자 병해충 증상을 크게 주 증상과 부 증상 두 가지로 나누었다. 주증상은 해당 감자 병해충의 진단에 매우 주요한 의미를 가지는 증상이고, 부증상은 해당 감자 병해충과 관련이 있는 증상들이다. 주증상과 부증상을 나누어 감자의 병해충을 표현하는 이유는 빠른 시간 내에 관련되는 병해충을 파악하고 이를 진단하기 위한 용도를 위함이다.

감자의 병해충 증상이 입력되면 전문가시스템은 주요 증상을 검토하여 관련이 있는 감자 병해충 list를 생성하게 된다. 그러나, 감자가 걸린 병해충은

한가지라는 가정으로 본 연구를 수행하므로, 이 list 중에서 어느 하나를 선정하여야할 필요가 있다. 이 때, 부증상을 이용하여 각 병해충을 검사하면서 부증상을 만족하지 않는 병해충 이름을 제거하는 방식을 활용하면 감자 병해충의 원인을 효과적으로 알아낼 수 있다.

또한, 각 감자 병해충과 유사한 병해충 정보, 병해충 사진 정보, 기타 관련 정보를 지식베이스함으로써, 효과적인 진단이 이루어지도록 시스템을 설계하고 구현하였다.

라. 감자 병해충 진단지식 구축

고령지 농업시험장의 전문가와 인터뷰를 통하여 감자 병해충 진단지식을 추출하였다. 감자 역병에 대한 지식은 주 증상과 부 증상으로 나누어 잎, 줄기, 괴경별로 추출하였으며 감자역병의 유사병해충도 추출하였다. 추출된 감자병해충 진단 지식은 다음의 표와 같다.

감자병해충진단 지식 표

병명	증상정도	부위	증상
감자잎 말립 바이러스	주증상	있	① <u>잎이 말리고</u> ^ 숟가락처럼 말린다(cupping) ② <u>황화되고 키가 작아진다.</u> (잎이 시들지는 않는다)
		있	① 때로는 잎 뒷면이 자색이 된다. ② 때로는 잎이 두꺼워진다(손으로 쥐어보면 종이 접는 것 같은 증상)
	부증상	줄기	① 직립생장
		괴경	① 괴경이 작아지고 ② 괴경을 절단하면 유관속이 망상으로 괴저된다. (그물모양의 괴저증상이 생긴다)(갈색점이 망상으로 생긴다)
	유사병해 충		① 곰팡이병 시들음병 ② 곰팡이병 검은 무늬병(잘록병) ③ 생리적 잎말림병
	사진 비고		- 가물 때 많이 생긴다 - 진딧물에 의해서만 전염
바이러스 병 (감자 Y 모자이크)	주증상	잎	① 퇴색된 황색 모자이크가 생긴다. ② <u>옆맥상에</u> 황색 모자이크 (X 모자이크와 다른점) 심한 경우에 옆맥을 따라서 갈색 또는 흑갈색의 반점이 생긴다. 심한 경우 잎이 주글주글해지고 말라서 떨어진다.
		줄기	① 때로는 갈색 줄무늬
	유사병해 충사진		감자A모자이크, 감자X모자이크 등
	비고		즙액전염(상처부위 접촉 등에 의한), 진딧물에 의한 전염

병명	증상정 도	부위	증상
바이러스 (감자 X 모자이크)	주증상	잎	① 퇴색된 황색 모자이크가 생긴다. ② 엽맥간(엽맥사이에) 황색 모자이크 (Y 모자이크와 다른점) 반점이 드물게 생기고 생기더라도 약하다. 잎이 다소 약하게 주글거린다.
	유사병 해충		감자S모자이크, 감자Y모자이크 등
	비고		점촉 혹은 즙액전염, 일부 저작구(씹는) 곤충에 의한 전염
세균병, 무름병 (연부병)	주증상	잎	- 유조직이 파괴, 물렁해지고 부패
		줄기	" "
		괴경	- 괴경의 유조직이 파괴되어 물렁해짐 - 백색이나 갈색 썩음증상, 흔히 불쾌한 냄새를 동반
	부증상	잎	
		줄기	
		괴경	건조한 조건에서 부패 부위는 마르면서 검고 움푹 들어감
	유사병 해충		둘레썩음병, 검은무늬병, 줄기검은병
	사진		
비고		- 고온 다습조건에서 발생 - 생육기간에 발생(식물체 상처를 통하여 침입 발생) - 저장중에도 발생	

병명	증상정도	부위	증상
세균병, 줄기검 은병 (흑각병)	주증상	잎	- 황화되고 소엽은 끝이 위로 말리며 나중에는 식물체 전체가 시들고 죽게된다
		줄기	- 지체부가 전형적으로 검은 잉크색으로 부패 - 씨감자 절편에서 줄기의 지체부, 때로는 줄기전체가 흑변 부패
		괴경	
	부증상	잎	- 생육불량, 왜소
		줄기	- 왜소, 생육초기에 뿔뿔 - 습한조건:부패하여 흐물흐물하고 끈적끈적해지며 식물전체로 퍼짐. - 건조한 조건:이병조직은 마르고 시들시들해짐 (줄기의 지체부 아래 부위에 한함)
		괴경	- 기부에 약한 유관속변색 증상 - 괴경정부가 썩는 증상
비고		생육초기 습한 토양, 서늘한 온도조건에서 다발	
세균병, 둘레썩 음병 (운부병)	주증상	잎	- 초기에는 물에 데친 모양, 생육후기에 잎과 줄기가 위조 - 엽육이 퇴색하여 잎 가장자리가 위쪽으로 말리며 - 나중에는 흑갈색으로 변하고 고사
		줄기	- 줄기의 지체부를 절단해 보면 유관속 부분이 황갈색으로 변하여 약간 물렁하고 쥐어짜면 치즈모양의 물질이 나옴

병명	증상정 도	부위	증상
세균병, 둘레썩 음병 (윤부병)	주증상	괴경	- 괴경은 피목 부분이 붉거나 표피에 균열 - 절단하면 유관속이 크림색 내지 담황색으로 변 한 윤상 - 쥐어짜면 치즈와 같은 냄새 없는 즙액이 나오며 피층과 내부육질이 분리
			부증상
		줄기	1-2줄기만 시들음
		괴경	괴경외부가 움푹해짐
	유사병 해충		무름병, 줄기검은병
	사진		
비고			개화직후 고온시 발생
세균병, 더뎡이 병	주증상	잎	없음
		줄기	없음
		괴경	① 더뎡이(부스럼)가 생긴다. (괴경 표면에 부스럼이 생긴다) ② 괴경 부스럼 증상이 유기 또는 침강한다. (주: 침강차이 - 더뎡이병 : 표피가 지저분하게 파열 - 역병 : 약간 들어감)

병명	증상정 도	부위	증상	
세균병, 더뎡이 병	부증상	잎	없음	
		줄기	없음	
		괴경	표면이 갈라짐(갈색)	
	유사병 해충 사진		가루더뎡이병	
	비고		- '대지', '남서'품종에서 많이 발생 - 연작(이어짓기), 건조한 토양, 알칼리성 토양에서 다발생	
	곰팡이 병, 가루더뎡 이병 (분상창 가병)	주증상	잎	없음
줄기			없음	
괴경			- 직경 0.5~2mm의 자줏빛을 띤 갈색의 작은 혹이 생기고, 서서히 어두운 색이 되면서 부패 - 암갈색의 포자낭에 가루투성이의 물질로 가득찬 얇은 함몰증상, 흰색 사마귀모양의 궤양을 형성	
부증상		잎	없음	
		줄기	없음	
		괴경	- 뿌리와 복지상에 다양한 유백색의 균핵(galls)으로 발달하는 작은 괴저반점이 생긴다.	
유사병 해충 사진			더뎡이병	
비고			습하고 서늘한 기후조건에서 발생	
곰팡이 병.역병 (돌립병)		주증상	잎	① <u>잎의 일부가 물에 데친 것 같다.</u> ② <u>잎의 뒷면에 백색 곰팡이가 생긴다.</u> (앞면에 균핵이 있으면 균핵병, 잎의 뒷면이 회백색이면 잿빛곰팡이병)

병명	증상정 도	부위	증상
곰팡이 병.역병 (돌립병)	주증상	줄기	① 갈변되고 변색되며 그 후에 <u>백색 곰팡이</u> 가 생긴다. ② 진행 후반에는 썩는다.
		괴경	- 초기에는 잘 모름 - 괴경이 <u>움푹해짐</u> - 괴경을 잘라보면 <u>얇게(표피에서 5mm정도) 갈변되어 있음</u>
	부증상	잎	초기 잎이 황화 작은반점, 습한 조건에서 불규칙하게 확대
		괴경	흑갈색 변색 움푹해지고 갈색반점(하역병은 흑색)
	유사병 해충 사진		젓빛 곰팡이병과 유사
	비고		저온 다습조건에서 발생
	곰팡이 병, 접동근 무늬병(하역병)	주증상	잎
줄기			일찍 노화
괴경		- 괴경이 불규칙하고, 함몰된다. - 회색에서 갈색으로 또는 자주색에서 흑색으로 변하는 등 다양하다. (역병은 갈색)	
부증상		잎	하위엽에서 주로 증상
유사병 해충			위조병(Verticillium), 젓빛곰팡이병
비고			고온건조한 조건에서 발생

병명	증상정 도	부위	증상
곰팡이 병, 잣빛곰팡 이병	주증상	있	① <u>잎의 앞면에 둥근 병반이 생긴다.</u> (특히, 초기에는 지제부 근처의 하엽에 많음 후기에는 모든 잎에 전달) ② <u>잎의 뒷면에 둥근 병반이 변색되어 회백색 곰팡 이가 생긴다</u>
		줄기	갈색반점
		괴경	없음
	부증상	있	생육말기에 뚜렷, 지제부에 잘 나타남
		줄기	없음
		괴경	괴경표면이 주름, 흑변, 마른 부패
	유사병 해충		역병, 균핵병
	사진		
	비고		
	곰팡이 병, 절편 부패병	주증상	있
줄기			가늘고 약한 줄기로 자람
괴경			- <u>절단면이 갈변 부패되어 발아 불능. 때로는 발아 하더라도 싹이 약하게 자람</u>
유사병 해충			검은썩음병
비고		- 주로 파종후부터 생육초기에 발생 - 불량한 종서절단 조건(광노출, 큐어링 부족 등) - 고온, 저온 등 불량한 토양조건에서 발생	

병명	증상정 도	부위	증상
곰팡이 병, 시들음 병	주증상	잎	① <u>잎이 시든다(상엽).</u> ② <u>잎이 말린다(상엽).</u> ③ <u>잎이 황갈색으로 변한다(하엽).</u>
		줄기	갈변
		괴경	유관속 변색
	부증상	잎	시들음
		줄기	① 줄기를 자르면 도관이 갈색으로 변해있다. ② 심하면 줄기가 썩는다.
		괴경	- 괴경을 절단하면 유관속 부분이 갈변되어있다. - 기중괴경(지상부에 감자가 달림)
	비고		- 마른썩음병(건부병)과 같은 병원균 - 생육초기 과습 냉량한 조건에서 발생
곰팡이 병, 마른썩 음병 (건부병)	주증상	잎	없음
		줄기	없음
		괴경	- 상처부위에 작은 갈색증상 - 장해부위의 표피는 함몰하고 주름, 동심원상으 로 되어 죽은 조직이 마름 - 괴경내부의 괴저부위는 얇은 황갈색에서 어두운 초콜렛색으로 갈변.
	부증상	괴경	- 오래된 죽은 조직은 다양한 색 - 건조하여 조직에 구멍이 뚫리게 됨
	유사병 해충		균핵병, 잿빛곰팡이병의 괴경증상
	비고		- 저장중 발생 - 다습조건에서 상처를 통해 발병

병명	증상정 도	부위	증상
곰팡이 병, 검은무 늪병 (흑지병, 잘록병)	주증상	잎	- <u>잎이 말리고</u> - 정부잎의 로젯트현상 - 생육초기 : <u>과중후 싹이 갈변, 정부잎 황화, 부패</u> (damping off) - 후기 : 붉은 색깔을 띤
		줄기	- 생육중기 : <u>지제부(땅과 줄기가 만나는 부분)가</u> <u>썩고 그 위치에 작은 괴경이 생긴다</u> (stem canker)
		괴경	수확기 : <u>괴경위에 흑갈색의 균핵이 파리똥처럼 생</u> <u>기고, 물로 씻기지 않는다.</u> (black scurf).
	부증상	줄기	- 액아가 자라고 나중에 기중괴경으로 된다. - 지제부 갈변
	유사병 해충		감자잎말림바이러스, 지제부 서리해
	비고		저온 과습토양에서 발생
생리적 장해, 괴경 내부 갈반점	주증상	괴경	- 표면에는 증상이 거의 없고 감자를 절단하면 육 질부는 크고 작은 불규칙한 갈색반점(죽은 세 포)이 다수 산재 - 변색부 세포는 사멸되어 콜크질 모양
	비고		- 큰 감자에서 많이 발생 - '대서'품종에서 심하다.

병명	증상정 도	부위	증상
생리적 장해, 흑색 심부병	주증상	괴경	<ul style="list-style-type: none"> - 괴경 내부의 중심부가 흑색 또는 흑갈색, 흑회색이 되는 증상 - 변색의 윤곽은 명확하나 형태는 불규칙적이어서 엷은 회색이 산재할 때도 있다. - 증상이 진전되면 중심부가 공동이 될 때도 있다.
	부증상	괴경	괴경의 표면에는 별다른 변화가 없으나 심하면 외피가 갈변되고 움푹 들어가게 된다.
	사진		
	비고		<ul style="list-style-type: none"> - 수미품종에서 심 - 주로 저장, 수송중 발생(25℃이상 고온, 0℃ 이하 저온, 괴경내 산소부족이 원인)
생리적 장해, 중심 공동	주증상	괴경	<ul style="list-style-type: none"> - 괴경의 중심부 주위에 공동형성. - 공동은 콜크질 조직에 둘러싸여 있는 소형의 별모양, 또는 몇개의 공동이 연결되어 있는 것들도 있다. - 세포의 괴저에 의해 진행되는 것이 아니라 내부조직의 팽창으로 쪼개어져서 렌즈모양이 공동이 생긴다. - 공동의 내부벽은 희거나 엷은 황갈색이다.
	비고		괴경이 클수록 심, '대서' 품종에서 많이 발생

2. 귀납적 기계학습을 이용한 진단지식 구축

가. 사례를 이용한 진단지식 구축

일반적으로 진단 지식베이스는 전문가와 인터뷰하여 가장 최적의 지식베이스를 구축하게 된다. 그러나, 이와 같은 방식은 시간이 소요되는 문제점을 가지고 있다. 따라서, 많은 분야에서는 기계학습을 이용한 지식베이스를 구축하고 있다. 이 방식은 경우에 따라서 양질의 지식베이스를 구축하지 못하는 문제점을 가지고 있지만 전문가가 추출된 지식의 수준을 결정해 주면 짧은 시간 내에 많은 지식을 추출할 수도 있게 된다. 따라서 본 연구에서는 사례를 이용하여 진단 지식을 추출하기 위한 연구를 수행하고 있다. 본 연구에서 수행하는 방식은 많은 사례를 구축하고 이로부터 지식베이스를 자동 추출하는 방식이다.

감자 진단 사례들을 수집하여 이를 기계학습 프로그램의 입력으로 표현하게 된다. 이 과정을 거치면 각 사례는 <속성-속성값>으로 표현되어 다음과 같은 데이터베이스가 수집되게 된다.

{<잎(색깔), 황화>, <줄기(생장), 직립>, <괴경(크기), 작다>},

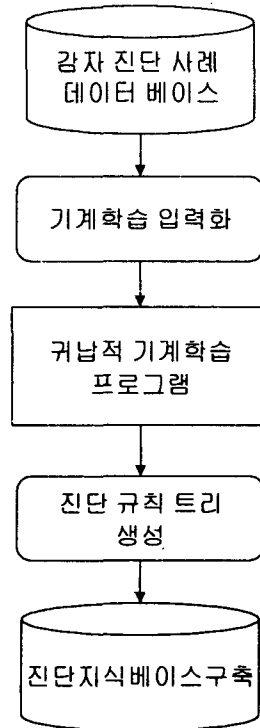
감자 잎 말립 바이러스

{<잎(모자이크), 황색>, <줄기(줄무늬), 갈색>}바이러스병

{<잎(모자이크), 황색>, <잎(모자이크위치), 엽맥>}바이러스병

나. 지식습득 구조

다음은 설계하고 있는 지식습득기의 구조이다.



지식 습득기 구조

이와 같은 감자 병해충 사례들을 수집하면 각 병해충 사례를 특징적으로 나타내는 <속성-속성값>을 추출할 수 있다. 이와 같은 방식의 가정은 많은 병해충 사례를 수집할 수 있다는 가정이다. 즉, 수백건의 사례를 추출할 수 있다면 이러한 사례들로부터 각 병해충을 특징 지을 수 있는 규칙을 추출할 수 있게된다.

이와 같은 방식은 기계학습(machine learning) 분야에서 많이 연구되고 있다. 즉, 모든 사례를 <속성-속성값>으로 표현하고 각 사례의 병명을 알고 있다면 이와 같은 사례들로부터 각 병해충을 알아내는 데 최적의 규칙을 발견할 수 있는 방식을 적용할 수 있다. 기본적인 아이디어는 다음과 같다. 즉, 아래의 사례베이스가 있다고 가정을 하자.

- { <속성₁, 노란색>, <속성₂, 직립>, <속성₃, 없음> } ⇒ 병해충₁
- { <속성₁, 정상>, <속성₂, 정상>, <속성₃, 모자이크> } ⇒ 병해충₂
- { <속성₁, 노란색>, <속성₂, 직립>, <속성₃, 모자이크> } ⇒ 병해충₁
- { <속성₁, 정상>, <속성₂, 정상>, <속성₃, 모자이크> } ⇒ 병해충₄
- { <속성₁, 정상>, <속성₂, 정상>, <속성₃, 모자이크> } ⇒ 병해충₅
- { <속성₁, 노란색>, <속성₂, 직립>, <속성₃, 없음> } ⇒ 병해충₁

이와같은 사례베이스로부터 알 수 있는 내용은

속성₁ 이 노란색이고 속성₂ 가 직립이면 병해충₁ 이라는 지식이다.

이 때 속성₃ 보다 속성₁, 속성₂ 이 더 중요하다는 것을 기계학습 방식이 알아내고 이로부터 지식베이스를 추출할 수 있게 된다. 본 연구에서는 엔트로피 개념을 이용하여 최적의 속성을 추출하고 이를 이용하여 지식베이스를 구축하는 연구를 수행하였다.

다. 엔트로피 개념을 이용한 지식습득

본 연구에서는 엔트로피 개념을 이용하여 감자 병해충 진단 지식을 추출하

는 연구를 수행하였다. 엔트로피는 다음과 같은 특성을 가지고 있다. 예를 들어 다음과 같은 네 가지 속성에 따라 나타난 병해충에 관한 7개의 사례가 있다고 가정하자.

{<속성1, 노란색>,<속성2, 직립>,<속성3, 옆맥상모자이크>,<속성4, 정상>} 병해충1

{<속성1, 황색>,<속성2, 정상>,<속성3, 옆맥간모자이크>,<속성4, 줄기검은색부패>} 병해충2

{<속성1, 정상>,<속성2, 직립>,<속성3, 정상>,<속성4, 정상>} 병해충3

{<속성1, 정상>,<속성2, 직립>,<속성3, 옆맥상모자이크>,<속성4, 정상>} 병해충3

{<속성1, 황색>,<속성2, 정상>,<속성3, 옆맥간모자이크>,<속성4, 줄기지체부부패>} 병해충5

{<속성1, 정상>,<속성2, 직립>,<속성3, 옆맥간모자이크>,<속성4, 정상>} 병해충3

{<속성1, 노란색>,<속성2, 직립>,<속성3, 옆맥상모자이크>,<속성4, 정상>} 병해충1

위에서 나타난 사례로 미루어보아 속성1이 정상이고 속성2가 직립이며, 속성4가 정상인 사례에서는 모두 병해충3이 발생하였다는 규칙이 성립한다는 것을 알 수 있다. 이러한 규칙을 얻기 위하여 본 연구에서 사용하는 기계학습 추론방식을 사용하면 속성에 따라 구분된 여러 속성값들을 자식노드로 하고 또한 이 자식노드에서 발생된 속성값들에 따라 나뉘는 자식노드들로 이루어진 결정트리의 생성으로 특정한 감자 병해충이 발생할 수 있는 여러 규칙들을 쉽게 추출할 수 있게 되는 것이다. 각각의 감자병해충을 수많은

사례에서 나타나는 속성값들로 인해 구분하여 특징 지을 수 있는 규칙을 만들기 위하여 본 연구에서 사용한 방식이 decision tree이다. 이 decision tree를 만들기 위해서 이용하고 있는 Gain값은 엔트로피에 의해 얻어지는데 이 엔트로피 값은 트리형성의 과정에서 각각의 속성중의 가장 최상의 것을 선택할 수 있도록 해주는 것으로 클래스에 따른 속성의 분포가 일정한 부분에 편중될수록 트리생성에 유리한 결과를 갖는다. 한 개의 감자 병해충이 정해지지 않은 사례(S)가 입력되었다고 가정하고 이 사례가 많은 감자 병해충중의 어느 쪽으로 진단이 날 것인지를 예를 드는 과정에서 이 엔트로피값은 지금까지 입력이 되어있는 각 감자 병해충이 결정되어 있는 사례들의 속성의 상관관계로 인하여 결정되는데 여기서 병충해1에 해당되는 사례들의 속성값의 분포를 p^+ , 병충해2에 해당하는 사례들의 속성값의 분포를 p^- 라 하자. 이 분포값들은 수치 0부터 1의 값으로 나타나며 중간값 0.5는 속성의 클래스(특정 병충해)별 분포가 편중되어 있지 않고 고루 퍼져있는 즉, 불확실도가 높다는 것을 뜻하며 이보다 작거나 클수록 속성별 클래스 분류가 잘 이루어져 있다는 것을 의미한다. 이를 수치상의 높고 낮음으로 불확실성을 정의하기 위하여 다음과 같은 식으로 엔트로피를 구한다.

$$Entropy(S) = -p^+ \log_2 p^+ - p^- \log_2 p^-$$

위의 식에서 추출된 엔트로피값은 그 값이 작으면 작을수록 사례S에 대한 특정 클래스구분이 용이하다는 것으로, 즉 잘 분류되는 것이 작은 값을 가지고 엔트로피 값이 작으면 분류가 잘된 것을 의미하는 것이다. 엔트로피값은 그 자체로는 트리 생성의 결정값이 되지않고 이를 다시 부모노드와 자식노드의 엔트로피값을 연관시켜 추출되는 값을 결정트리 생성에 직접 이용하

는데 이것이 바로 Information Gain값이다. 이 엔트로피값은 수치 0부터 1의 값으로 나타나며 중간값 0.5는 속성의 클래스별 분포가 편중되어 있지 않고 고루 퍼져있는 즉, 불확실도가 높다는 것을 뜻하며 이보다 작거나 클수록 속성별 클래스 분류가 잘 이루어져 있다는 것을 의미한다. 이 엔트로피값은 그 자체로는 트리 생성의 결정값이 되지않고 이를 다시 부모노드와 자식노드의 엔트로피값을 연관시켜 추출되는 값을 결정트리 생성에 직접 이용하는데 이것이 바로 Information Gain값이다. 이 Information Gain값은 트리형성의 과정에서 각각의 속성중의 가장 최상의 것을 선택할 수 있도록 해주는 것으로 다음과 같은 식으로 산출된다.

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

위의 식의 결과값이 클수록 선택된 속성에 따른 클래스의 분류결과가 더욱 향상된다. 하지만 여기서 고려해야할 것이 있는데 만약 속성이 다량의 속성값을 갖는다고 해서 그 속성들이 모두 중요한 의미를 갖는다고 정의할 수는 없기 때문에 이러한 속성들이 나타날 경우 Gain이 이 속성들을 선택하게 되는 부작용을 방지하기 위해 특정한 속성의 속성값이 퍼져있는 분포를 Gain값으로 나누어 정상적인 결과가 추출되도록 하는 방법이 있다. 이 방법으로 추출되는 값이 Gain Ratio값이며 그 식은 다음과 같다.

$$GainRatio(S, A) \equiv \frac{Gain(S, A)}{SplitInformation(S, A)}$$

$$SplitInformation(S, A) \equiv - \sum_{i=1}^c \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}$$

라. 결정 트리 생성

이 GainRatio로 인하여 결정트리를 생성하게 된다. 이 결정트리를 생성하는 과정을 위의 사례1)의 경우에 적용하여 나타내면 먼저 다음과 같은 3단계의 순서로 인하여 결정트리가 생성된다.

- 1) 속성에 따른 엔트로피 계산.
- 2) 위의 계산값을 이용하여 여러 속성들 중에서 중요한 속성 가려내기.
- 3) 2)의 과정을 거쳐 결정된 중요한 속성들의 Gain값을 산출.
- 4) 위의 Gain값을 이용하여 부모노드와 그에 따라붙는 자식노드를 생성하여 결정트리 추출.

{<속성1, 노란색>, <속성2, 직립>, <속성3, 옆맥상모자이크>, <속성4, 정상>}

병해충1

{<속성1, 황색>, <속성2, 정상>, <속성3, 옆맥간모자이크>, <속성4, 줄기검은 색부패>} 병해충2

{<속성1, 정상>, <속성2, 직립>, <속성3, 정상>, <속성4, 정상>} 병해충3

{<속성1, 정상>, <속성2, 직립>, <속성3, 옆맥상모자이크>, <속성4, 정상>}

병해충3

{<속성1, 황색>, <속성2, 정상>, <속성3, 옆맥간모자이크>, <속성4, 줄기지제 부부패>} 병해충5

{<속성1, 정상>, <속성2, 직립>, <속성3, 옆맥간모자이크>, <속성4, 정상>}

병해충3

{<속성1, 노란색>, <속성2, 직립>, <속성3, 옆맥상모자이크>, <속성4, 정상>}

병해충1

다음은 위의 예들을 알아보기 쉽게 표로 나타낸 것이다.

예	속성1	속성2	속성3	속성4	병해충
1	노란색	직립	옆맥상모자이크	정상	병해충1
2	황색	정상	옆맥간모자이크	줄기검은색부패	병해충1
3	정상	직립	정상	정상	병해충3
4	정상	직립	옆맥상모자이크	정상	병해충3
5	황색	정상	옆맥간모자이크	줄기지제부부패	병해충1
6	정상	직립	옆맥간모자이크	정상	병해충3
7	노란색	직립	옆맥상모자이크	정상	병해충1

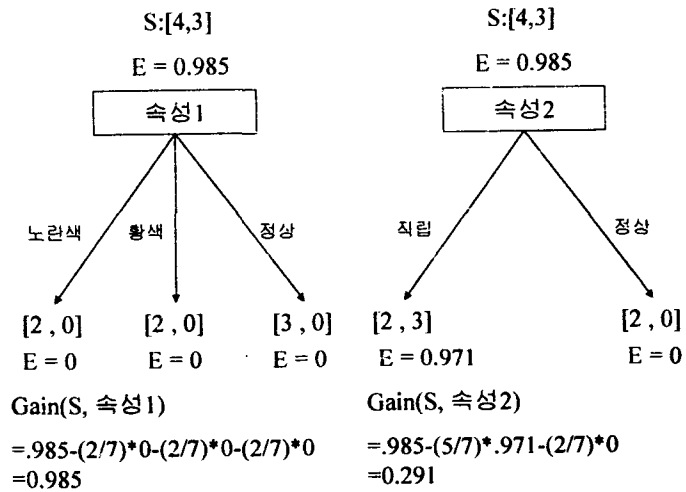
위의 나열된 4개의 속성 중에서 결정트리의 최상위 노드로 삼을 속성을 구하기 위하여 각각 속성을 부모노드로 하여 이로 인해 발생하는 자식노드간의 상관관계를 나타내는 Gain값을 계산하여 가장 높은 값을 가지는 속성을 최상위 노드로 결정한다. 이 과정을 이후에도 계속 반복해 나가면 가장 최적의 결정트리가 생성되는데 위의 예를 들어 처음의 최적 속성을 결정하는 단계는 다음과 같다. 먼저 속성1의 경우를 계산해보면 병해충의 분포는 S [병해충1: 4, 병해충: 3]이고 이의 엔트로피 값은 위에서 언급했던 식으로 계산하면 병해충1일 확률 p_1 이 $4/7$ 또한 병해충3일 확률 p_2 가 $3/7$ 이므로 계산식은 아래와 같다.

$$Entropy(S) = -\frac{4}{7} \log_2 \frac{4}{7} - \frac{3}{7} \log_2 \frac{3}{7} = 0.985$$

여기서 구한 클래스별 엔트로피 값을 기초하여 각 속성에 대한 Gain값을 구하여 결정트리의 생성과정을 진행하게 된다.

위의 속성1과 속성2의 Gain값 계산결과를 보면 속성1의 Gain값이 0.985로 속성2의 Gain값보다 크기 때문에 가장 최적의 속성으로 속성1을 선택하게

된다. 이후에도 위의 계산방식과 동일한 방법으로 자식노드를 결정짓는 속성을 계산 결과에 따라 정함으로써 최종적인 결정트리를 생성하게 된다



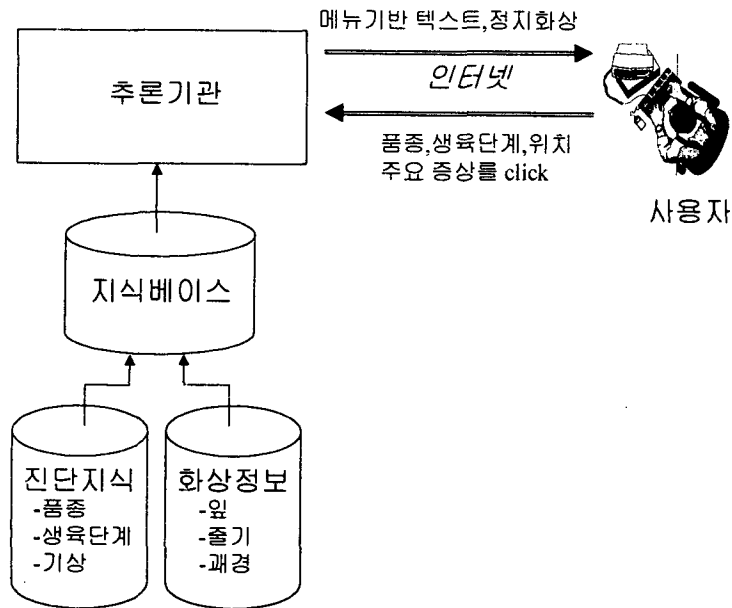
결정 트리

제3절 감자 병해충 진단 시스템

1. 진단 시스템 구조

가. 감자 병해충 진단 전문가 시스템 구조

감자 병해충 진단 전문가시스템의 구조는 다음과 같다.



기존의 전문가 시스템 구축 방식은 크게 다음과 같은 분야로 세분된다.

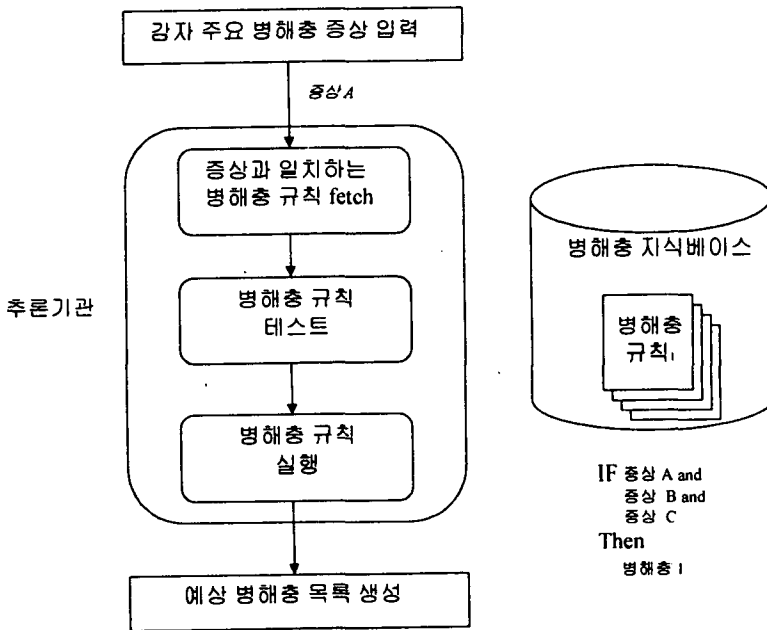
- 추론 기관 구축
- 지식베이스 구축

감자역병 전문가 시스템의 지식베이스는 고령지 시험장의 전문가로부터 습득한 지식베이스로 오랜 경험에 의해서 축적된 지식베이스이다. 이와 같은

지식베이스는 현재, 진단 지식과 화상 정보로 구축되도록 설계되고 있다. 진단 지식베이스는 각 감자 병해충별로 품종, 생육단계, 기상 등과 관련된 정보를 규칙 기반으로 표현되고 있으며, 화상 정보는 각 병해충의 잎/줄기/과경별 관련 화상 정보를 보유하게 된다.

나. 감자 병해충 진단 과정

이와 같은 지식베이스는 규칙 기반으로 구축되며, 추론 기관은 각 사용자가 declare한 감자 병해충 증상을 바탕으로 감자 병해충 진단을 수행하게 된다. 다음은 본 연구에서 설계 및 구축하고 있는 감자 병해충 추론 기관의 대략적인 flow 이다.



감자 병해충 진단 추론 기관은 병해충 지식베이스에 구축된 병해충 규칙을

활용하여 감자 병해충 이름을 알아내는 기능을 수행한다. 따라서, 사용자가 제공한 병의 주요 증상을 바탕으로 각 병해충 규칙을 검사하여 가장 유사한 병해충 이름을 알아내는 기능을 수행한다. 감자역병 전문가 시스템의 진단 추론 기관은 다음과 같은 기능을 수행하도록 시스템을 설계하고 구현하였다.

- 병해충 증상 입력
- 예상 병해충 목록 생성
- 예상 병해충 목록의 각 병해충의 가능성을 탐색
- 가능성이 가장 높은 병해충 선정

첫 번째로, 사용자는 자신의 감자가 겪고 있는 병해충의 증상을 입력하게 된다. 이 과정에서는 사용자가 품종, 생육 단계, 주요 부위별(잎/줄기/괴경) 주요 증상을 입력하게 된다. 이와 같은 초기 입력은 전문가시스템이 예상 병해충 목록을 생성하는 데 도움이 된다.

두 번째로, 전문가시스템은 입력된 주요 증상으로부터 예상 병해충 목록을 생성하게 된다. 이와 같은 작업을 위해서 추론 기관은 입력된 감자의 주요 증상과 각 규칙의 주증상 부분을 pattern matching하는 작업을 수행한다. 예를 들면, “감자의 잎이 황화 되었다”는 증상이 입력되면, 진단 지식베이스를 검사하여 각 규칙의 주증상 조건문에 “symptom(잎, 색깔, 황화)”라는 증상을 보유한 모든 규칙을 fetch하게 된다.

이 때 fetch된 규칙은 “symptom(잎, 색깔, 황화)”라는 조건문 이외의 다른

조건문을 포함하게 된다. 물론, 하나의 조건문만 가지게 되면 문제가 다르지만, 본 연구에서 구축하고 있는 지식베이스에는 여러 개의 조건문을 가지는 규칙이 대부분이다. 따라서, 추론 기관은 fetch된 각 규칙의 조건문을 다시 테스트하는 과정이 필요하다. 이 과정에서 fetch된 규칙의 조건문을 테스트하기 위해서는 사용자에게 다시 질의하는 과정이 필요할 수 있다. 예를 들면 fetch된 규칙이 다음과 같다고 가정하자.

Rule1000 : If symptom(있, 형태, 말린다) and symptom(있, 색깔, 황화)
 and symptom(줄기, 생장, 직립)
 Then 병명(감자잎 말림 바이러스)

위의 규칙에는 주어진 조건 말고도 두 개의 조건이 더 있다. 즉, symptom(있, 형태, 말린다), symptom(줄기, 생장, 직립) 라는 조건을 확인할 필요가 있다. 따라서, 전문가시스템의 추론 기관은 다음과 같은 순서로 이와 같은 추가 조건을 확인한다.

- 사용자가 추가 조건에 대한 내용을 입력하였는 지를 확인한다. 사용자가 있, 줄기에 대한 해당 속성을 제공하였다면 이를 바탕으로 조건문을 테스트한다.
- 사용자가 해당 속성을 제공하지 않은 경우는 이를 사용자에게 확인하여야한다.
- 사용자에게 확인하는 과정은 크게 두 가지 방식이 있다. 첫 번째는 해당 증상의 사진을 이용하는 방식과 해당 증상을 텍스트로 제공하는 방식이다. 사진을 사용자에게 제공하여 검증

받는 방식이 가장 정확하지만 사진이 없는 경우는 텍스트를 사용자에게 제공한다.

- 위의 예에서는 잎이 말리는 사진이 데이터베이스에 저장되었는가를 확인하고 있으면 이를 사용자에게 제공하여 확인을 받는다. 사진이 없는 경우는 텍스트를 사용자에게 제공한다.

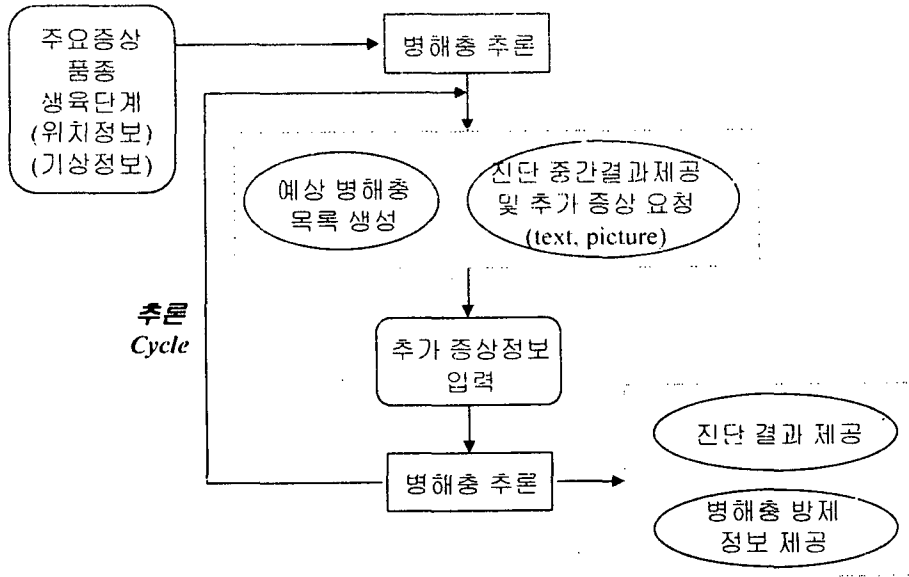
이와 같은 순서로 각 규칙의 조건문을 테스트하여 예상 병해충 목록을 생성한다. 마지막 단계를 위해서, 테스트할 내용은 해당 symptom에 사진이 있는지를 나타내는 index 지식, 해당 symptom의 텍스트를 나타내는 index 지식이 필요하다. 일반적으로 이와 같은 index 지식은 다음과 같이 두 종류의 predicate로 표현되고 있다.

```
exist_picture(symptom((잎, 형태, 말린다), 병명(감자잎 말림 바이러스),  
                , PictureNumber).
```

```
message(symptom((잎, 형태, 말린다), 병명(감자잎 말림 바이러스),  
          감자의 잎이 말리고 숫가락처럼 말립니까?).
```

따라서, fetch된 규칙으로부터 필요한 두 정보 symptom((잎, 형태, 말린다), 병명(감자잎 말림 바이러스))를 이용하여 사진이 있는 지, 없으면 어떤 텍스트를 사용자에게 제공할 것인지를 알아볼 수 있다.

다음 그림은 본 연구의 추론기관이 감자 병해충 진단을 위해서 추론하는 과정을 보여 주고 있다. 추론 cycle은 반복적으로 병해충의 원인을 파악하게 된다. 추론 cycle은 fetch된 규칙을 실행하기 위해서 조건문을 테스트하고 그 결과에 따라서 가설을 생성하는 과정을 반복한다.



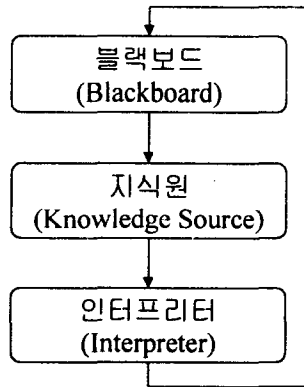
이와 같은 전문가시스템 추론 방식은 블랙보드 추론 방식을 활용하고 있다. 블랙보드 방식의 추론은 추론 과정을 블랙보드에 위치하고 이를 독립된 규칙들이 수행하는 방식을 취하고 있다. 일반적인 블랙보드 추론 방식은 다음과 같다.

2. 감자 병해충 추론 시스템

가. 감자 병해충 추론기관 구성

일반적인 블랙보드 추론기관은 블랙보드(Blackboard), 지식원(Knowledge Source), 인터프리터(Interpreter) 3가지 요소로 구성되어 있다. 블랙보드는

프로그램 진행 중에 발생하는 모든 종류의 동적 상태를 보유하고 있는 것으로 여러 개의 계층구조로 이루어져 있다. 즉, 크기는 사실과 가설의 집합으로 구분되어 있고 각각의 사실과 가설들도 개념화의 정도에 따라서 계층적인 구조를 가지고 있다. 이러한 블랙보드 구조는 문제해결의 과정을 명시적으로 표현하는데는 상당히 좋은 장점이 있기 때문에, 여러 용도의 응용에 이용된다.



두 번째로 지식원이란 블랙보드 추론기관의 지식구조에 해당하는 것으로, 매우 다양한 종류의 형태가 될 수 있다. 즉, 경우에 따라서는 규칙이 될 수도 있고 프레임 구조의 제어 규칙이나 분야 규칙이 될 수도 있다. 그러므로, 일반적인 규칙 시스템에서는 규칙을 지식원으로 간주할 수 있고, 사례기반 시스템에서는 사례가 지식원이 되는 것으로 생각할 수 있다. 그러면, 일반적인 시스템의 규칙이나 사례와 블랙보드 시스템의 지식원의 차이점은 어디에 있는가 하는 것이 중요하다. 이러한 차이점은 블랙보드 시스템에서는 모든 지식원이 상호 독립적이고 모든 지식원은 블랙보드에 있는 상태에 따라서만 트리거 된다는 점이다. 즉, 일반적인 시스템에서는, 사용되는 지식원 사이에 계층적 구조가 있어서 하나의 지식원이 다른 지식원을 트리거하는 경우가

있으나, 블랙보드 시스템에서는 어떠한 지식원도 다른 지식원을 구동시킬 수가 없다. 각각의 지식원은 서로 독립적으로 동작하고, 어떠한 지식원도 다른 지식원을 구동하거나, 다른 지식원으로부터 구동받지도 않는다. 이와 같은 독립성은 블랙보드 시스템의 장점으로 작용하는 경우가 많이 있다. 즉, 필요에 따라서 새로운 지식원을 얼마든지 추가할 수 있는 가능성이 있기 때문에, 시스템의 확장과 수정이 편리한 특성을 가지게 된다.

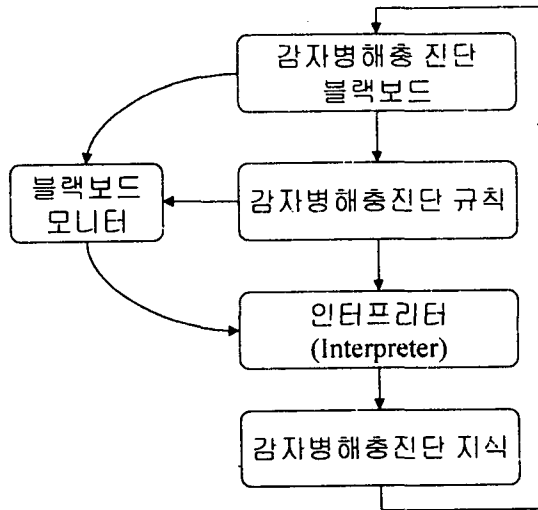
이와 같은 지식원은 현재의 블랙보드 상태에 따라서 각각 실행 가능한 행위를 제시하게 되는데, 이러한 행위로부터 가장 적절한 행위를 선정하여 실행하는 인터프리터가 필요로 한다. 이러한 인터프리터의 중요한 기능은 각각의 지식원이 제시하고 있는 다음의 행위들이 문제 해결에 얼마나 효용성이 있는가를 평가하는 작업으로 이와 같은 과정은 복잡한 스케줄러를 사용하거나 또는 또다른 블랙보드 시스템을 사용하기도 한다.

일반적으로 블랙보드의 구조는 다른 시스템과는 달리 계층적인 구조를 가지고 있다. 즉, 감자병해충 진단 과정을 진행하면서 동적 상태를 개념화하므로, 각각의 개념화 상태에 따라서 블랙보드가 계층을 가지게 된다. 계층적 구조를 가지는 블랙보드의 구조에서, 블랙보드 시스템 내에 있는 각각의 지식원들은 해당하는 블랙보드 계층에 민감하게 동작하는 특성을 가지고 있다. 즉, 모든 지식원이 모든 블랙보드의 계층의 변화에 동작반응을 하는 것이 아니고 해당하는 블랙보드 계층의 변화에만 반응을 보이도록 시스템을 설계하고 있다. 이와 같은 시스템의 설계는 제어의 폭을 침해화하므로써, 쓸데없는 추론을 최소화하는 것을 목적으로 한다.

이러한 블랙보드의 계층적인 구조는 계층만의 변화를 감지하는 모듈의 필요성을 증대시킨다. 즉, 각각의 계층의 변화를 탐지하여, 그 계층의 변화에 관련이 있는 지식원들만을 알아내고 그 지식원들만 인터프리트할 수 있는 기능을 필요로 한다. 이와 같은 기능은 수많은 지식원들을 모두 평가하는 과정이 시간이 소요되는 작업이므로 꼭 필요한 지식원들만을 알아내는 역할을 수행하는 데 필요하다.

블랙보드 시스템에서는 블랙보드 모니터라는 모듈을 활용하여 계층의 변화를 감지하고 해당하는 지식원을 알아내는 역할을 수행한다. 블랙보드 모니터는 계층 변화와 그 변화에 관련이 있는 지식원을 알아내는 테이블을 시스템이 시작하는 과정에서 동적으로 생성하여 사용한다. 블랙보드 모니터는 이와 같은 테이블을 동적으로 생성하기 위해서, 각각의 지식원이 명시적으로 표현하고 있는 데, 이러한 조건은 블랙보드 모니터가 블랙보드 변화와 관련이 있는 지식원들을 알아내는 데에 유용하게 이용된다.

감자역병 전문가시스템은 블랙보드 기반의 감자병해충진단을 위한 시스템을 설계하고 있다. 어떠한 시스템이나 한 번에 완벽한 시스템을 수행하는 데는 상당한 어려움이 있기 마련이다. 이와 같은 문제점을 해결하기 위해서는 지식원을 편리하게 추가하거나 수정할 수 있는 기능이 있어야 한다. 기존의 많은 전문가시스템들은 규칙사이의 관계가 암시적으로 표현되어 있기 때문에 이를 수정 보완하기 위해서는 많은 노력이 필요하다. 그러나 블랙보드 시스템에서는 지식원의 추가나 삭제가 상당히 용이하게 가능하므로 점진적으로 시스템을 확장할 수 있는 가능성이 높게된다.



블랙보드 기반의 감자병해충진단 시스템의 구조를 보면 다음 그림과 같다. 감자병해충진단 블랙보드, 지식원에 해당하는 감자병해충진단 규칙, 인터프리터, 그리고 블랙보드의 변화와 변화된 지식과 관련된 규칙을 알아서 인터프리터에 알려주는 블랙보드 모니터로 구성되어 있다.

나. 블랙보드기반 추론

블랙보드 기반의 감자병해충진단 전문가 시스템은 크게 D(eliberate)단계, S(chedule)단계, E(xecute)단계의 3단계 제어 과정을 반복 수행함으로써 감자병해충을 효율적으로 진단하게 된다. 먼저, 감자병해충진단 블랙보드에 감자병해충에 대한 증상이나 지식이 들어오면 블랙보드 모니터는 그것을 감지하고, 그 지식이 어떤 감자병해충진단 규칙에 해당하는가를 추출하게 된다. 즉, '잎이 말린다'라는 증상이 블랙보드에 들어왔을 때 이것에 해당하는 규칙을 추출하게 된다. 이 과정이 D(eliberate) 단계이다.

이렇게 추출된 규칙 중에서 어느 것을 선택하여야 가장 빨리 감자병충해에 대한 진단을 할 수 있는가를 결정해야 한다. 즉, '잎이 말린다'라는 증상에 대해 관련된 규칙이 여러 개가 있을 경우 그 규칙들 중에서 어느 것이 감자병해충 진단을 하는데 더 중요한가를 결정하는 것이다. 이 과정이 S(schedule)단계이다.

S단계에서 스케줄링된 규칙을 인터프리터에 알려주면 인터프리터는 해당하는 규칙의 조건이 만족하는가를 테스트하여서 그 규칙을 실행시켜서 감자병해충에 대한 결과를 얻어내거나 감자병해충진단을 위한 새로운 지식을 얻어내어 그것을 다시 블랙보드에 보내게 된다. 즉, '잎이 말린다'라는 증상에 관련된 규칙 중에 '잎이 말리고, 잎의 색이 황화되었고, 줄기의 생장이 직립이면 감자잎 말림 바이러스이다'라는 규칙이 있을 때 '잎의 색이 황화되었는가', '줄기의 생장이 직립인가'를 테스트하여 그것이 만족되면 '감자잎 말림 바이러스'라는 감자병충해 진단을 하게 된다. 이것은 새로운 사실로 블랙보드로 들어가게 된다. 이 과정이 E(execute)단계이다. 이렇게, 블랙보드 기반의 감자병해충진단 전문가 시스템은 Deliberate, Schedule, Execute의 3단계를 반복하면서 진단을 하게 된다.

3. 감자 병해충 지식베이스

가. 규칙기반 지식베이스

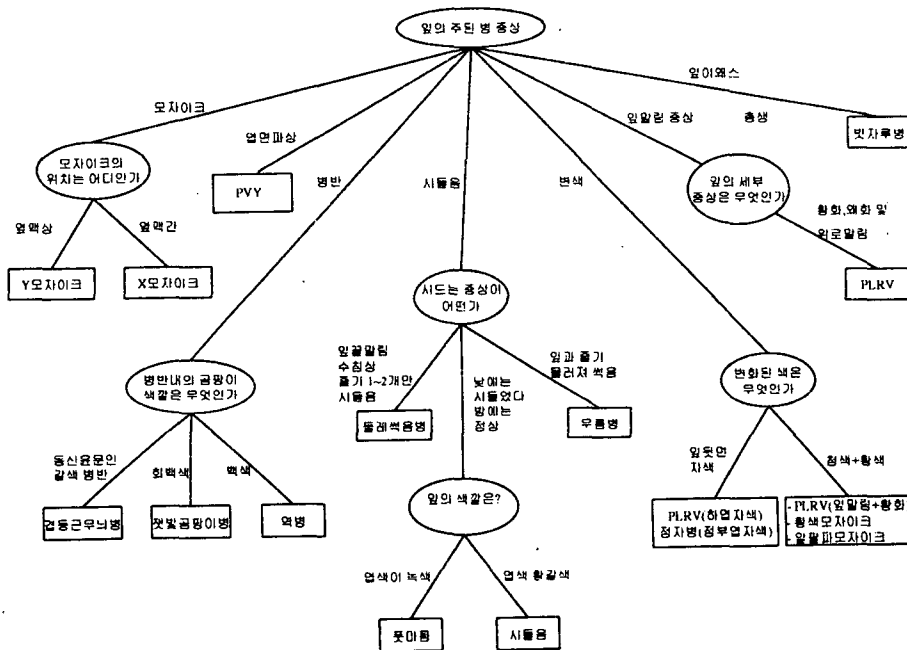
감자역병 전문가 시스템의 지식베이스를 구축하는데 있어 주요한 병에 대한 특징을 추출하는 것이 중요하다. 즉 감자역병을 빠르고 정확하게 알아내기 위해서 사용자에게 앞부분에 질문하는 질문은 매우 중요하다. 즉 결정적

인 질문을 사용자에게 질문을 하고 사용자의 대답에 따라 예상되는 감자역병의 종류를 줄이고 확실시되는 감자 역병을 예상하는 것이 무엇보다 중요하다. 그래서 감자 전문가와 인터뷰를 통한 감자역병 지식을 획득할 때 이 점을 충분히 고려하여 중요한 특징을 추출하였다.

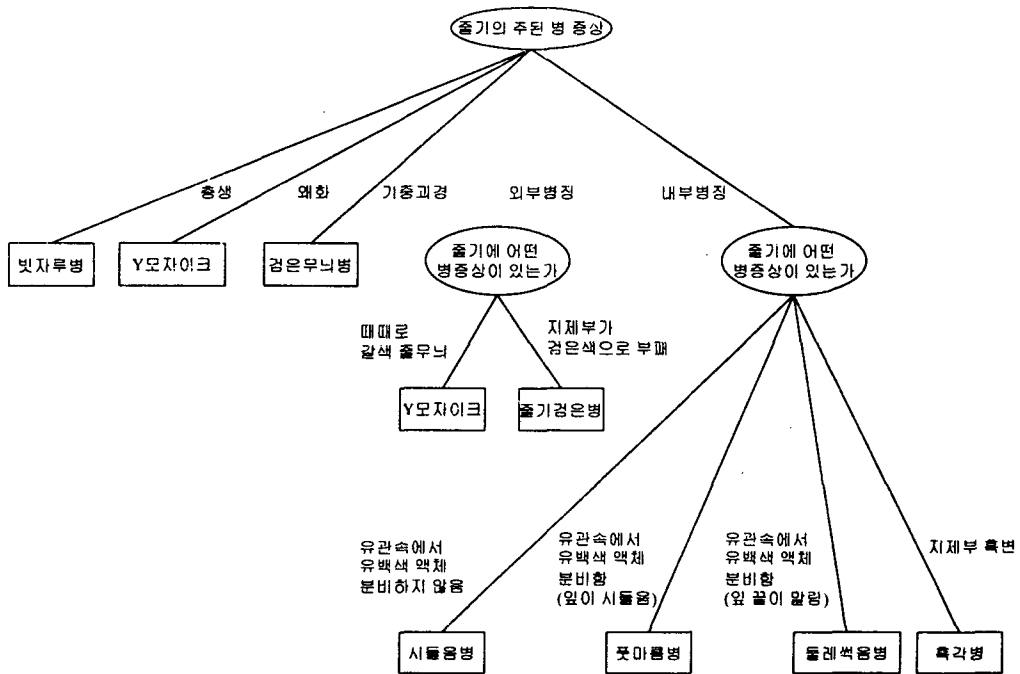
나. 지식베이스 구축방법

고령지 농업시험장의 전문가의 지식을 추출하여 결정 트리를 작성하여 지식베이스 구축에 이용하였다. 고령지 농업시험장의 전문가로부터 병충해의 주된 병증상에 관한 지식을 추출하고 중요한 병증상을 고려하여 결정 트리를 만들었다. 구축된 결정트리는 다음과 같다.

감자역병 결정 트리



<잎이 주된 증상인 병해충 결정 트리>



<줄기가 주된 증상인 병해충 결정 트리>

구축된 결정 트리를 이용하여 지식베이스를 구축한다. 즉 고령지 농업시험장의 전문가로부터 추출한 지식을 규칙화하여 전문가 시스템에 활용하였다. 즉 규칙 기반의 지식베이스를 구축하여 적용하였다. 다음은 규칙기반의 지식베이스의 기본 구조이다.

RuleNumber : If Symptom₁(Type, Attribute, Value)
 and Symptom₂(Type, Attribute, Value)

 and Symptom_n(Type, Attribute, Value)
 Then Diseases(Name)

따라 잎, 줄기, 피경에 주된 병증상이 있는 트리들로 나누었다. 감자 병해충 진단 지식베이스는 우선 구축된 결정 트리를 이용하여 병증상과 병해충진단 결과를 가지고 규칙 기반의 지식베이스형태로 구축하였다. 다음은 구축된 지식베이스의 감자 병해충 진단 Rule이다.

감자역병 진단 Rule

- Rule1000 : If symptom(줄기, 모양, 총생) Then Diseases(빛자루병)
- Rule1001 : If symptom(줄기, 모양, 왜화) Then Diseases(Y모자이크)
- Rule1002 : If symptom(줄기, 모양, 기중피경) Then Diseases(검은무늬병)
- Rule1003 : If symptom(줄기, 병징, 외부) and symptom(줄기, 색깔, 갈색 줄무늬) Then Diseases(Y모자이크)
- Rule1004 : If symptom(줄기, 병징, 외부) and symptom(줄기, 색깔, 지체부 검은색) Then Diseases(줄기검은병)
- Rule1005 : If symptom(줄기, 병징, 내부) and symptom(유관, 액체, 유백색 액체 분비 없음) Then Diseases(시들음병)
- Rule1006 : If symptom(줄기, 병징, 내부) and symptom(유관, 액체, 유백색 액체 분비 있음) symptom(잎, 모양, 시들음) Then Diseases(풋마름병)
- Rule1007 : If symptom(줄기, 병징, 내부) and symptom(유관, 액체, 유백색 액체 분비 있음) symptom(잎, 모양, 끝이 말림)
Then Diseases(둘레 썩음병)
- Rule1008 : If symptom(줄기, 병징, 내부) and symptom(지체부, 색깔, 흑변)
Then Diseases(흑각병)

- Rule1009 : If symptom(잎, 병징, 모자이크) and symptom(모자이크, 위치, 옆맥상) Then Diseases(Y모자이크)
- Rule1010 : If symptom(잎, 병징, 모자이크) and symptom(모자이크, 위치, 옆맥간) Then Diseases(X모자이크)
- Rule1011 : If symptom(잎, 병징, 옆면과상) Then Diseases(PVY)
- Rule1012 : If symptom(잎, 병징, 병반) and symptom(병반, 색깔, 동선
운문인 갈색) Then Diseases(겹등근무늬병)
- Rule1013 : If symptom(잎, 병징, 병반) and symptom(병반, 색깔, 회백색)
Then Diseases(잿빛곰팡이병)
- Rule1014 : If symptom(잎, 병징, 병반) and symptom(병반, 색깔, 백색)
Then Diseases(역병)
- Rule1015 : If symptom(잎, 모양, 시들음) and symptom(잎, 시들은 모양,
끝말림 수침상 줄기 1~2개만 시들음) Then Diseases(둘레썩음병)
- Rule1016 : If symptom(잎, 모양, 시들음) and symptom(잎, 시들은 모양,
낮에는 시들었다 밤에는 정상) and symptom(잎, 색깔, 엽색이 녹색)
Then Diseases(꽃마름)
- Rule1017 : If symptom(잎, 모양, 시들음) and symptom(잎, 시들은 모양,
낮에는 시들었다 밤에는 정상) and symptom(잎, 색깔, 엽색이
황갈색) Then Diseases(시들음)
- Rule1018 : If symptom(잎, 모양, 시들음) and symptom(잎, 시들은 모양,
앞과 줄기 물러져 썩음) Then Diseases(무름병)
- Rule1019 : If symptom(잎, 색깔, 변색) and symptom(잎, 색깔, 앞뒷면
하엽 자색) Then Diseases(PLRV)
- Rule1020 : If symptom(잎, 색깔, 변색) and symptom(잎, 색깔, 앞뒷면 정

부엽 자색) Then Diseases(정자병)

Rule1021 : If symptom(잎, 색깔, 변색) and symptom(잎, 색깔, 청색+황색)
Then Diseases(PLRV)

Rule1022 : If symptom(잎, 색깔, 변색) and symptom(잎, 색깔, 청색+
황색) Then Diseases(황색모자이크)

Rule1023 : If symptom(잎, 색깔, 변색) and symptom(잎, 색깔, 청색+
황색) Then Diseases(알팔파모자이크)

Rule1024 : If symptom(잎, 모양, 말림) and symptom(잎, 모양, 왜화)
and symptom(잎, 색깔, 황화) Then Diseases(PLRV)

Rule1025 : If symptom(잎, 모양, 왜소) and symptom(잎, 모양, 총생)
Then Diseases(빛자루병)

Rule1026 : If symptom(괴경, 병징, 외부) and symptom(괴경, 모양, 소
형흑색균핵) Then Diseases(검은무늬병)

Rule1027 : If symptom(괴경, 병징, 외부) and symptom(괴경, 모양, 윤
문괴저) and symptom(괴경, 색깔, 갈색) Then Diseases
(바이러스(PVY,PMTV))

Rule1028 : If symptom(괴경, 병징, 외부) and symptom(괴경, 모양, 피 목비
대) Then Diseases(무름병)

Rule1029 : If symptom(괴경, 병징, 외부) and symptom(괴경, 모양, 피 목비
대) Then Diseases(분상창가병)

Rule1030 : If symptom(괴경, 병징, 외부) and symptom(괴경, 색깔, 변색)
and symptom(괴경, 색깔, 갈색) Then Diseases(역병)

Rule1031 : If symptom(괴경, 병징, 외부) and symptom(괴경, 색깔, 변

- 색) and symptom(괴경, 색깔, 흑색) Then Diseases(하역병)
- Rule1032 : If symptom(괴경, 병징, 외부) and symptom(괴경, 색깔, 변색)
and symptom(괴경, 색깔, 은백색) Then Diseases(은무늬병)
- Rule1033 : If symptom(괴경, 병징, 외부) and symptom(괴경, 색깔, 변색)
and symptom(괴경, 모양, 동심원함몰) Then Diseases(건부병)
- Rule1034 : If symptom(괴경, 병징, 외부) and symptom(괴경, 색깔, 변색)
and symptom(괴경, 모양, 미소균핵) and symptom(괴경, 미소균
핵, 흑색) Then Diseases(탄저병)
- Rule1035 : If symptom(괴경, 병징, 외부) and symptom(괴경, 모양, 돌출병
반) and symptom(괴경, 모양, 융기 함몰 코르크화)
and symptom(괴경, 색깔, 흑갈색) Then Diseases(더탱이병)
- Rule1036 : If symptom(괴경, 병징, 외부) and symptom(괴경, 색깔, 눈변색)
Then Diseases(시들음병)
- Rule1037 : If symptom(괴경, 병징, 외부) and symptom(괴경, 색깔, 복지변
색) Then Diseases(꽃마름병)
- Rule1038 : If symptom(괴경, 병징, 외부) and symptom(괴경, 색깔, 복지변
색) Then Diseases(들레씩음병)
- Rule1039 : If symptom(괴경, 병징, 외부) and symptom(괴경, 색깔, 복지변
색) Then Diseases(균핵병)
- Rule1040 : If symptom(괴경, 병징, 외부) and symptom(괴경, 색깔, 복지변
색) Then Diseases(시들음병)
- Rule1041 : If symptom(괴경, 병징, 외부) and symptom(괴경, 색깔, 복지변
색) Then Diseases(거부병)
- Rule1042 : If symptom(괴경, 병징, 내부) and symptom(괴경, 색깔, 괴

하내부 흑청색) Then Diseases(흑점병)

Rule1043 : If symptom(괴경, 병징, 내부) and symptom(괴경, 모양, 괴경내부(활형)괴저) Then Diseases(바이러스(TRV, PMTV))

Rule1044 : If symptom(괴경, 병징, 내부) and symptom(괴경, 색깔, 복지부착면 갈변) Then Diseases(바이러스 세균 곰팡이 제조제 피해)

Rule1045 : If symptom(괴경, 병징, 내부) and symptom(괴경, 모양, 내부경화 및 괴저) Then Diseases(내부갈색반점병)

Rule1046 : If symptom(괴경, 병징, 내부) and symptom(괴경, 모양, 내부경화 및 괴저) Then Diseases(내부갈색반점병)

Rule1047 : If symptom(괴경, 병징, 내부) and symptom(괴경, 모양, 내부경화 및 괴저) Then Diseases(망상괴저병)

Rule1048 : If symptom(괴경, 병징, 내부) and symptom(괴경, 모양, 내부경화 및 괴저) Then Diseases(흑색심부병)

Rule1049 : If symptom(괴경, 병징, 내부) and symptom(괴경, 색깔, 유관속변색) and symptom(유관, 액체, 유관색 점질물분비) and symptom(괴경, 색깔, 밀크색) Then Diseases(두레씩음병)

Rule1050 : If symptom(괴경, 병징, 내부) and symptom(괴경, 색깔, 유관속변색) and symptom(유관, 액체, 유관색 점질물분비) and symptom(괴경, 색깔, 회백색) Then Diseases(꽃마름병)

Rule1051 : If symptom(괴경, 병징, 내부) and symptom(괴경, 색깔, 유관속변색) and symptom(유관, 액체, 유관색 점질물분비) and symptom(괴경, 색깔, 회백색) Then Diseases(꽃마름병)

Rule1052 : If symptom(괴경, 병징, 내부) and symptom(괴경, 색깔, 유관속변색) and symptom(괴경, 색깔, 흑변) Then Diseases(저온장애)

- Rule1053 : If symptom(괴경, 병징, 내부) and symptom(괴경, 색깔, 유관속변색) and symptom(유관, 색깔, 갈색경화)
Then Diseases(시들음병)
- Rule1054 : If symptom(괴경, 병징, 내부) and symptom(괴경, 색깔, 유관속변색) and symptom(육질부, 색깔, 갈색반점)
Then Diseases(내부갈색반점)
- Rule1055 : If symptom(괴경, 병징, 내부) and symptom(괴경, 모양, 중심부공동 및 열 개) Then Diseases(내부갈색반점)
- Rule1056 : If symptom(괴경, 병징, 부패) and symptom(괴경, 모양, 표피균열 없음) Then Diseases(꽃마름병)
- Rule1057 : If symptom(괴경, 병징, 부패) and symptom(괴경, 모양, 표피균열 있음) and symptom(괴경, 색깔, 변색)
Then Diseases(둘레썩음병)
- Rule1058 : If symptom(괴경, 병징, 부패) and symptom(괴경, 모양, 연질공동) Then Diseases(건부병)
- Rule1059 : If symptom(괴경, 병징, 부패) and symptom(괴경, 모양, 연질공동) Then Diseases(균핵병)
- Rule1060 : If symptom(괴경, 병징, 부패) and symptom(괴경, 색깔, 회색변색) Then Diseases(저온장애)
- Rule1061 : If symptom(괴경, 병징, 부패) and symptom(괴경, 색깔, 회색변색) Then Diseases(흑색심부병)
- Rule1062 : If symptom(괴경, 병징, 부패) and symptom(괴경, 모양, 수침상) Then Diseases(홍색부패)
- Rule1063 : If symptom(괴경, 병징, 부패) and symptom(괴경, 모양, 수

침상) Then Diseases(젯빛곰팡이병)

제4절 인터넷에서의 전문가 시스템 인터페이스

1. 멀티미디어 원격 진단 시스템

가. 원격 진단 시스템

감자 병해충 진단 전문가시스템을 Java 언어로 구축함으로써 WWW에 연동할 수 있도록 하였다. 따라서, 원거리에 있는 감자 재배 농민은 인터넷을 이용할 수 있는 경우에 고�령지 시험장의 URL을 입력하여 바로 전문가시스템을 구축할 수 있도록 시스템을 구축하였다.

다음 section에 있는 바와 같이 메뉴 기반의 텍스트 전달 기능, 감자 병해충에 대한 정지화상 전달 기능등을 가지고 있다. 이와같은 원격 진단 전문가시스템은 고�령지 시험장의 main 시스템에 연결하여 감자 재배 농민이 main 시스템을 구동하는 과정에서 필요에 따라 전문가시스템을 구동할 수 있도록 시스템을 구축하였다.

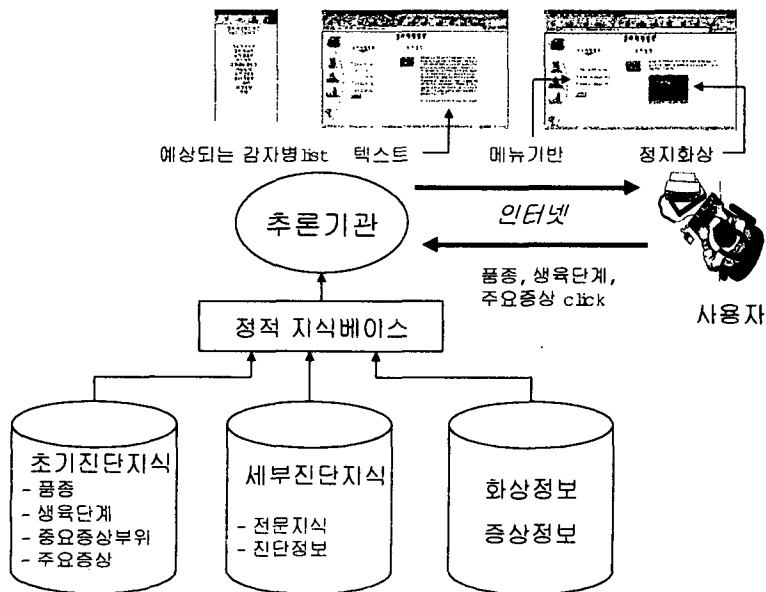
개발된 시스템은 사용자가 인터넷을 통해서 시스템을 구동한 후에 전문가시스템과 상호 작용을 하도록 설계하였다. 즉, 사용자가 감자의 주요 병해충 증상을 원격으로 제공하면 이와 같은 정보를 바탕으로 전문가시스템이 정지화상과 텍스트를 원거리에 있는 사용자에게 바로 전달하는 과정을 반복함으로써 감자 병해충을 진단하는 방식을 취하고 있다.

나. 멀티미디어 인터페이스

인터넷상에서 사용자는 감자 병해충 전문가시스템에 URL을 통해서 접속하

게 된다. 전문가시스템은 사용자로부터 menu방식의 입력을 통해 추론하게 되며 사용자에게 텍스트, 정지화상, 메뉴등을 제공한다. 사용자는 초기에 품종, 생육단계, 주요증상이 있는 부위, 주요 증상 등을 전문가시스템에게 주며 전문가 시스템은 사용자가 알기 쉽게 정지화상과 상세 설명 정보를 제공하여 사용자로 하여금 확실한 병을 알 수 있도록 한다.

전문가시스템은 정적지식베이스로써 초기진단지식인 품종, 생육단계, 중요 증상 부위, 주요증상을 가지고 예상되는 감자역병을 추론한다. 감자역병 전문가시스템은 추론에 필요한 정보를 사용자에게 질의하게 되고 사용자는 여기서 더 자세한 정보를 전문가시스템에게 알려주면 전문가시스템은 세부진단 지식을 이용하여 사용자에게 감자역병을 알려주게 된다. 아래의 그림은 감자 병해충 전문가시스템이 사용자로부터 인터넷을 통해 사용되는 것을 나타낸다.

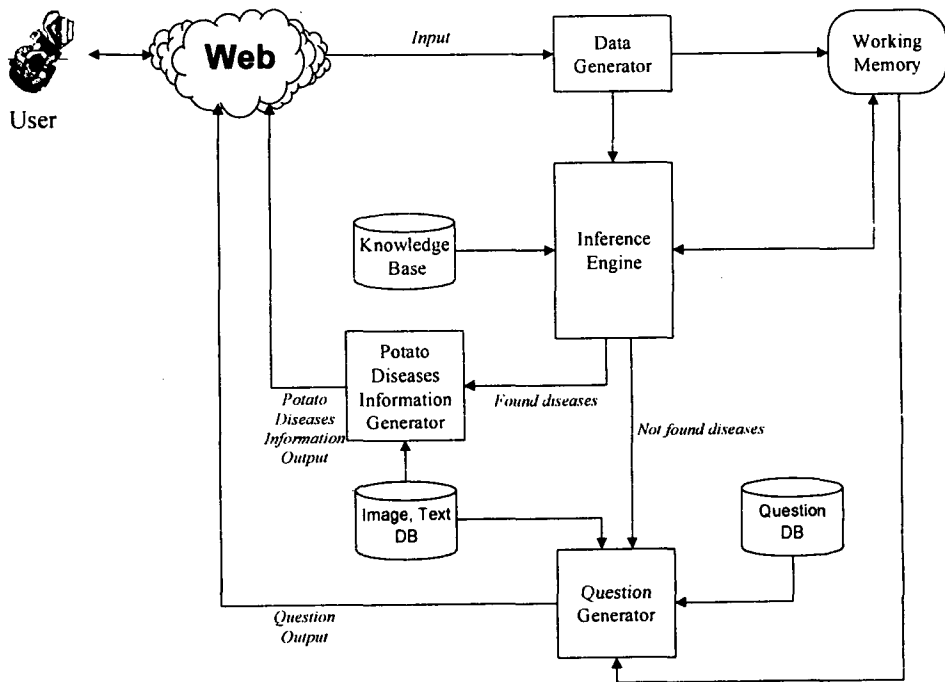


멀티미디어기반 사용자 인터페이스

2. 인터넷에서의 전문가 시스템 구조

가. 인터넷 기반 시스템 구조

감자역병 전문가 시스템은 인터넷을 통하여 사용자와 상호작용을 함으로써 감자 역병을 추론하는 시스템 구조를 가진다. Web에서 사용자와 상호작용하기 위해서 감자역병 추론모듈뿐만 아니라 인터넷 통신 모듈이 필요하다. 아래 그림은 web에서 사용자와 상호 작용하면서 감자역병을 추론할 수 있는 시스템 구조이다. 아래 구조에서 Data Generator, Potato Diseases Information Generator, Question Generator가 인터넷에서 사용자와 상호 작용하면서 추론할 수 있도록 도와주는 모듈이다.



원격 진단 전문가시스템 구조도

나. 사용자 인터페이스 구조

Data Generator는 사용자가 전문가시스템의 URL을 이용하여 접속하여 전문가시스템과 상호작용 할 때 이용된다. 즉 사용자가 전문가시스템에게 정보를 제공하면 전문가시스템은 그 정보를 이용하여 전문가시스템이 추론 할 수 있도록 정보를 가공하거나 저장한다. 즉 사용자가 전문가시스템에게 정보를 제공하고 이 정보를 받아 Data Generator가 정보를 알맞게 저장한다. 이런 작업을 한 후 전문가시스템은 추론엔진과 지식베이스를 이용하여 감자의 병을 추론하거나 정보가 부족하여 추론하기 어려울 경우 부족한 정보를 제공하여 사용자에게 질문을 할 수 있게 만든다.

Potato Diseases Information Generator는 사용자에게 감자역병에 관한 정보를 인터넷을 통하여 제공한다. 즉 감자역병의 정지화상과 텍스트를 사용자에게 제공함으로써 사용자의 감자에 걸린 병을 쉽게 이해할 수 있도록 한다. Potato Diseases Information Generator는 추론엔진으로부터 추론된 감자 역병정보만 입력 값으로 받게 되며 출력은 사용자에게 감자역병 정보만 제공하고 질문은 만들지 않는다.

Question Generator는 추론엔진이 감자역병을 추론할 때 추론에 필요한 정보가 부족하여 사용자에게 질문을 만들어 인터넷을 통하여 제공하는 역할을 한다. 추론엔진은 지식베이스를 이용하여 추론을 함으로 추론에 필요한 부족한 지식 또한 알 수 있다. 그래서 이 부족한 정보를 Question Generator에게 주면 정지화상과 텍스트를 이용하여 사용자가 쉽게 이해할 수 있는 형태로 제공하게 되며 사용자의 질문에 대한 답을 얻을 수 있도록 한다.

제5절 전문가 시스템 구현

감자역병 전문가 시스템은 전문가의 지식을 시스템에서 사용할 수 있게 하는 모듈과 전문가 지식을 기반으로 추론하는 모듈로 구성되어 있다. 또한 효율적으로 추론하기 위해 사용자가 쉽게 이해할 수 있도록 정지화상 정보와 텍스트를 메뉴기반의 방식으로 질문과 설명을 만드는 모듈이 필요하다. 그리고 인터넷기반의 전문가 시스템이므로 네트워크를 이용하여 사용자에게 정보를 전달해 주고받을 수 있는 통신모듈이 필요하다.

1. 감자 병해충 진단 모듈

가. 인터넷을 이용한 사용자 입력 모듈

감자 역병 전문가 시스템을 인터넷을 통하여 제공하기 위해서 java 언어를 사용하여 인터넷 연동 모듈을 구축하였다. Java에서 web을 이용한 프로그래밍을 지원하기 위해 제공되는 java web server를 이용하여 사용자와 전문가 시스템이 정보를 교환할 수 있도록 하였다.

```
// 사용자로부터 감자의 병 증상을 입력받는 부분이며 입력받은 // 정  
// 보를 추론 모듈로 보내주는 역할을 한다.
```

```
public class DataGenerator extends HttpServlet{
```

```
String value = "";
```

```
// 인터넷에서 Post와 Get방식을 이용하여 정보를 받는데 그 중에
```



```

// Get방식을 이용하여 사용자 정보를 얻는다.
public void doGet(HttpServletRequest request, HttpServletResponse
                response) throws ServletException, IOException
{
    // 받은 정보를 사용하기 위해 가공을 한다.
    PrintStream out = new PrintStream(response.getOutputStream());
    Enumeration values = request.getParameterNames();

    // 사용자로부터 입력받은 감자역병 증상을 추론을 하기 위해
    // Working Memory에 저장을 한다.
    String WKmemory = request.getParameter("data");

    // 추론을 하기 위해 Working Memory에 있는 정보를 얻는다.
    StringTokenizer st = new StringTokenizer( WKmemory , "&");

    // Working Memory에 있는 정보등을 이용하여 추론엔진에게 감자병
    // 을 추론할 수 있도록 필요한 정보를 제공한다.
    InferenceEngine(query, num, WKmemory, out);
}

```

나. 감자역병 추론 모듈

```

// 사용자로부터 입력받은 정보를 이용하여 감자역병을 추론한다. 감
// 자역병이 추론되면 사용자에게 감자역병 정보를 제공하고 추론하

```

```

// 기에 불충분하면 필요한 정보를 사용자에게 다시 질문한다.
void InferenceEngine(String query, int num, String WKmemory,
    PrintStream out) throws java.io.UnsupportedEncodingException{

// 추론하기 위한 정보를 JDBC를 이용하여 데이터 베이스에서
// 얻는다.
    DBConnect DB = new DBConnect();

// 추론된 결과를 사용자에게 징지화상과 텍스트로 전송하기 위한
// 모듈이다. 추론엔진이 불충분한 정보의 내용을 알아내고 필요한
// 정보를 사용자에게 적절하게 얻기 위해 질문을 만든다.
    OutputGenerator OP = new OutputGenerator();

// 감자역병이 추론되면 사용자에게 추론된 감자역병 정보를 제공
// 한다.
    OP.FoundDiseases(value, out);

// 감자역병을 추론하지 못해서 추론에 필요한 정보를 사용자로 부
// 터 얻기 위해 질문을 한다.
    OP.QuestionGenerator(query, num ,WKmemory, out);
}

```

다. 정보 획득을 위한 질의 모듈

```

// 사용자에게 제공할 징지화상과 텍스트를 얻는다. 또한 사용자에게
// 얻을 정보를 위해 질문을 하기 위한 정보도 가진다. 추론엔진이

```

```

// 필요한 정보를 요청하면 이 정보를 얻기 위한 질문을 질문 DB에서
// 얻는다.

public DBConnect()throws SQLException{

    try{

        // JDBC를 이용하여 DB에 연결한다.

        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

        // potato라는 DB이름으로 연결한다. potato에는 화상정보와 텍스
        // 트 정보, 질문 정보 추론에 필요한 지식이 들어 있다.

        con = DriverManager.getConnection("jdbc:odbc:potato");

        stmt = con.createStatement();

    }

    // 연결된 DB를 끊기 위해한 모듈이다. 정보를 얻기 위해 DB에 접속
    // 하고 정보를 얻고 나서는 DB와의 연결을 끊는다.

    public void destroy(){

        try{

            stmt.close();

            con.close();

        }catch(SQLException e){

            System.out.println("executeQuery error ");

        }

    }

```

```

// 텍스트나 질문을 DB에 넣기 위해서 한글처리를 한다.
static String toDB(String str) throws java.io.Unsupported
    EncodingException
{
    String n = new String(str.getBytes("KSC5601"), "8859_1");
    return n;
}

// 텍스트나 질문을 DB에서 얻기 위해서 한글처리를 한다.
static String fromDB(String str) throws java.io.Unsupported
    EncodingException
{
    String n = new String(str.getBytes("8859_1"), "KSC5601");
    return n;
}
}

```

2. 사용자 인터페이스 모듈

감자역병을 추론하기 위해서 사용자에게 정보를 받는다. 정보를 받을 때 정지화상과 텍스트를 이용하여 사용자가 메뉴에서 선택함으로써 쉽게 사용자로부터 추론에 필요한 정보를 획득하게 된다. 또한 추론된 감자역병에 대한 정보를 사용자에게 제공해주는 인터페이스 모듈이다. 인터넷기반이므로 web에서의 전송 모듈이 포함되어 있다.

가. 멀티미디어 기반 사용자 인터페이스 모듈

```
// 추론엔진에서 감자역병을 추론하다가 필요한 정보를 Question  
// Generator에게 전달한다. 그러면 QuestionGenerator에서 감자역  
// 병을 추론하기 위해서 질문을 만든다. 질문을 만들 때 사용자가  
// 메뉴방식으로 선택하게 하며 정지화상 정보와 설명 텍스트를 이  
// 용하여 정확한 정보를 얻는다.
```

```
public void QuestionGenerator(String query,int num, String  
WKmemory, PrintStream out)throws java.io.Unsupported  
EncodingException {  
try{  
  
// 사용자에게 질문할 내용을 Database로 구축했으므로 Database  
// 에 연결을 한다.  
DBConnect DB = new DBConnect();  
DBConnect DB_ques = new DBConnect();  
  
String address = "<form  
action=http://203.253.21.192:8080/servlet/DataGenerator  
method=GET>";  
// 질문을 가져올 준비를 한다.  
getAttribute = "cond" + num;  
ResultSet rs = DB.stmt.executeQuery(query);
```

```
//인터넷 기반의 시스템 이므로 감자역병 전문가 시스템이 있는
//인터넷 주소를 이용한다.
```

```
out.println(address);
```

```
out.println("<input type=hidden name=memory value=no>");
```

```
que_query = "SELECT * FROM qu WHERE symptom = '" +
            WKmemory + "_main'";
```

```
ResultSet que_m = DB_ques.stmt.executeQuery(que_query);
```

```
// 질문의 기본 틀인 HTML 코드이다. 어떤 질문이든지 변하지 않
// 고 삽입되는 코드이다.
```

```
out.println("<html>");
```

```
out.println("<head><title>감자 역병 전문가 시스템
            </title>");
```

```
out.println("</head>");
```

```
out.println("<body background=\\'/pic/back2.jpg\\'>");
```

```
out.println("<font color=#000000>");
```

```
que_m.next();
```

```
// 사용자에게 질문할 내용을 가져온다. 메뉴기반 방식이므로 중
// 요 질문과 그 질문에 대한 메뉴를 사용자에게 보내 줌으로 인
// 해 정확한 정보를 얻는다. 여기서는 중요질문을 얻는다.
```

```
String m_question = DB_ques.fromDB(que_m.getString
            ("main_question"));
```

```

// 질문을 사용자가 볼수 있도록 보낸다.
out.println("<font color=green size=2 >" + m_question +
"</font>");
out.println("<br><br>");

out.println("<font color=#000000 size=2>");
int select = 0;

// 사용자가 질문을 읽고 메뉴에서 질문의 대답을 선택할 수 있
// 도록 대답을 메뉴형식으로 보낸다.
while(rs.next())
{
    value = rs.getString(getattribute);
    if(!value.equals(same_value))
    {
        // 메뉴형식의 대답을 만들기 위해 Database와 연결하여
        // 해당 대답들을 가지고 온다.
        que_query = "SELECT * FROM qu WHERE symptom = '" +
            WKmemory + "&" + value + "'";
        ResultSet que_rs = DB_ques.stmt.executeQuery
            (que_query);
        que_rs.next();
        String question = DB_ques.fromDB(que_rs.getString

```

```

        ("question"));

// 라디오 버튼을 사용하여 사용자가 선택할 수 있도록 한
// 다.
out.print("<input type='radio' name=data
           value="+WKmemory + "&" + value);

if(select == 0)
out.print(" CHECKED ");

// 사용자가 선택한 대답이 감자 역병 전문가 시스템으로
// 올 수 있도록 한다.
out.println(" ><a href=\"http://203.253.21.192:8080/
           explanation/" + WKmemory + "&" + value + ".html\"
           target=\"frame1\">" + question + "</a>");

select = 1;
out.println("<br><br>");
same_value = value;
}
}

```

// Database와 연결된 것을 끊는다.


```

DB.destroy();
DB_ques.destroy();
out.println("</font>");

// 사용자가 라디오 버튼을 선택하고 감자 역병 전문가 시스템에
// 게 선택된 대답을 보낸다.
out.println("<input type='submit' value=' 선택 '></form>");
out.println("</body></html>");

}catch(SQLException e){
    System.out.println("executeQuery error ");
}
}

나. 추론 결과 생성 모듈
// 감자 역병이 발견되었을 때 감자역병이 추론되었다는 정보와 감자
// 역병에 대한 상세 정보를 새 창을 이용해서 보여 준다.
public void FoundDiseases(String output, PrintStream out)throws
java.io.UnsupportedEncodingException {

    // 감자역병 추론 사실을 사용자에게 알려주는 내용
    out.println("<html>");
    out.println("<head>");
    out.println("<title>감자병이 추론 되었습니다.</title>");
    out.println("<script>");

```

```

// 사용자에게 감자역병 정보를 정지화상과 텍스트를 이용해서
// 사용자에게 제공하게 된다. 정보가 제공될 때 새로운 창을
// 이용해서 사용자에게 보여 준다.

out.println("open('http://203.253.21.192:8080/explanation
    /_" + output + ".html','popup', 'width=600,height=
    600 ,scrollbars=yes')" );

out.println("</script>");

out.println("</head><body background=\\\"/pic/back2.jpg\\\">
    <font color=#000000>");

out.println("<br><br><br><center>감차 병이 추론 되었습니
다.</center></font>");

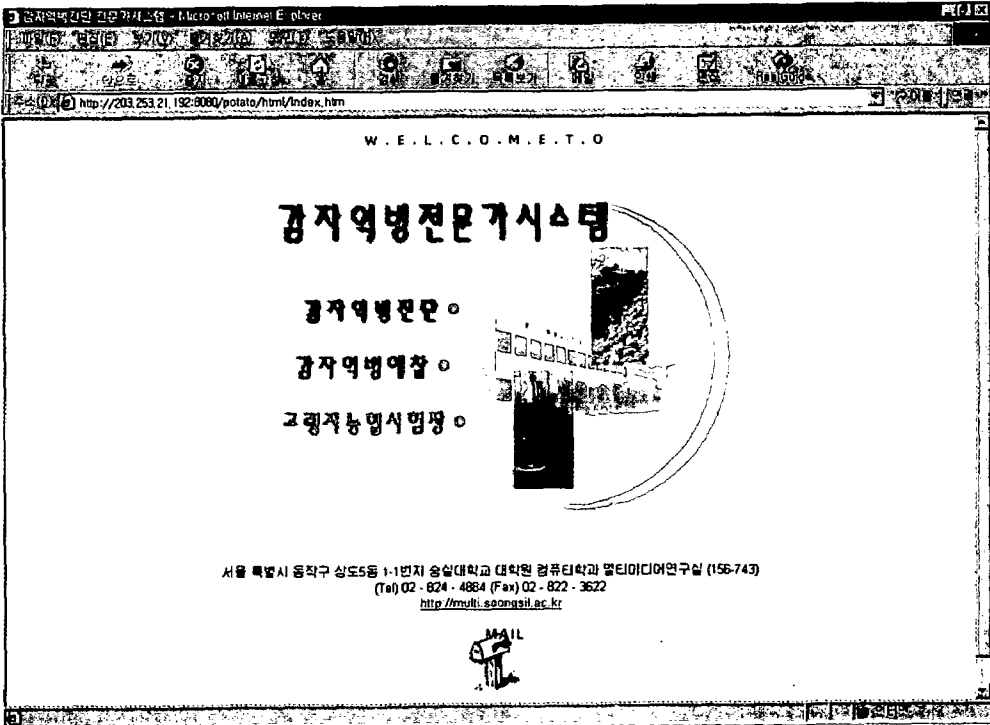
out.println("</body></html>");

}

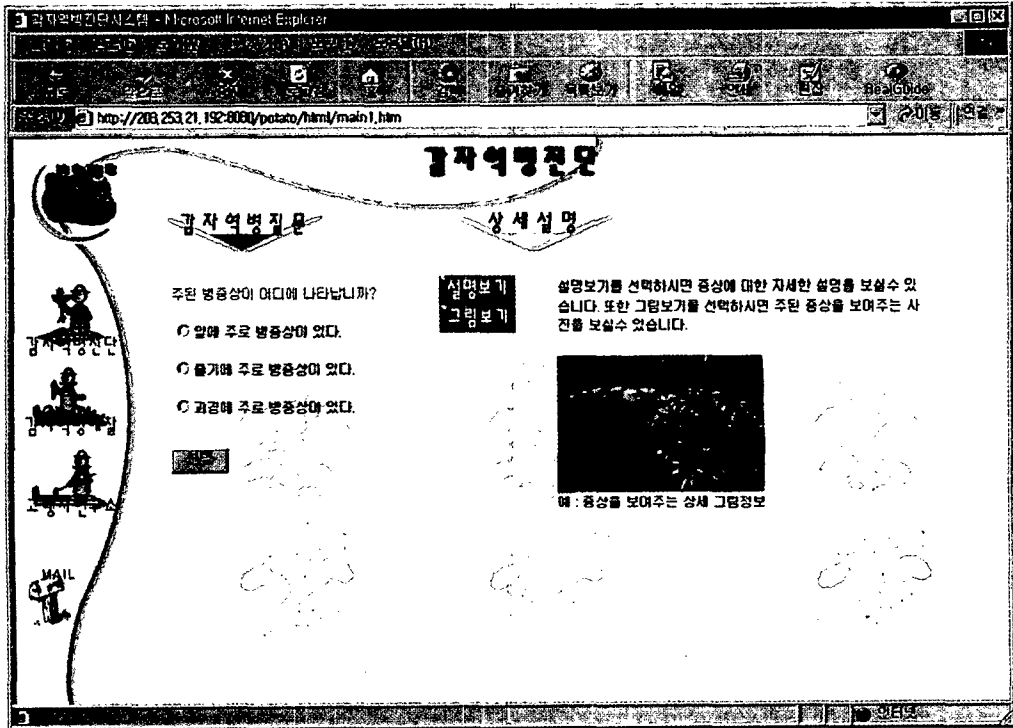
```

3. 구현 결과

감자 병해충 전문가시스템을 구축하고 web에서 실험을 통하여 성능을 확인하였다. 전문가 시스템은 서울의 송실대에 위치하고 있으며 test는 강원도 고령지 농업시험장에서 했으며 네트워크사정에 따라 정지화상이 보이는 시간이 1~2초 정도 소요됐다. 국가 정책사업인 초고속통신망이 완료되는 시기에는 이런 문제점들이 해결될 것으로 생각되며 농민이 쉽게 집에서 감자의 역병에 대하여 진단할 수 있을 거라 생각된다. 아래 그림은 전문가 시스템 초기 화면이다.



감자역병진단 전문가 시스템 안에 3개의 메뉴가 있다. 첫 번째 감자 역병진단은 감자에 역병이 발생했을 때 농민이 쉽게 진단할 수 있도록 하는 메뉴이며 감자 역병 예찰은 감자의 역병을 예찰하여 농약의 살포의 결정등에 쓰인다. 그리고 고령지 농업시험장 메뉴는 사람 전문가에게 직접 농민이 질문등을 하기를 원할 때 이용될 수 있다. 감자역병진단 메뉴를 통해서 감자역병을 진단하게 되면 다음과 같은 화면이 보이게 된다. 사용자가 쉽게 알아 볼수 있도록 감자역병 질문과 질문에 대한 상세 설명을 옆에 위치시켰다. 그리고 상세 설명은 그림보기 설명보기로 감자역병에 대하여 질문을 보충할 때 텍스트와 정지화상을 이용하여 사용자가 정확하게 감자병을 알수 있도록 하였다.



처음 질문은 감자의 주 증상을 질문하는 것으로써 잎과 줄기 괴경 중에서 어디에 병 증상이 제일 많이 보이는 가를 질문하는 것이다. 사용자는 메뉴를 보고 잎, 줄기, 괴경을 선택하면서 설명을 볼 수 있으며 정지화상 또한 볼 수 있는 것이다. 아래는 그림은 줄기에 이상이 있을 경우를 농민이 선택했을 경우 보여 지는 상황이다. 줄기에 총생 증상, 왜화 증상, 기중 괴경 증상, 외부 병징, 내부 병징이 있는 가에 대하여 설명보기를 통하여 텍스트 정보와 그림보기를 이용하여 정지화상을 보면서 사용자에게 양질의 정보를 제공하는 것이다. 아래 그림은 사용자가 기중괴경 증상에 한 텍스트 정보를 확인한 경우와 외부병징의 정지화상을 보는 경우이다. 사용자는 의심이 가는 모든 메뉴를 선택해서 설명과 그림을 본 뒤에 감자 역병을 선택하게 된다.

감자역병진단서식업 - Microsoft Internet Explorer

http://203.253.21.192:8080/potato/html/main1.htm

감자역병진단

감자역병진문

감기의 주된 병증상은 무엇입니까?

- 출혈 증상이 있다.
- 열화 증상이 있다.
- 기온 괴롭 증상이 있다.
- 외부 병징이 있다.
- 내부 병징이 있다.

상세설명

설명보기
그림보기

파종직후 봄에 가장 심하게 피해를 받는데 지하부 썩어 죽으며, 저온과습한 토양에서 특히 출현이 지연된다. 이는 약한 식물체를 얼약하고 고르지 못한 토양에 심었을 때에도 나타내며 수량감소를 초래한다. 출현하는 감자 썩이나 지라는 줄기를 감염시켜 줄기가 고사하거나 반점이 줄기줄기를 부분적으로 감염시켜서 줄기가 고사하거나 반점이 줄기줄기를 부분적으로 또는 전체적으로 에워싸는 등 다양한 증상을 나타낸다. 식물체를 위해 시키거나 정엽부위의 로제트현상, 목질관린 줄기의 코르크층은 과사되고 잎이 적자색을 띠며, 기공과경이 상기고 잎이 위로 말리는 증전현상, 황녹화 현상 등마 식물체에서 심하게 나타난다.

복지의 적갈색 피해가 복지의 절단이나 기현서를 유발

MAIL
이동

감자역병진단서식업 - Microsoft Internet Explorer

http://203.253.21.192:8080/potato/html/main1.htm

감자역병진단

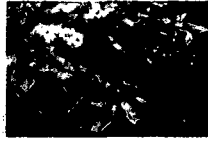
감자역병진문

감기의 주된 병증상은 무엇입니까?

- 출혈 증상이 있다.
- 열화 증상이 있다.
- 기온 괴롭 증상이 있다.
- 외부 병징이 있다.
- 내부 병징이 있다.

상세설명


설명보기
그림보기



MAIL
이동

아래 그림은 사용자가 전문가시스템의 질문에 메뉴를 보면서 대답하는 경우를 반복해서 했을 경우 아래와 같이 감자 병해충을 추론할 수 있다. 다음은 줄기 검은 병에 대한 추론된 결과를 보여 준다. 정지화상과 세부적인 정보를 사용자에게 제공함으로써 사용자가 보고 자신의 감자역병에 대하여 적절한 대처를 할 수 있도록 한다. 결국 사용자는 감자 역병 전문가시스템을 사용함으로써 빠르고 쉽게 사람 전문가의 지식을 이용하여 자신의 감자 역병을 진단할 수 있는 것이다.

줄기검은병(흑각병, Blackleg) - Microsoft Internet Explorer



줄기검은병(흑각병, Blackleg)

- (1) 특성
본 세균병은 감자의 줄기를 가해하고 괴경을 무르게 하며 썩게 하기도 한다. 줄기검은병(E. C. spp. atroseptica)과 무름병(E. C. spp. carotovora)은 같은 세균 속(Erwinia carotovora)에 속하나 아종(Subspecies, 亞種)에 의해 구별한다. 감자 재배하는 어디 서나 볼 수 있다.
- (2) 증상
감자생육 모든 단계에서 나타난다. 이병식물의 줄기는 전형적으로 검은 잉크색으로 부패되며 보통 씨감자 절편이 부패하면서 시작되는 줄기의 지체부까지 진행되거나 때로는 줄기전체가 흑변 부패한다. 이병주는 왜소하거나 생육초기에 뻗뻗해지고 서는 경향이 있다. 지상부 잎은 황화되고 소엽은 끝이 위로 말리며 나중에는 식물체 전체가 시들고 천천히 꾸부러지며 결국 죽게 된다. 어린 줄기들이 침해를 받으며 출현 전에 죽게 된다. 줄기, 열병과 잎들은 비바람, 태풍에 등에 의해 입은 상처를 통해 감염되고 위아래로 병이 퍼져나가 전형적인 줄기검은병 증상을 나타내게 된다. 습한 기후 하에서는 부패하여 흐물흐물하고 끈적끈적해지며 식물전체로 퍼진다. 건조한 조건에서는 이병조직은 마르고 시들시들해져서 줄기의 지체부 아래 부위에 한정된다. 이병식물이 생산한 괴경은 기부에 약한 유관속변색 증상에서 괴경정부가 썩는 증상을 보여준다.
- (3) 병원균
줄기검은병은 주로 세균 Eca와 때로는 무름병균(Ecc)에 의해 발병되는데, 전염원은 주로 씨감자 괴경 내외에 존재하며 파종 후 생육기간 내내 절편을 부패시킬 수 있으며 토양에 수많은 세균을 방출하며 줄기를 감염시킨다. 세균은 감자 생육기간동안 기주의 근권에서 증식 생활한다. 병원균은 괴경에서 월동하며 얼마동안 토양에서 생존할 수 있고 따뜻하고 건조한 조건보다 시원하고 습한 조건에서 보다 오래 생존한다. 이병식물 잔재

제4장 실시간 감자 역병 관리 전문가 시스템

제4장 실시간 감자 역병 관리 전문가 시스템

제1절 서론

1. 연구 배경

강원고랭지는 표고가 높아 여름철에 서늘하고, 바이러스를 전염하는 진딧물의 비래가 적어서 일찍부터 씨감자 주생산지로 발전되어 왔다. 그러나 이와 같은 기후조건은 씨감자 생산의 결정적인 제한요인인 감자疫病의 발생 호적 조건(20℃ 전후의 평균기온과 多雨多霧)과 거의 일치한다. 다행히 역병의 발생 및 진전에 미치는 기상조건의 “규칙성” 덕분에 이 병의 조기예찰이 가능하므로 정확한 예찰에 의한 적절한 방제수단이 마련된다면 합리적인 관리가 가능하다. 현재 국내에서 가동중인 감자역병 예찰정보체제는 대관령 주변의 일부 농장을 대상으로 하고 있으며 또한 병 초발생일 예찰에 한하여 수행되고 있어 종합적인 관리체제로서는 미흡한 실정이다. 이 지역 농업생태계를 고려한 최적 방제계획(spray schedule) 수립을 위해서는 고도의 의사결정능력을 토대로 임상진단 등 실제 농업현장에서 필요로 하는 양질의 정보가 실시간(real-time)에 전달되어야 한다. 이러한 정보를 농민들에게 시각적이고 실시간으로 전달하기 위한 수단 가운데 최근에 급속하게 확산되고 있는 인터넷 (특히, 웹)의 활용이 유망하다.

따라서, 본 연구과제에서 개발한 ‘인터넷 기반 감자역병 전문가시스템’은 기상정보를 이용하여 감자재배 농장의 감자역병 초기발생 및 진전정도를 예측하고, 약제 살포시기, 약제종류 등 최적 방제수단을 전문가시스템에 의하여 결정하며 여기에서 얻어진 지식정보를 인터넷을 통하여 영농현장에 전달

하게 된다. 즉 감자역병의 정확한 조기 예찰과 방제대책 제공으로 약제살포를 최소한으로 줄이면서 경제적으로 손실이 없는 수준에서 역병을 관리하는 시스템이다. 이 시스템은 작물-병-환경을 하나의 시스템으로 간주하면서 합리적인 수준에서 병해충을 관리하는 종합적인 병해충관리(IPM :Integrated Pest Management)체제의 실용형이다. 농업의 경쟁력 강화가 생존차원에서 요구되고 있는 국내현실에서 이 시스템은 타산업분야에서 광범위하게 활용되고 있는 정보통신기술, 인공지능, 그리고 전문가의 지식 결합을 통해 첨단 농업생태계 관리기술을 선보였다.

2. 연구 개발 목적 및 필요성

본 제 3 세부과제의 최종 목표는 농민이 손쉽게 사용할 수 있는 웹기반 역병 예찰 및 방제 전문가 시스템을 위한 사용자 편의위주의 인터페이스와 AWS로부터 수집된 농업 관련 기상 자료를 실시간으로 수집하는데 있다. 또한 역병예찰 및 방제를 위한 소프트웨어를 개발하고, 이를 기반으로 제 1 세부과제에서 제공하는 역병 예찰 모형을 구현하며, 이를 위한 데이터베이스를 구축하고 제 2 세부과제의 전문가 시스템 모듈을 통합하여 인터넷상에서 이들을 구현할 수 있는 Java기반 웹서버시스템을 개발하는데 있다.

구분	연구개발목표	기술개발내용 및 범위
2 차 년도 (당해 연도)	● 웹 서버 기능 및 사용자 인터페 이스 기능 확장	▶ Java Web Server의 Servlet을 이용한 서버 개발 ▶ Java Web Server의 Servlet을 이용한 서버와 JDBC 등의 연동기술 개발 ▶ WWW 및 인터넷을 이용한 멀티미디어 사용자 인터페이스 성능 개선
	● 강원도 고령지 농업 시험장에 위 치한 AWS raw data 구축 및 AWS raw data 입력 및 관리 도 구 개발	▶ 강원도 고령지 농업 시험장에 위치한 AWS raw data 구축 ▶ AWS raw data 자동 입력 시스템 개발 ▶ 구축된 기상정보 raw data를 관리하기 위하여 Oracle Server 환경에서 작동하는 DB시스템 개발
	● 실시간 감자 역병 예찰 모듈 개발	▶ 감자 역병 예찰 알고리즘 조사 ▶ 감자 역병 예찰 알고리즘 설계 및 구현
	● 시스템 통합 및 필드 시험	▶ 감자 역병 예찰 시스템, 감자 역병 전문가 시스템, raw data 관리 DB시스템 등의 통합 ▶ 통합시스템의 필드 시험

표 1-1. 연구개발 목표에 따른 기술개발 내용 및 범위

3. 연구 내용 및 범위

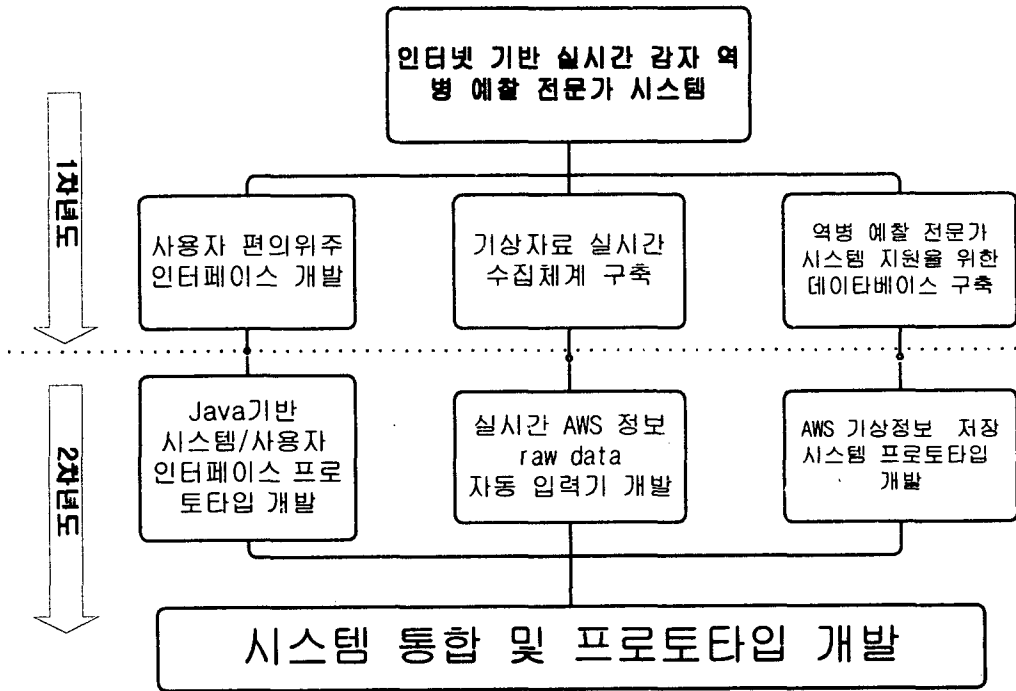


그림 1-1. 본 과제의 연구범위

본 과제의 연구범위는 그림 1-2와 같이 인터넷 기반 클라이언트/서버시스템, 사용자 인터페이스, 실시간 감자 예찰, 기상정보 저장시스템으로 나뉜다. 또한 AWS 기상정보 raw data를 쉽게 구축하고 관리할 수 있는 raw data 입력 및 관리 도구도 포함된다. 2차 년도에 개발한 연구내용은 표 1-2와 같다.

연구 내용	연구 범위
● 웹 서버 기능 및 사용자인터페이스 기능 확장	<ul style="list-style-type: none"> ▶ Java Web Server의 Servlet을 이용한 서버 개발 ▶ Java Web Server의 Servlet을 이용한 서버와 JDBC 등의 연동기술 개발 ▶ WWW 및 인터넷을 이용한 멀티미디어 사용자 인터페이스 성능 개선

<p>● 강원도 고령지 농업 시험장에 위치한 AWS raw data 구축 및 AWS raw data 입력 및 관리 도구 개발</p>	<p>▶ 강원도 고령지 농업 시험장에 위치한 AWS raw data 구축</p> <p>▶ AWS raw data 자동 입력 시스템 개발</p> <p>▶ 구축된 기상정보 raw data를 관리하기 위하여 Oracle Server 환경에서 작동하는 DB시스템 개발</p>
<p>● 실시간 감자 역병 예찰 모듈 개발</p>	<p>▶ 감자 역병 예찰 알고리즘 조사</p> <p>▶ 감자 역병 예찰 알고리즘 설계 및 구현</p>
<p>● 시스템 통합 및 필드 시험</p>	<p>▶ 감자 역병 예찰 시스템, 감자 역병 전문가 시스템, raw data 관리 DB시스템 등의 통합</p> <p>▶ 통합시스템의 필드 시험</p>

표 1-2. 2차 년도 연구 내용 및 범위

일상생활에서 컴퓨터 사용의 기회가 적은 농민들을 위하여 본 시스템은 직관적이고 이해가 쉬우며 이용이 편리하도록 사용자 편의 위주의 인터페이스를 개발하였다. 본 과제에서는 농민들이 인터넷이 연결되어 있는 곳이면 어디서든 접속하여 감자 역병을 예찰하고 방제할 수 있는 서비스를 제공한다. 또한, 그래픽을 이용한 정돈된 화면 구성의 인터페이스는 사용자뿐만 아니라 시스템을 유지하는 시스템 관리자도 편하게 이용할 수 있다. 농장 검색은 지도를 통하여 이루어지고, 그래프로 제공되는 감자 역병 예찰 결과는 사용자가 더욱 이해하기 쉽도록 되어 있다.

무인 기상 관측 장비(AWS)는 지형 및 고도에 따라 다양한 기후적 특징이 나타내는 주요 씨감자 생산 지역에 설치한다. 감자 역병 예찰에 필요한 기상 자료는 시간대별 기온, 강수량, 상대습도로 각 AWS에 설치되어 있는 모뎀과 PC208 프로그램을 이용하여 실시간으로 수집한다. 따라서, 사용자는 감자 역병 예찰에 필요한 가장 최근의 기상 관측 자료를 활용할 수 있다. AWS를 통하여 들어오는 실시간 기상 관측 자료는 현재의 상태를 반영한

병 예찰에 이용될 뿐 아니라 분석과 이력데이터 구축을 위해서 데이터베이스화 되어야 한다. 오랜기간 축적된 기상 관측 raw data의 양은 방대하게 되며, 병 예찰을 위해서 잦은 자료의 검색, 갱신, 삽입, 삭제등의 작업이 필요하다. 본 과제에서는 JDBC를 사용하여 오라클 데이터베이스와 자바로 구축된 인터페이스 모듈과 연동시켰다. 인터넷에서 작동하는 시스템은 현재 인터넷 상에 산재되어 있는 다른 농업관련 정보와의 연계가 용이하므로 사용자에게 더 많은 관련 정보를 제공할 수 있다. 그런데, 자바를 이용한 시스템은 인터넷 프로그래밍이 용이하여 인터넷 상에서의 각 모듈의 통합 및 확장이 쉽다. 자바의 이러한 장점을 살려 본 과제는 순수 자바로 작성되었으며, 동작 서버는 자바를 가장 잘 지원하는 자바 웹 서버로 구성되어 있다.

4. 연구 수행 방법

본 과제의 연구개발목표에 따른 연구수행 방법은 다음과 같다.

연구개발목표	연구수행 방법
웹 서버 기능 및 사용자 인터페이스 기능 확장	<ul style="list-style-type: none"> ▶ Servlet API를 이용하여 구현 ▶ JDBC 관련 API를 이용하여 구현 ▶ 산업디자인학과의 자문으로 사용자 인터페이스의 개선
AWS로부터 추출된 raw data 구축 및 raw data 자동 입력기 개발	<ul style="list-style-type: none"> ▶ AWS 자료를 참조하여 raw data 획득 ▶ Java를 이용하여 raw data 자동입력시스템 개발 ▶ raw data를 Oracle Database에 database로 만들 ▶ Database와 실시간 감자 역병예찰 시스템을 연동하여 Java로 프로토타입 개발
실시간 감자 역병 예찰 모듈 개발	<ul style="list-style-type: none"> ▶ 감자 역병 예찰 알고리즘을 이용하여 Java로 구현 ▶ 실시간으로 생성된 AWS 정보를 반영할 수 있도록 프로그램 구현
시스템 통합 및 필드 시험	<ul style="list-style-type: none"> ▶ 각 서브 시스템을 실제 네트워크 환경에서 테스트하여 필드 테스트

표 1-3. 연구개발 목표에 따른 연구수행 방법

5. 연구 개발의 중요성

가. 통합성, 확장성을 위한 Java 기술 도입

본 과제에서 개발한 실시간 감자 역병 예찰 시스템은 국내에서 최초로 객체 지향 언어이자 플랫폼인 Java 기술을 사용하기 때문에 이미 상용서비스를 제공하는 타 교통정보시스템 뿐만 아니라 인터넷에서 제공되는 이 기종간의 서비스와도 통합이 용이하며, 사용자 요구가 변경되어도 단기간 내에 이를 반영 할 수 있고, 시스템 유지보수 비용이 상대적으로 적게드는 장점을 가지고 있다.

나. 효율적인 사용자 인터페이스 개발

본 연구의 2차 년도에서는 1차 년도에서 개발한 사용자 인터페이스를 더욱 개선하여 시스템 사용이 편리하고, 직관적으로 사용자가 시스템에서 제공하는 내용을 쉽게 이해할 수 있으며, 어느 곳에서도 시스템의 접근이 용이하도록 인간공학, 산업 디자인학, 미학 전문가들의 자문을 구해 사용자 인터페이스를 제공하고 있기 때문에 차원 높은 MMI(Men Machine Interface) 기술을 확보하였다.

다. AWS raw data 관리 DB 시스템 개발

지난 2년간 본 과제를 수행하면서 연구자가 느낀 가장 중요한 점은 우리나라 농업에 중요한 기상자료의 정확한 database 구축이 가장 중요한 사항이라고 판단하였다. 예를 들면 강우량, 시간대별 온도, 일평균 온도, 시간대별 습도, 일평균 습도 등에 대한 상세한 통계적 자료의 구축이 시급한 실정이다. 이러한 자료가 구축이 되어야 전국 AWS를 네트워크로 보고 측정되는 기상 자료를 파라미터로 하여 모델을 구축한 후 시뮬레이션을 수행하여야 농업 기술 발전에 기본적인 모델이 수립될 것으로 판단된다. 이러한 raw data 구축은 많은 경비와 장시간의 노력이 소모되겠지만 현실적인 농업 정보시스템 구축 및 농업 기술 개선을 위해서는 꼭 필요한 사항이다. 이를 위하여 본 과제의 2차 년도에서는 raw data 관리 DB 시스템을 Oracle Server 8 환경 하에서 개발하여 입력, 수정, 삭제 등의 작업을 전체정보와 부분정보를 동시에 단일 화면에 표현하여 실제 사용자 용이하게 수행할 수 있도록 하였다. 이는 변동하는 데이터를 관리자가 직접 눈으로 확인함으로써 이상유무를 판단할 수 있으며 관리자는 각 AWS의 실제 데이터 수치로

작업할 수 있을 뿐만 아니라 raw data의 관리에 드는 많은 경비를 절감할 수 있다.

제2절 연구 개발 수행 내용 및 결과

1. 사용자 및 시스템 인터페이스

본 세부과제의 목표 중 하나는 인터넷에 연결된 다양한 이기종 시스템 환경에서 잘 동작하는 사용자 인터페이스를 개발하기 위하여 요즘 널리 관심을 고조시키고 있는 Java를 이용하여 사용자 및 시스템 인터페이스를 개발하는 것이다. 이는 Web 환경 하에서 작동하는 서버의 Servlet Version과 Web이 아닌 일반적인 TCP/IP 환경에서도 동일한 동작을 하는 raw data 데이터베이스 관리 응용의 Application Version으로 구성되어있다.

2 차년도에서는 1 차년도에서 개발된 Java의 웹 서버 기능의 한 부분인 Servlet을 최신 버전으로 재개발하였다. 또한 Server측에서 Database에 접속하기 위한 방법으로는 Java언어의 JDBC(Java DataBase Connectivity)을 사용하여 Database Server와 접속을 유지하고, 질의를 보내며, 질의에 대한 결과를 제공받게 된다. 모든 시스템 모듈의 통합으로 시스템의 모든 프로그램은 100% 순수 Java 언어로 작성이 되어졌다. 이로 인하여 개발된 시스템은 통합성, 호환성, 확장성, 이식성, 보안성 등의 Java에서 지원하는 모든 장점들을 제공받으며 안정된 서버를 구축할 수 있다. 또한 어떤 클라이언트 환경에서도 동일한 사용자 인터페이스를 유지할 수 있다.

가. Java 기술

Java는 미국의 SUN사에서 개발된 객체지향 인터프리터형의 언어로서 초기에는 가전제품의 제어에 사용하고자 개발되었으나 WWW로의 응용으로

인해 무한한 가능성을 가지게 되었다. Java applet은 기존의 WWW에서 사용되는 HTML문서 내에 삽입되는 형태의 간단한 프로그램으로서 기존의 웹이 마치 그림책과 같은 정적인 형태를 띄는 것에 반해서 인터랙티브하게 상호작용이 가능한 형태의 동적인 효과를 줄 수 있다. 또한 Java 프로그램이 네트워크를 통해 전달되어 클라이언트에서 실행되는 특징을 가지고 있기 때문에 WWW에 새로운 기능을 추가하거나 새로운 프로토콜이 개발되었을 경우 웹 브라우저를 변경할 필요 없이 그 새로운 기능을 수행할 수 있는 Java 프로그램을 같이 전송하여 실행할 수 있게 된다. 즉 기존의 WWW은 데이터만을 전송해 주고 그 데이터를 처리하는 함수들을 클라이언트 측에서 가지고 있어야만 하는 형태이지만, Java를 사용하게 된다면 마치 OOP에서의 클래스가 데이터와 그 데이터를 조작하는 메소드(Method)를 같이 포함하는 형태인 것처럼 그 데이터의 처리기능까지도 같이 전송해주는 형태가 된다는 것이다.

그러므로 Java를 사용하게 된다면 단순히 Java 인터프리터를 포함하고 있는 웹 브라우저만을 가지고도 주식시세를 동적으로 분석하여 보여주거나, 온라인 게임을 구현하거나 심지어는 온라인으로 워드프로세서와 스프레드시트와 같은 애플리케이션 프로그램까지도 사용할 수 있다.

1) Java의 장점 및 특징

Java는 C++의 유사한 구조를 가지고 있으며 비교적 사용하지 않는 특성들을 제거한 간단한 형태의 고급언어로서 필요한 모든 종류의 고급 언어적인 특징을 가지고 있다. 이의 장점은 다음과 같다.

가) 객체지향성 및 단순성

Java는 C++과 유사한 형태의 객체지향 언어이고 기존의 C++ 등의 언어들이 가지고 있는 연산자 재정의, 다중상속, 포인터 오퍼레이션, 다차원 배열 등의 불필요하고 복잡한 기능들을 대폭 삭제하여 배우기 쉽고 단순한 구조를 가지고 있다. 또한 Java는 초기 설계가 TV 등의 가전제품과 같은 단순한 구조 및 적은 용량의 컴퓨터에서 수행되도록 설계되었으므로 시스템에 부하가 적다.

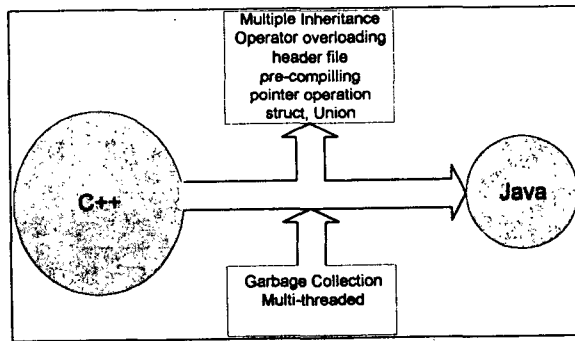


그림 2-1. Java 의 언어적 특성

나) 컴파일된 중간코드의 인터프리터

Java는 소스코드를 컴파일하여 생성된 바이트 코드를 인터프리트해서 수행하는 형태로 구성되어 있으므로 그 인터프리터가 수행될 수 있는 기계라면 그 종류에 관계없이 수행될 수 있어 기계독립적(Machine Independent)이고 클라이언트 측에 전송되는 코드가 컴파일된 형태이므로 그 소스에 대한 보안성을 가지고 있다.

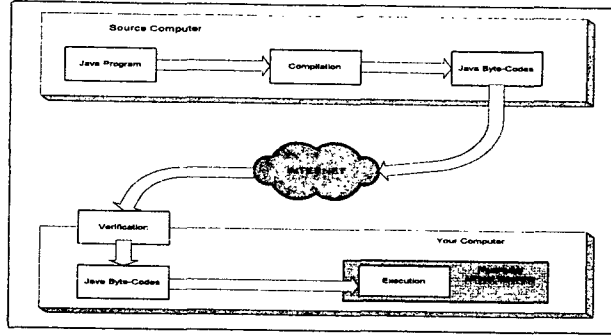


그림 2-2. 바이트 코드 실행까지의 경로

다) 안전성

Java는 초보적인 프로그래머들에게 있어서의 가장 많은 오류를 발생케 하는 포인터 연산을 배제하고 있으며 자체적인 가비지 콜렉션(Garbage Collection)기능을 가지고 있는 메모리구조를 가지고 있어 안정적이고 컴퓨터내의 가상기계(Virtual Machine) 내에서 수행되므로 폭주 등의 문제에 대해서 안전하다. 또한 프로그램에 버그가 있을 경우라 하더라도 시스템이 다운되는 대신 OOP의 예외처리기능(Exception Handling) 등을 이용해 처리하게 된다.

라) 병렬성

현재의 UNIX, Windows NT, Windows 95와 같은 운영체제들은 기본적으로 다중 스레드(Multithread)를 제공한다. Java는 이러한 병렬적인 처리를 위한 스레드 동기화 등을 프로그래밍 언어 차원에서 지원할 수 있으므로 이러한 시스템의 자원들을 효율적으로 사용할 수 있다. 또한 Java의 스레

드는 운영체제의 스레드와 직접 매핑될 수 있으므로 멀티프로세서를 지원해주는 운영체제에서는 매우 뛰어난 성능을 발휘할 수 있다.

마) 손쉬운 프로그래밍 환경

Java는 앞서 살펴본 것과 같이 가비지 컬렉션과 자동적인 메모리 할당 등을 통해 메모리 관리가 매우 손쉬우며, 스레드 동기화 등의 기능을 프로그래밍 언어차원에서 가지고 있고 다양한 클래스들을 가지고 있으며, AWT(Abstract Window Toolkit) 등을 이용할 수 있어 프로그램 개발이 매우 용이하다. 더욱이 최근에는 SUN microsystems의 Java Workshop, Symantec사의 Java Cafe, Microsoft사의 Visual J++, Borland 사의 J-Builder 등의 다양한 비주얼 툴 들의 개발에 힘입어 타 언어와 겨루어 전혀 손색이 없는 개발환경을 갖추게 되었다

나. 본 연구에서 사용한 Java 기술

본 연구에서는 최근의 Java 기술을 이용하여 소프트웨어를 구현하였다. 본 연구에 사용된 IFC, JFC, Java Comm, JDBC 등 Java 기술은 구현 도구들에 있어서 중심적인 기술들을 제공하였다. 이들 기술의 주요 내용은 그림 표 3-1과 같다

IFC (Internet Foundation Class)
경량 컴포넌트를 제공하는 것으로 화면 디자인 및 클라이언트의 기본 그래픽 위젯으로 이용하였다. 단순한 그래픽 위젯뿐 아니라 이벤트 전달 방법을 개선하여 많은 성능 향상을 느낄 수 있다.
JFC (Java Foundation Class)
Java의 경량 컴포넌트를 제공하는 것으로 IFC를 기반으로 하여 look and feel을 지원하는 100% 호환 확장 AWT 집합이다. 이 중 가장 많이 사용되는 부분은 AWT 그래픽 컴포넌트인 Swing은 매우 좋은 GUI를 제공해준다.
Java Comm (Java Communication)
RTU와 통신을 위한 I/O 처리를 지원하는 표준화된 직렬 통신 포트 접근 환경이다. 실제 안정적으로 동작하며, 플랫폼에 독립적으로 직렬 통신 포트를 이용할 수 있다는 장점을 가진다.
JDBC (Java DataBase Connectivity)
Java와 데이터베이스를 연결하여 SQL 문을 처리해주는 드라이버이다. 이를 이용하여 실시간으로 발생하는 데이터를 데이터베이스에 저장하고 시스템의 변화 추이를 분석하는데 이용된다. 모니터링 된 시스템 정보는 매시간마다 데이터베이스에 갱신되고, 이를 일정한 시간동안 다시 검색하고 싶을 때 이를 용하여 데이터베이스에서 정보를 얻어오게 된다.

표 2-4. 본 연구과제에 응용된 소프트웨어 기반 기술

1) IFC (Internet Foundation Class, NETSCAPE 인터넷 기본 클래스)

IFC는 웹 브라우저로 널리 알려진 Netscape 사에서 만든 Java GUI 환경을 지원하는 개방화된 클래스 집합이다. IFC는 Netscape ONE 환경에 대해 개방형 개발 아키텍처와 함께 Netscape ONE 응용 프로그램은 복잡한 통신 중심 응용 프로그램의 구축을 쉽게 하는 서비스 영역을 액세스할 수 있다. 또한, 실행 중인 Java, JavaScript, 원래 구성 요소로 플랫폼 독립적 개체의 공통 집합, API, 네트워크 중심 응용 프로그램을 구축하고 전개하는데 필요한 서비스 등을 제공한다. 그림 2-3은 IFC의 구성 요소를 자세히 보여주고 있다.

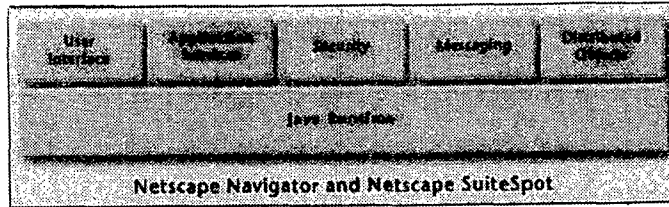


그림 2-3. Netscape 인터넷 기본 클래스

IFC의 첫번째 구성 요소는 개발자에게 플랫폼과 독립적 사용자 인터페이스 (UI) 컨트롤 및 서비스를 제공한다. Netscape는 더 융통성 있고 강력한 네트워크 응용 프로그램의 구조를 지원하기 위해 계속 확장될 것이다. 앞으로의 버전은 플랫폼에 독립적인 접근 방법을 제공할 예정이라고 한다. 이러한 접근 방법은 보안성과 암호화, 메시지, 분산된 객체 서비스 등을 포함한다. 예를 들어 클라이언트 워크플로우 응용 프로그램 구현 보안성 및 메시지는 Netscape 엔터프라이즈 서버 (Netscape Enterprise Server)의 HTTP 서비스 API를 통해 서버로부터 문서를 검색하고, 보안 서비스를 통해 문서의 근원을 검증하고, 문서를 다음 처리 단계로의 전송 등을 가능하게 해준다. 인터넷 기본 클래스의 기능은 사용자 인터페이스, 메시지 서비스, 보안 서비스, 분산 객체 서비스 등을 제공해 주는데 이중 본 연구 개발에서는 IFC의 사용자 인터페이스를 기본 인터페이스로 하였으며, 메시지 서비스의 일부를 응용하였다.

2) JFC (Java Foundation Class)

Sun Microsystems™는 JDK 1.1.2 이상의 버전에서 통합된 GUI를 제

공하기 위하여 Netscape, IBM, Lighthouse Design 등의 기술을 바탕으로 AWT보다 더 특엔필이 강한 GUI를 개발하였다. 이러한 노력의 결과로 JFC(Java Foundation Classes)로 불리는 API를 개발하였으며 이것을 이용하면 바로 상업적으로 사용이 가능한 개발 툴을 개발자에게 제공하였다.

JFC는 5가지 API로 구별된다. AWT, Java 2D, Accessibility, Drag and Drop, 그리고 Swing이다. AWT 컴포넌트는 JDK 1.1.2 또는 이후 버전의 AWT를 말한다. Java 2D는 IBM/Taligent의 기술을 라이선스한 그래픽 API이다. Accessibility API는 JFC의 화면확대기술 등을 제공하는 보조적인 기능들의 집합이다. Drag and Drop은 JavaBeansTM의 다음 세대를 이끌어갈 기술 중의 하나이다. "Glasgow"로 불리는 차세대 JavaBeansTM는 현재 독립적으로 개발 중에 있다. Swing은 형식적인 툴을 기반으로 하는 애플리케이션을 위한 API이다. Swing은 미려한 위젯과 플랫폼 독립적인 GUI등을 제공한다. 본 과제에서는 이중에서 Swing 컴포넌트를 이용하였다.

비록 Swing이 IFC에 기반하고 있지만 개발자 입장에서 보면 서로 거의 비슷한 점이 없다. Swing 위젯과 가시적인 화면들의 느낌은 많은 기본적으로 IFC로 부터 파생된 것이다. AWT 1.1. 위젯과 이벤트 모델은 여전히 JFC-Swing에서도 역할을 한다. (단, AWT 1.0 모델과는 호환성이 없다.) Swing은 Components, JComponents, 그리고 관련된 지원 클래스로 구성된다. AWT와 같이 Swing 컴포넌트는 모두 JavaBeans의 이벤트 모델을 수용하고 있으며, 그 자체가 JavaBeans이다.

Swing 위젯의 부분 집합은 기본 AWT위젯과 유사하다. 몇가지 경우, Swing 버전들은 운영체제에서 지원하는 기본 컴포넌트가 아닌 단순한 경량 컴포넌트 들이다. 경량 컴포넌트는 AWT 1.1 에서 소개된 컴포넌트 구조

이다. 경량 컴포넌트는 Java가 동작하는 기본 운영체제에서 지원하지 않는 컴포넌트들을 사용할 수 있도록 컴포넌트 구조를 만든 것이다. 대신에, MVC (Model-View-Controller)에 결합되어 있으며 이것은 2부에 설명할 것이다. Swing은 몇 가지 새로운 트리, 탭 기반의 팬, 분리된 팬 컴포넌트들을 포함하고 있으며 GUI들의 모양과 기능들을 많이 향상시켰다.

AWT에서 `java.awt.Container`의 하위 클래스들은 `java.awt.Components`를 포함할 수 있다. 따라서 Containers는 자기 자신이 Components가 되기 때문에, 얼마든지 깊은 단계의 컴포넌트를 가질 수 있다. 예를 들면, 두 개의 패널을 가진 하나의 프레임 내에서 각 패널들은 두개의 버튼을 가질 수 있다. Swing은 여기서 더 발전한 컨테이너 개념을 수용하였다. 모든 JComponents는 `java.awt.Container`의 하위 클래스이다. 이것은 Swing 위젯들 (JLabel과 같은)이 다른 AWT나 Swing 컴포넌트들을 포함할 수 있다는 것을 의미한다. 이러한 기법들을 통하여 구현이 어려운 GUI를 쉽게 구현할 수 있게 해준다.

3) Java Communication API

Java 통신 API는 보이스 메일, 팩스, 스마트 카드 등을 위한 애플리케이션을 플랫폼 독립적으로 개발할 수 있게 해주는 역할을 한다. 현재, RS232 직렬 포트와 IEEE 1284 병렬 포트를 접근할 수 있는 방법들을 제공해준다. 현재 (1998년 8월) 제공되는 기능은 다음과 같다.

- 시스템의 모든 통신 포트의 접근
- 통신 포트의 개방과 사용자 권한의 설정
- 다중 애플리케이션 사이의 사용자 권한 경쟁 해결

- 통신 포트의 입출력의 동기적/비동기적 제어
- 통신 포트 상태 변화에 따른 이벤트를 JavaBeans 스타일로 제공

4) JDBC (Java Database Connectivity)

본 연구에서 가장 중요한 이슈가 바로 데이터베이스와의 접속이다. JDBC는 Java를 통한 데이터베이스 접근의 표준을 제공하는데 그 목적으로 하고 있다. 97년 5월 현재 JDBC 1.1 버전이 공개된 상태이다. 순수한 Java 클래스 형태로 제공되는 JDBC API는 ODBC나 기존의 데이터베이스를 액세스하는 인터페이스보다 상위 레벨을 구현할 목적으로 고안되었다. 또한 브라우저에서 보안에 위배되지 않는 조건에서 다양한 데이터베이스 접근 방법을 제공함으로써 기존의 수동적인 데이터베이스 접근을 벗어나 보다 더 능동적인 데이터베이스의 이용이 가능하게 되었다.

JDBC의 특징을 살펴보면 다음과 같다. JDBC는 통합적인 SQL 데이터베이스를 이용할 수 있는 프레임워크로서 DBMS에 비종속적인 Java 애플리케이션을 구현하도록 고안되었다. JDBC는 SQL CLI(Call Level Interface)를 지원하기 위한 X/Open 정의에 기반을 두고 있다. 이는 기존의 ODBC 메카니즘을 연상하면 쉽게 이해할 수 있으며, ODBC보다 상위 레벨의 API를 지원한다. JDBC는 SQL을 사용해 데이터를 액세스할 수 있는 함수 라이브러리를 제공한다. 그리고 API와의 인터페이스를 위한 드라이버는 ANSI SQL-2 엔트리 레벨 이상을 지원할 수 있어야 한다. JDBC는 SQL 스트링을 DBMS에게 전달할 수 있는 수단을 제공하고 더불어, 실행될 SQL 스트링에 대한 올바른 표현에 대해 컴파일 단계에서 기본적인 검사가 이뤄진다. 또한 각각의 데이터베이스에 종속적인 특정 함수에 대해서도 개

발자가 이용할 수 있도록 다양한 통로를 제공한다.

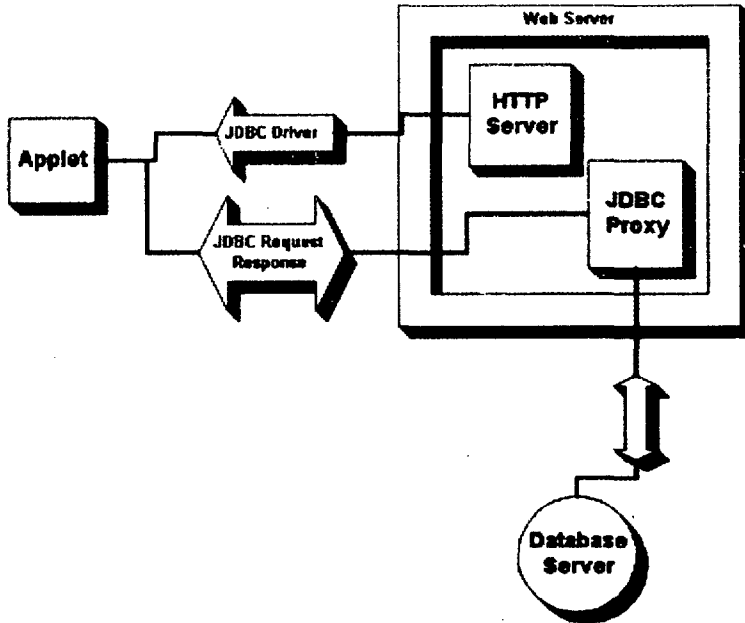


그림 2-4. JDBC를 이용한 접근 방법

JDBC를 통하여 DBMS에 접근하는 가장 일반적인 방법은 그림 2-4와 같다.

JDBC(Java DataBase Connectivity)는 ODBC(Open Database Connectivity)와 유사한 구조를 갖고 있으면서 ODBC를 포함하는 상위수준으로 설계되었다. JDBC API와 이를 구동하는 드라이버로 구성되어 있고 어떠한 드라이버를 이용하느냐에 의해 시스템의 구성이 달라진다. JavaSoft에서 JDBC API를 발표하고 이를 지원하는 드라이버는 써드 파티(third party)에 분담함으로써 많은 벤더의 참여와 공동 개발의 효과를 기대하고 있다.

가) JDBC 드라이버 타입

타입	JDBC/ODBC Bridge
설명	JDBC function call을 ODBC function call로 바꾼다. 개발자가 작성한 application이나 applet을 현재 ODBC와 연결된 많은 솔루션으로 전환하고자 할 경우 용이하다.
장점	JDBC를 이용하는 미들웨어 제품 지원이 미흡한데 기존의 시스템을 유지하면서 바로 개발가능하다. 다시 말해 기존의 이미 개발된 시스템의 확장에 용이하다.
제한 사항	JDBC/ODBC Bridge는 C++로 작성된 DLL이나 shared library에 의해 구현되므로 런타임시 네트워크를 통해 다운로드 되어질 수 없다. 모든 클라이언트는 JDBC/ODBC Bridge와 ODBC 드라이버를 가지고 있어야 하고 실행하고자 하는 application과 applet도 클라이언트에 있어야 하므로 웹 브라우저를 통해 구현할 수 없다. 주로 기존의 환경을 유지하며 Java 애플리케이션으로 개발하고자 할 경우 유리하다.

타입	Native-API, Partly-Java Drivers
설명	JDBC call을 특정 데이터베이스에서 사용하는 클라이언트 API로 바꾼다. 이 타입의 드라이버는 SQL*Net, I-Connect, OpenClient등의 프로토콜을 이용하고, DLL이나 shared library를 통하여 이러한 프로토콜과 인터페이스 한다. 애플릿의 수행에 필요한 파일들이 다운로드될 수 없고, 아직까지 애플릿이 갖는 보안 제약 때문에 브라우저에서 수행시 보안위배(security violation)를 일으킨다.
장점	기존의 DBMS 벤더에서 제공하는 미들웨어를 그대로 이용할 수 있다.
제한 사항	네트워크 클라이언트 모듈인 SQL*Net, I-Connect, OpenClient 역시 C나 C++로 작성된 DLL 혹은 shared library로 구성되어 있다. 네트워크를 통해 다운로드 되어질수 없으므로 여전히 클라이언트는 필요한 모든 구성요소를 갖추고 있어야 한다. 이 드라이버 역시 브라우저에서 이용할 수 없다. (참고: Native-API, Partly-Java 드라이버는 '로컬 시스템의 자원을 접근할 수 없다'는 애플릿의 규약에 제약을 받는다. 그렇지만 이를 극복하기 위한 일시 방편으로 브라우저에서 제공하는 Plug & Play기능을 이용하여 웹에서 이용할 수 있도록 만든 드라이버도 있다.)

타입	Net-Protocol All-Java Drivers
설명	이 드라이버는 대개 서버 어플리케이션이 존재한다. 다운로드된 애플릿과 서버 어플리케이션은 DBMS와 무관한 프로토콜로 인터페이스하고 다시 서버 어플리케이션에서 DBMS를 액세스하기 위해 DBMS 전용 프로토콜로 변환한다. 이 솔루션에서 제공되는 드라이버는 주로 통합 구성된 한 개의 드라이버를 이용하는데 이 드라이버를 이용하면 여러 개의 DBMS를 동시에 사용할 수 있다. 다운로드된 애플릿은 서버에 있는 어플리케이션과 인터페이스하며 이 서버는 다시 클라이언트 애플릿과 DBMS사이의 중계역할을 수행하게 된다. 이것은 현재 애플릿이 갖는 보안 제약사항을 최소화시키며 확장성이 가장 우수하다
장점	이 드라이버는 아주 유연한 솔루션을 제공한다. 모두 순수 Java로 구성된 드라이버이므로 네트워크를 통해 다운로드될 수 있고, 서버에 있는 어플리케이션은 원거리 지역(서버 머신 외부)에 있는 DBMS와의 연결을 가능하게 해준다. 애플릿이 갖는 보안 제약사항을 그대로 적용한 구성이므로 안전한 애플릿(trusted applet)과 불안정한 애플릿(untrusted applet) 모두를 지원한다.
제한 사항	보안(security)사항에 많은 유연성을 제공하는 구성이므로 동시에 보안의 허점을 노출시킬 수 있는 위험요소를 내포하고 있다. 방화벽을 이용할 경우 인터넷상에서 정상적인 작동을 하지 않을 수 있고, 만약 불안정한 애플릿(untrusted applet)일 경우 서버 어플리케이션과 DBMS는 같은 머신에 있어야 한다. 이러한 모든 제약사항이 이 제품을 제공하는 벤더에서 어떻게 설계했느냐에 의해 좌우되는데 여러 기능을 통제하는 수단이 많은 만큼 관리(administration)가 다른 드라이버에 비해 비교적 힘들다.

타입	Native-Protocol All-Java Drivers
설명	이 드라이버는 DBMS와 직접 인터페이스 하므로 클라이언트에서 요구한 JDBC call을 바로 DBMS가 사용하는 프로토콜로 전환한다. 주로 DBMS 클라이언트용 네트워크 모듈(SQL*Net, I-Connect 또는 OpenClient)을 순수 Java 프로그램으로 대체한 형태이다. 3-tier 구조로 이루어졌기 때문에 여러개의 DBMS를 이용할 경우 각각의 해당 드라이버를 동시에 적재(load)하면 된다.

장점	기존의 DBMS에서 사용하던 방식을 그대로 순수 Java로 재구현하여 사용할 수 있다. 또한 모든 미들웨어 성격의 모듈들이 Java로 구현되었기 때문에 네트워크를 통해 이동되어 질 수 있다. 다시 말해 기존의 ODBC 클라이언트용 네트워크 모듈없이 서버에서 다운로드 받아 실행 할 수 있다는 것이다.
제한 사항	이 드라이버의 최선두 공급자는 DBMS 벤더이다. 그러므로 써드 파트의 개발은 DBMS 벤더에 종속되어 진다. 쉽게 말해, DBMS 벤더에서 서버쪽의 네트워크 모듈을 고칠 경우 써드 파트 역시 해당 부분의 모듈을 고쳐야 하는 것이다. 기존의 CS(client-server)환경에서 이용되었던 네트워크 모듈을 이용하는 경우이므로 인터넷의 이용을 고려하지 않은 상태이다. 그러므로 프락시(proxy)나 방화벽을 통과할 경우 많은 문제가 발생하고, 시스템 구성시 옵션에 많은 제약이 뒤따른다.

표 2-2. JDBC Driver Types

위의 구분은 Sunsoft와 협력업체인 Intersolv의 정의를 따른 것이다. 그렇지만 JDBC가 버전 업되고 기타 드라이버 제공업체들이 개발을 추진해오면서 계속 위의 구분에 변형을 추가하고 있으므로 보다 포괄적인 데이터베이스 접근 방식으로 구분해 보면 다음과 같다.

(1) 3-tier

Native-Protocol All-Java 타입의 드라이버가 갖는 시스템 구성과 유사하다. 클라이언트로 다운로드된 애플릿이 DBMS에 직접 접속하는 방식으로 기존의 비주얼 툴에서 이용되었던 방식과 유사하다. 다만 실행 프로그램이 서버에 위치하여 실행시 클라이언트로 다운로드 되어 실행된다는 점이 다르

다. 하나의 애플릿으로 여러 DBMS를 이용하려고 할 경우 각각의 DBMS에 해당하는 드라이버를 모두 적재하는 방식으로 구현한다.

(2) 3-tier

Net-Protocol All-Java 타입의 드라이버가 갖는 시스템 구성과 유사하다. 보통 드라이버는 다수의 드라이버를 하나로 묶어 놓은 통합된 형태로 존재하고 서버에는 Java 서버 애플리케이션이 실행된다. 보통 이 서버를 경유하여 DBMS를 액세스하게 되는데 바로 이 서버의 존재 유무가 multi-tier의 구성을 가능하게 한다. 보통 미들웨어에서 거론되는 tier개념과 다른 것이, 미들웨어에서는 서버의 역할에 의해 tier가 분리되는 데 JDBC의 경우 몇 단계의 서버를 거치는가에 의해 결정된다. 클라이언트 애플릿과 서버는 여러 DB를 액세스하기 위한 통합형 드라이버를 지원할 수 있는 프로토콜을 이용하고 서버에서 DBMS를 액세스할 때 비로소 DBMS용 전용 프로토콜로 전환된다. 클라이언트와 서버 그리고 서버와 DBMS는 각각 다른 프로토콜을 이용한다. 위의 3-tier구조에 비해 유연한 시스템 구성이 가능한데 DBMS와 웹 서버가 같은 머신에 위치하지 않아도 된다.

2. WWW 기술을 이용한 Servlet 모듈 개발

다양한 웹 기술들 중에서 Servlet은 웹의 단점인 정적인 동작환경을 동적으로 바꾸어주는 역할을 수행한다. 따라서 현재 발생하는 어떠한 상황을 동시에 웹 서비스에 반영할 수 있는 현재의 유일한 표준 기술이라고 할 수 있는 것이다. 개발환경에 따라 웹 브라우저에 플러그인을 설치하는 방법도 있으나 이는 모든 브라우저에 해당 프로그램을 설치하여야한다는 단점이 있

다. 이와 같은 경우 서버의 부하를 줄여 보다 많은 사용자의 요구를 받아들일 수 있다는 점을 장점으로 꼽을 수 있다. 본 시스템의 각 모듈들간의 구성은 그림 2-5와 같다.

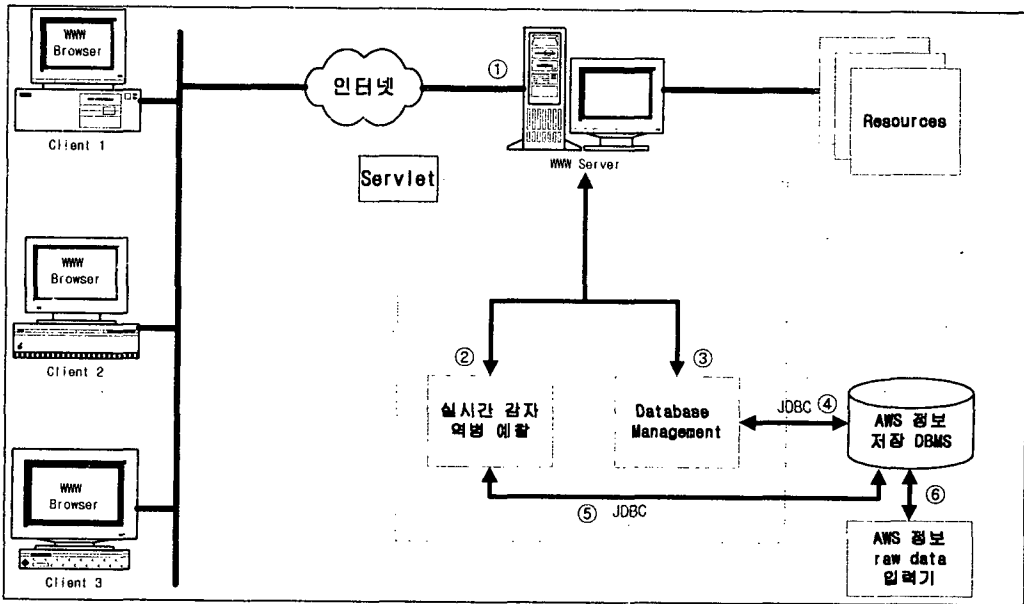


그림 2-5. 전체 시스템에서 각 모듈들간의 구성도

① 클라이언트 측의 웹 브라우저의 URL(Uniform Resource Location) 요구에 대하여 서버 측의 웹 서버는 리소스의 제공을 담당하게 된다. 응답하게 되는 리소스로는 비트 이미지, HTML (HyperText Markup Language), Java 클래스 파일 등이 있다. 본 과제에서는 선마이크로시스템사의 Java 웹 서버를 이용하게 된다.

② 본 시스템에서 웹 서버는 클라이언트 측의 요구에 대한 각각의 서비스 루틴을 호출하게 되는데 이 경우는 실시간 감자 역병 예찰에 대한 사용

자의 요청이 있는 경우에 해당한다.

③ 새로운 사용자가 농장 정보를 등록하거나 또는 기존의 사용자가 자신의 농장 정보를 변경 시 Java Servlet은 데이터베이스 관리 루틴에게 추가 또는 변경된 농장 정보를 보내게 된다. 그리고 데이터베이스 관리 루틴은 ④의 과정을 통하여 되돌려 받은 정보를 사용자측의 Java 애플릿에게 되돌려주게 된다.

④ 데이터베이스 관리 루틴은 사용자측의 요구를 SQL(Standard Query Language)로 바꾼 후에 JDBC(Java DataBase Connectivity)를 통하여 DBMS(DataBase Management System)에 접속하여 클라이언트에게 보내줄 데이터를 받게 된다.

⑤ 실시간 감자 역병 예찰 루틴은 먼저, 사용자의 요구로써 받은 발아일을 ④와 마찬가지로 JDBC를 이용하여 DBMS에 접속한다. 사용자의 농장 AWS 정보를 획득한 후 현재 강우량, 습도, 온도 등의 기상 자료를 이용하여 감자 역병 예찰 알고리즘을 수행하고 그 결과를 전달한다.

⑥ AWS 정보 raw data 입력기는 AWS 정보 저장 DMBS와 연결되어 DB(DataBase)에 저장되어 있는 강우량, 습도, 온도 등의 기상 정보를 추가, 수정, 삭제하게 된다. 이 때 입력기는 Java 어플리케이션으로 작성하고 있다.

본 시스템에서 웹 서버로는 선마이크로시스템사의 Java 웹 서버를 사용하였다. Java 웹 서버는 Java소프트의 “Java 서버제품군”중의 한 구성원으

로 'Jeeves'라는 이름으로 발표되기도 했다. Java 서버제품군은 인터넷의 server side client/server 어플리케이션과 인트라넷, 네트워크 컴퓨팅 환경에 Java를 이용할 수 있게 해주는 제품이다. Java 서버제품군은 Java 서버 툴킷(Java Server Toolkit)과 Java 웹 서버(Java Web Server), Java 서블릿 개발 키트(Java Servlet Development Kit), Java NC 서버(Java NC Server)로 구성되어 있으며 모두 Java 서버 API에 기반을 두고 있다. Java 서버 API는 크로스플랫폼, 네트워크 중심의 서버 솔루션을 제공하고 있다. 이 API로 웹 서버, 프록시 서버, 프린트 서버, 부트 서버 등을 개발할 수 있는데 Java소프트가 이 API를 이용해 첫 번째로 만든 것이 바로 Java 웹 서버이다. 현재는 Java 웹서버 1.1까지 발표되었고, JDK1.1 파이널 이후의 환경에서 수행된다. Java 웹 서버는 말 그대로 Java로 만든 웹 서버이다. Java로 만들었기 때문에 Java의 장점을 고스란히 웹 서버에 적용시킬 수 있게 된다. 또한 기존의 웹 서버가 하던 기능은 모두 가지고 있으며 서블릿을 추가함으로써 웹 서버의 능력을 극대화시키고 있다.

3. 실시간 감자 역병 예찰 시스템의 사용자 인터페이스

2 차년도에 구현한 사용자 인터페이스에 대한 자료는 그림 2-6과 같다. 사용자는 Internet이 사용 가능한 일반 개인용 컴퓨터로 시스템에 접속하게 되며 웹사이트에 접속하였을 때의 초기 화면으로 감자역병예찰, 감자역병진단 등의 메뉴를 선택 할 수 있다.

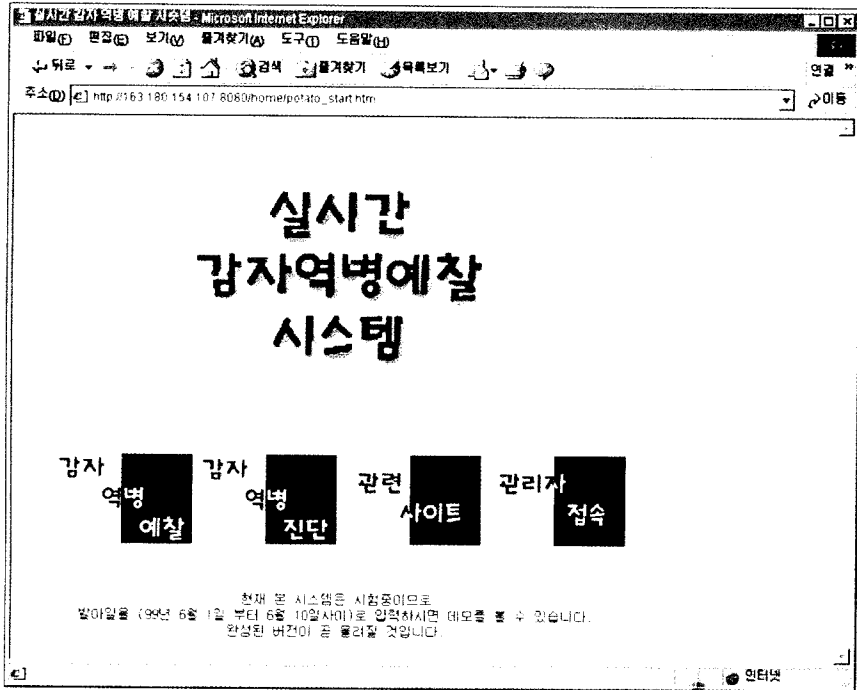
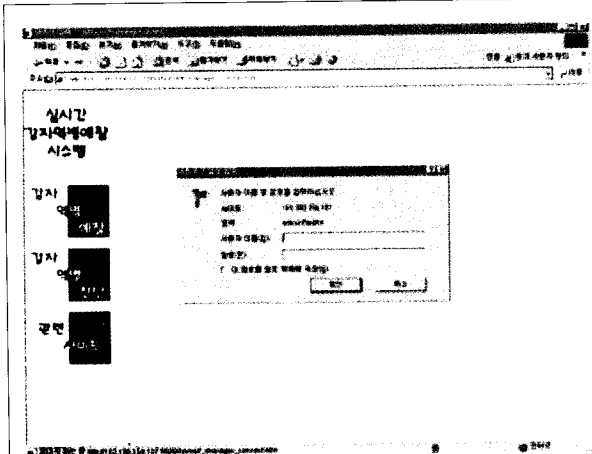


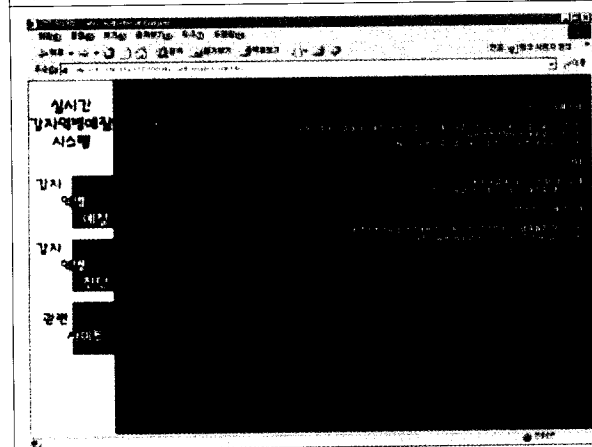
그림 2-6 실시간 감자 역병 예찰 시스템의 사용자 인터페이스

사용자 인터페이스의 각 주요 기능에 대한 설명은 다음 표 2-3과 같다.

사용자 인터페이스의 주요 기능	설 명
	<p>실제로 감자의 역병에 대한 관리를 하는 부분으로서 크게 세 가지 메뉴로 구성된다. AWS의 정보를 이용하기 위하여 농장의 정보를 등록하는 메뉴, 등록된 농장의 정보를 이용한 역병 예찰, 그리고 AWS가 설치되어 있지 않은 곳에 대한 역병 예찰 메뉴가 그것이다. 농장 등록을 하기 전에 관리자는 미리 AWS를 등록해 두어야 하며, 사용자는 등록된 AWS중 선택하여 자신의 농장에 대한 역병 예찰을 할 수 있다.</p>

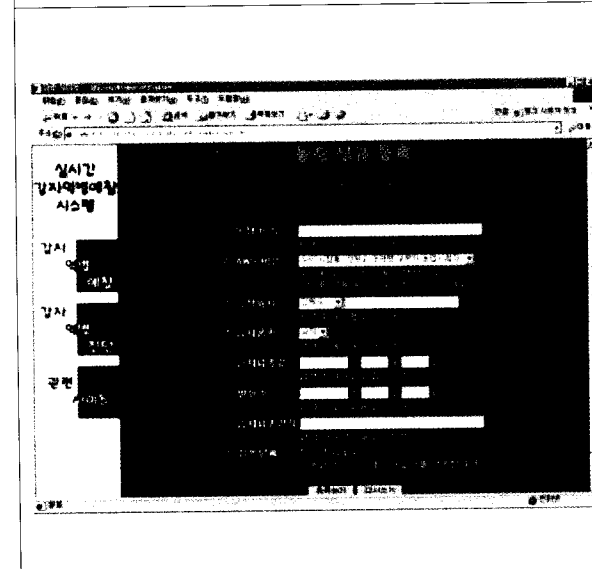


관리자 접속 메뉴로서, 새로운 AWS의 등록이나 AWS의 기상관측자료의 자동 수집의 수행과 관련된 일을 수행 할 수 있다.



농업에 관련된 홈페이지들을 모아 링크 시켜놓은 메뉴이다. 사용자가 농업에 대한 다른 정보들을 얻고자 할 때 유용하게 쓰일 만한 곳을 연결시켜준다.

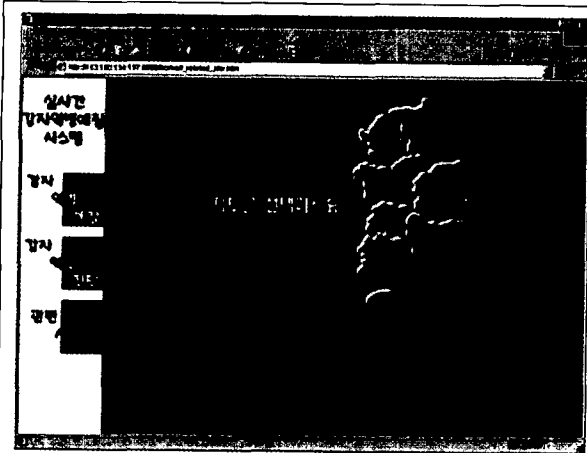
또한, 본 시스템에 관계된 기관 또는 관리자와 연결할 수 있는 전화번호, 주소, 전자우편 주소 등을 제공한다.



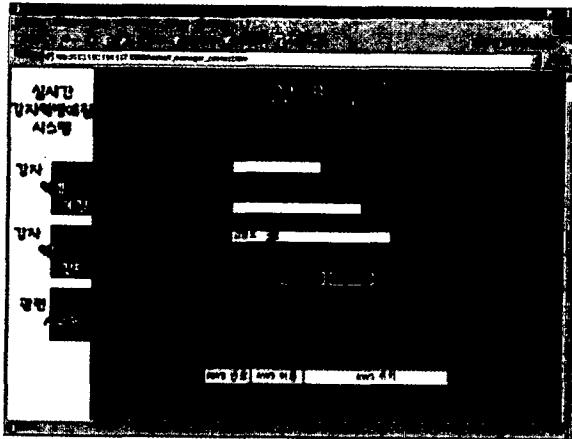
AWS의 실시간 기상정보를 이용한 예측을 위해서는 미리 농장에 대한 정보에 데이터베이스가 구축되어야 한다.

이 부분은 농장의 신규 등록을 위한 부분으로, 사용자와 관련된 기초 자료를 입력받고, 그 자료를 데이터베이스에 저장한다.

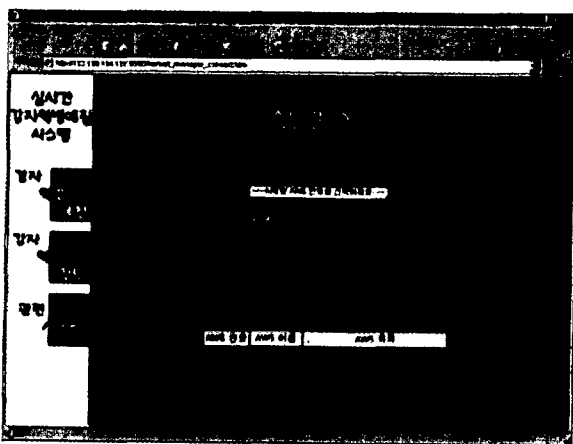
한 번 등록을 끝내면, 재 접속하였을 경우에는 자신이 등록한 농장을 선택만 하면 그 농장에 대한 역병관리를 할 수 있다.



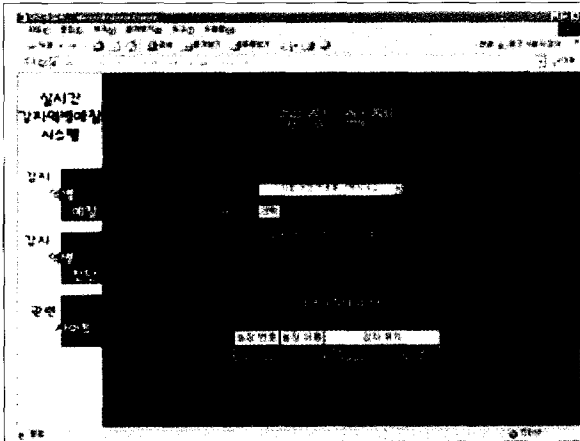
시스템에 등록되어 있는 농장을 선택하는 부분으로, 자신이 원하는 농장이 포함되어 있는 도를 지도상에서 클릭하면 그 도내에 포함되어 있는 등록된 농장의 리스트를 보여주게 되어 있다. 지도 위에 포인터가 위치하면 그 부분의 지명이 화면에 표시되어 선택이 용이하도록 구성되어 있다.



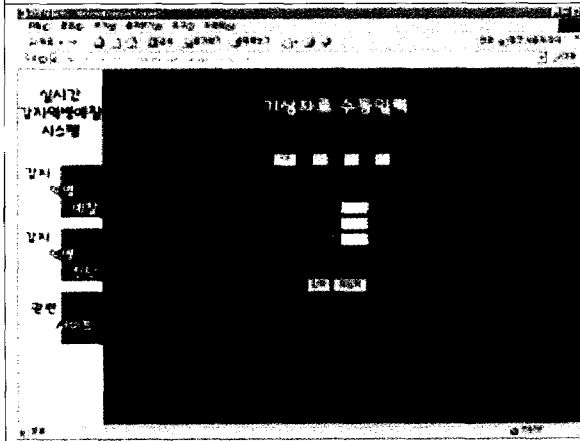
관리자 접속 메뉴로서, 새로운 AWS를 등록할 수 있는 페이지이다. 번호는 관리자가 직접 등록할 수 있으며, 이미 등록된 AWS의 리스트도 보여 준다.



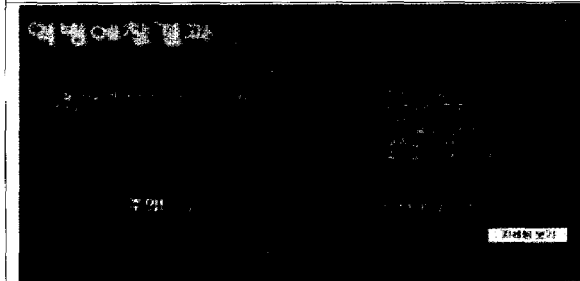
관리자 접속 메뉴로서, 기존의 AWS를 삭제할 수 있는 페이지이다. AWS를 식별할 수 있는 번호를 리스트에서 선택하고 선택된 AWS를 삭제한다.



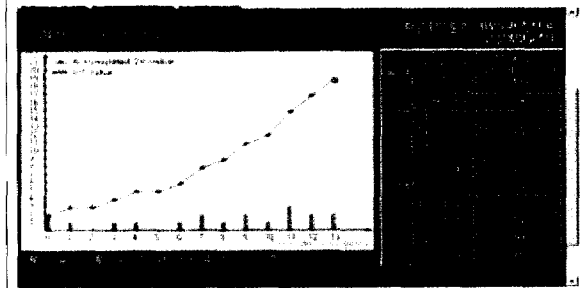
관리자 접속 메뉴로서, 등록된 농장들을 삭제할 수 있는 페이지이다. 각 농장을 식별할 수 있는 고유 이름과 농장 번호를 리스트에서 선택하고 선택된 농장을 삭제한다.



관리자 접속 메뉴로서, 기상 자료를 수동 입력할 수 있는 페이지이다. AWS가 설치되어 있지 않은 곳에서도 역병 예찰이 가능하도록, 몇 일간의 하루 평균값만을 입력해주면, 이동 평균법을 이용하여 입력한 값에 대해 역병 예찰 정보를 제공한다.



AWS에 의해 구축된 실시간 기상 정보에 대한 데이터베이스를 기반으로 BLITE-CAST법과 이동 평균법을 이용하여 역병 예찰을 한다. 각 도별로 등록되어 있는 농장을 보여주며, 더욱 정확한 역병 예찰을 위해 받아일을 다시 지정해 줄 수 있다.



역병 예찰을 시도하면, 역병 발병이 예상되는 경우, 큰 글씨로 그 사실을 알려주며, 더욱 자세한 정보를 보기 위한 버튼을 사용하면, BLITECAST법으로부터 얻어진 그래프와, 이동평균법의 예찰 결과, 최근 10일간의 기상 관측 자료 등을 관찰할 수 있다.

4. JDBC를 이용한 데이터베이스와의 연동

JDBC는 Java 프로그램 내에서 SQL문을 실행하기 위한 Java API이다. "Java DataBase Connectivity"의 약자로 간주되기도 하지만 사실상은 상표 이름이다. JDBC는 Java로 작성되어진 클래스와 인터페이스들로 구성되어 있다. 툴/데이터베이스 개발자들을 위한 표준 API를 제공하고 pure Java API를 사용하여 데이터베이스 어플리케이션을 만들게 해준다. JDBC를 사용하면, 어떠한 관계 데이터베이스(relational database)로도 SQL문을 전송하기 쉽다. 즉, JDBC API를 사용하면 Sybase, Oracle, Informix에 접근하는 프로그램을 따로 만들 필요가 없다. 단지 하나의 프로그램을 작성하고 그 프로그램에서 SQL 문을 적당한 데이터베이스에 전송할 수 있다. 또한 어플리케이션을 Java로 작성한다면, 어플리케이션을 플랫폼에 따라 다르게 작성하지 않아도 되기 때문에 Java와 JDBC의 결합은 하나의 프로그램이 어디에서나 동작할 수 있게 해준다.

Java는 사용하기에 견고하고 안전하고 쉽고 이해하기 쉬우며 네트워크 상에서 자동적으로 다운로드되기 때문에 데이터베이스 어플리케이션을 만드는 데 있어서 최적의 언어이다. 단지 필요한 것은 다양한 데이터베이스에 연결하는 방법일 것이다. JDBC는 이러한 것을 위한 메카니즘이다.

JDBC는 Java의 기능을 확장한다. 예를 들어 Java와 JDBC API를 사용하면, 리모트 데이터베이스에서 얻은 정보를 사용하는 애플릿을 웹페이지에 포함시킬 수 있다. 또는 기업에서 서로 다른 플랫폼을 사용하는 사원들이 인트라넷을 사용하여 하나 이상의 내부 데이터베이스에 연결하는 데에도 JDBC가 사용될 수 있다. Java를 사용하는 프로그래머들이 증가하면서, 쉬운 데이터베이스 접근에 대한 필요가 점차적으로 요구되고 있다. MIS 관

리자들은 Java와 JDBC 결합을 좋아한다. 왜냐하면 산재해 있는 정보를 쉽고 경제적으로 만들기 때문이다. 기업은 인스톨 되어있는 데이터베이스를 계속 사용할 수 있으며 정보가 다른 DBMS들에 저장되어있어도 쉽게 정보에 접근할 수 있다. 새로운 어플리케이션을 개발하는데 소요되는 시간도 짧다. 인스톨과 버전 업도 매우 간단하다. 단지 프로그래머가 어플리케이션을 업데이트하여 서버에 넣으면 모든 사람은 최신버전에 접근할 수 있게 된다.

가. 자바의 역할

JDBC는 간단히 다음 세 가지 기능을 가지고 있다.

1. 데이터베이스와 연결한다.
2. SQL문을 전송한다.
3. 결과를 처리한다.

아래 그림 3-9의 코드는 위에서 말한 세 가지 단계들을 위한 간단한 예제이다.

```
Connection con = DriverManager.getConnection("jdbc:odbc:wobat", "login", "password");
Statement stmt = con.createStatement();
ResultSet rs = stmt.executeQuery("SELECT a, b, c FROM Table1");
while ( rs.next() ) {
    int x = getInt("a");
    String s = getString("b");
    float f = getFloat("c");
}
```

그림 2-7. 데이터베이스와 JDBC를 사용하여 연결되는 Java 소스코드의 예

나. 본 과제에서의 접근 방법

본 과제에서는 이러한 JDBC를 데이터베이스와 연동되는 모듈들 내부에서 사용자에게 투명하게 제공되므로 사용자에게는 보다 SQL 문을 알지 못하더라도 데이터베이스의 데이터들을 쉽게 유지 및 관리할 수 있게 된다. 그림에서 JDBC가 사용되는 곳은 그림 2-5에서 ④,⑤,⑥에 해당한다.

또한 본 과제에서는 100% 순수한 Java로 작성되고 구성된 서버 시스템 구축을 목표로 하여 서버의 대부분의 컴포넌트들을 Java로 구성하였기 때문에 데이터베이스에 접근하기 위한 방법 역시 Java의 측면에서 접근하여 Java에서 제공하는 JDBC를 이용하게 된다. 과제에 사용될 데이터베이스 서버는 MS사의 MS-SQL 서버 7.0 으로서 이 서버는 ODBC를 지원하고 있기 때문에 JDBC-ODBC Bridge를 이용하여 결국 데이터베이스 서버에 접속되게 되는 것이다. 따라서 복잡한 ODBC는 염두에 두지 않고서도 단순히 JDBC-ODBC Bridge 드라이버를 사용하는 JDBC 만으로 데이터베이스에 실제 접근이 가능하게 되는 것이다.

5. 본 과제에서 개발한 AWS 정보 입력기의 Application Version

Application version의 개발 목적은 일반적인 인터넷상의 웹 기술을 이용하는 것이 아니라 TCP/IP 상에서 클라이언트/서버 구조를 가지며 애플릿 버전과 동일한 사용자 환경을 제공하는데 있다. 이는 현재의 PC 환경에서 뿐만 아니라 차후 널리 보급될 Java 칩이 사용된 Network Computer 또한 Java 가상 기계가 설치된 어떠한 기기에서도 기본적으로는 같은 사용자 환

정을 제공하게 된다.

Application version에서는 JDK를 이용하여 서버 측에서 데몬 형식으로 상주하는 서버 프로그램을 제작되고 있다. 이렇게 제작된 Application version에서는 Web Server를 사용하지 않고 단독 서버를 이용하여 클라이언트 측에 서비스를 하게 된다. 그러나 이 역시 데이터베이스에 접속하여 AWS 관리 정보, AWS raw data 정보 등의 데이터를 얻어와야 하므로 데이터베이스로의 접근이 필요하게 되며 이는 위에서 언급된 Java의 JDBC를 이용하게 되며, 구체적으로 본 과제에서는 MS사의 MS-SQL 서버 7.0을 사용하게 될 것이므로 JDBC-ODBC Bridge를 이용하여 최종적으로 데이터베이스와 접속이 이루어지게 된다.

이러한 버전의 장점은 일단 네트워크에 대한 부하를 대폭 감소시킬 수 있다는 것이다. 그 이유는 애플릿은 브라우저가 매번 서버에 접속할 때마다 운영에 관계된 모든 프로그램과 미디어 데이터들을 다운로드 받아 실행되기 때문에 그에 대한 서버의 부하뿐만 아니라 네트워크에도 심한 부하가 발생하게 된다. 그러나 애플리케이션 버전은 운영에 관계된 프로그램들이 미리 로컬에 설치되어 있기 때문에(최초설치) 차후 본 서비스를 제공받으려면 단지 서버에게 라우팅에 관한 정보만 얻어오면 되기 때문에 많은 부하를 발생시키는 미디어의 전송 따위는 생략되어 서버 측에도 많은 부하를 줄일 수 있게 된다. 실시간 감자 역병 예찰 시스템에서 AWS 정보 입력기 사용자 인터페이스에 대한 화면은 그림 2-8과 같다.

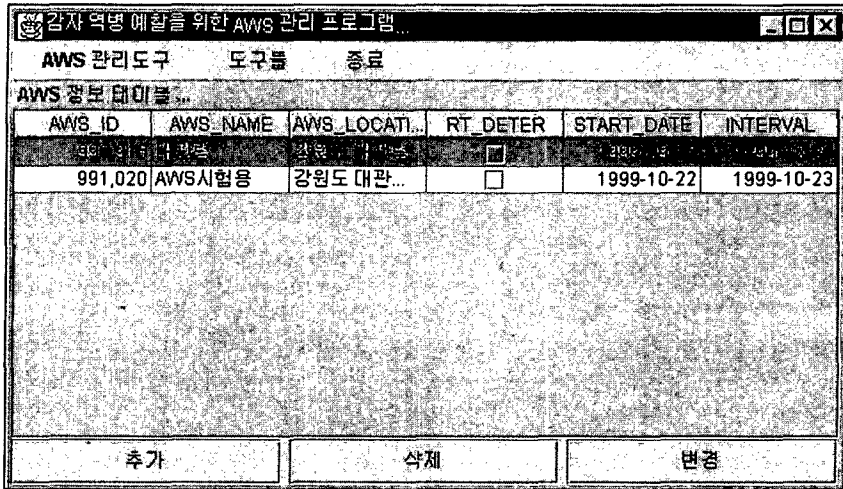


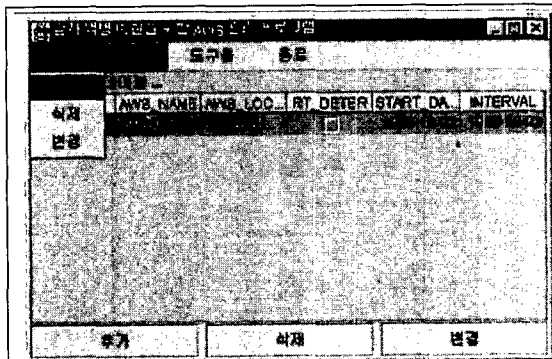
그림 2-8 AWS 정보 입력기 사용자 인터페이스

사용자 인터페이스가 개선된 AWS 정보 입력기를 이용하여 AWS 정보 검색을 한 경우를 그림 2-9에 나타내었다.

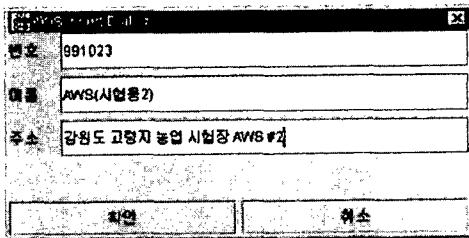
DETER_DATE	DETER_TIME	TEMPER	HUMIDITY	WIND
1999-05-26	1999-05-26	8.340	99.6	41.32
1999-05-26	1999-05-26	7.93	100	43.880
1999-05-26	1999-05-26	7.58	100	39.14
1999-05-26	1999-05-26	7.1	100	25.610
1999-05-26	1999-05-26	6.49	99.900	32.32
1999-05-26	1999-05-26	8.75	90.400	33.8
1999-05-26	1999-05-26	10.800	78	17.270
1999-05-26	1999-05-26	13.41	61.430	16.57
1999-05-26	1999-05-26	15.840	48.3	16.950
1999-05-26	1999-05-26	17.49	38.62	22.770
1999-05-26	1999-05-26	18.700	36.200	19.47
1999-05-26	1999-05-26	20.010	31.8	22.020
1999-05-26	1999-05-26	20.360	32.32	18.370

그림 2-9 AWS 정보 검색 결과

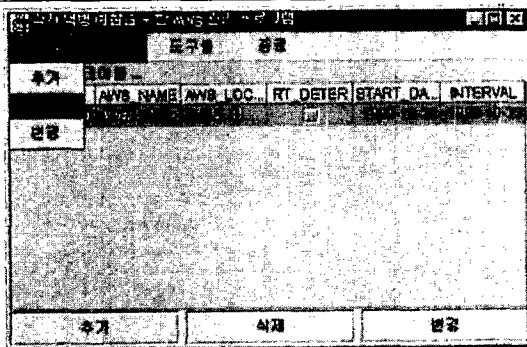
AWS 정보 입력기 사용자 인터페이스의 각 주요 부분에 대한 설명은 다음 표 2-4와 같다.



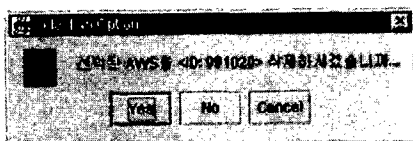
(AWS 추가 선택)



(AWS 추가 정보 입력)



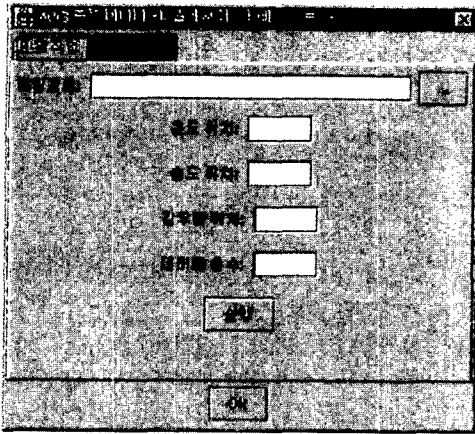
(삭제할 AWS 선택)



(선택된 AWS 삭제 여부 확인)

AWS 관리를 위한 도구로서 새로운 AWS를 추가할 경우 AWS 관리 도구 메뉴 또는 추가 버튼을 선택한다. 그후 AWS 추가 정보 입력을 위한 다이얼로그 박스를 통해서 AWS 정보를 입력한 후 확인을 선택한다.

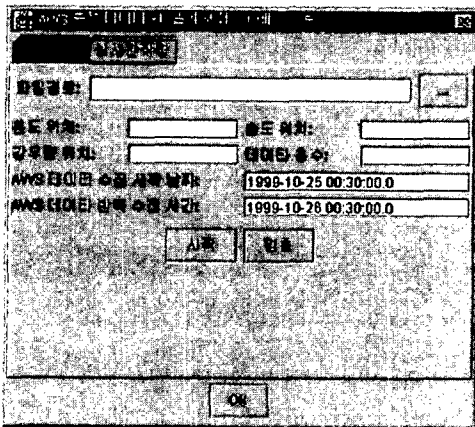
AWS 관리를 위한 도구로서 기존에 존재하는 AWS를 삭제할 경우 AWS 관리 도구 메뉴에 삭제 메뉴를 선택하거나 또는 삭제 버튼을 선택한다. 그후 현재 선택된 AWS를 삭제 여부를 확인 하는 다이얼로그가 실행이 되고 확인을 누르면 선택된 AWS 정보가 삭제가 된다.



(AWS 정보 입력 : 단일 작업)

AWS 정보 입력 작업을 위한 도구로서 AWS에 고장이 발생했을 경우 수작업으로 데이터를 추가 할 수 있어야 한다.

이를 위해 본 연구에서 개발한 AWS 정보 입력기는 단일작업 모드를 지원한다. AWS raw data 파일의 경로를 설정하고 감자 역병 예찰에 필요한 온도, 습도, 강수량 데이터의 위치와 총 데이터의 수를 입력하고 실행 버튼을 선택하면 설정된 경로에 있는 파일로부터 데이터베이스로 측정된 데이터들이 순차적으로 입력된다.



(AWS 정보 입력 : 실시간 작업)

AWS 정보 입력 작업을 위한 도구로서 보다 정확한 감자 역병 예찰을 위해서는 AWS와 실시간으로 연결하여 예찰 알고리즘에 필요한 데이터를 손상없이 데이터베이스화되어야 한다.

이를 위해 AWS 정보 입력기는 실시간 작업 모드를 지원한다. AWS에서 실시간으로 받아온 데이터를 일정한 위치에 저장하도록 파일 경로를 설정한 후 온도, 습도, 강수량 데이터의 위치와 총 데이터 수, AWS 데이터 수집 시작 시간과 반복 시간을 설정하고 시작 버튼을 선택하면 수집시간에 실행하여 설정된 반복시간 마다 측정된 AWS raw data를 데이터베이스로 저장한다..

표 2-4AWS 정보 입력기 사용자 인터페이스

제3절 연구 개발 목표 달성도 및 대외 기여도

1. 연구 개발 목표 달성도

연구개발목표	세부 개발 목표	달성 정도	달성률 (%)
<ul style="list-style-type: none"> ● 웹 서버 기능 및 사용자 인터페이스 기능 확장 	<ul style="list-style-type: none"> ▶ Java Web Server의 Servlet을 이용한 서버 개발 ▶ Java Web Server의 Servlet을 이용한 서버와 JDBC 등의 연동 기술 개발 ▶ WWW 및 인터넷을 이용한 멀티미디어 사용자 인터페이스 성능 개선 	<ul style="list-style-type: none"> ▶ Java Web Server의 Servlet을 이용한 서버의 개발 완료 ▶ Server과 JDBC의 연동 기술 구현 ▶ 성능이 개선된 사용자 인터페이스 구현 	100
<ul style="list-style-type: none"> ● 강원도 고령지 농업 시험장내 위치한 AWS raw data구축 및 raw data 자동 입력기 개발 	<ul style="list-style-type: none"> ▶ 강원도 고령지 농업 시험장에 위치한 AWS raw data 구축 ▶ AWS raw data 자동 입력 시스템 개발 ▶ 구축된 기상정보 raw data를 관리하기 위하여 Oracle Server 환경에서 작동하는 DB시스템 개발 	<ul style="list-style-type: none"> ▶ 강원도 고령지 농업 시험장에 위치한 AWS raw data 구축 완료 ▶ AWS raw data 자동 입력 시스템 개발 완료 ▶ 구축된 기상 정보 raw data를 관리하기 위하여 MS-SQL 서버에서 작동하는 DB 시스템 개발 완료 	100
<ul style="list-style-type: none"> ● 실시간 감자 역병 예찰 모듈 개발 	<ul style="list-style-type: none"> ▶ 감자 역병 예찰 알고리즘 조사 ▶ 감자 역병 예찰 알고리즘 설계 및 구현 	<ul style="list-style-type: none"> ▶ 감자 역병 예찰 알고리즘 분석(BRITECAST, 이동평균법) ▶ 감자 역병 예찰 알고리즘 (BRITECAST, 이동평균법) 구현 완료 	100
<ul style="list-style-type: none"> ● 시스템 통합 및 필드 시험 	<ul style="list-style-type: none"> ▶ 감자 역병 예찰 시스템, 감자 역병 전문가 시스템, raw data 관리 DB시스템 등의 통합 ▶ 통합시스템 필드 시험 	<ul style="list-style-type: none"> ▶ 감자 역병 예찰 시스템, 감자 역병 전문가 시스템, raw data 관리 DB 시스템 등의 통합 완료 ▶ 통합시스템 필드 시험 완료 	100

2. 평가의 착안점에 따른 달성도

평가의 착안점	평가의견
● 사용자 인터페이스는 직관적으로 이해가 쉽고 사용하기가 편리한가?	● 산업디자인학과 전문가의 자문으로 직관적이고 이해하기 쉬운 사용자 인터페이스를 구현하였다.
● WWW 접속은 잘 되는가?	● 네트워크를 이용할 수 있는 환경이라면 WWW 환경을 수용할 수 있다.
● 시스템이 원하는 정보를 쉽게 검색할 수 있는가?	● 감자 역병 예찰 알고리즘 및 Java Web Server의 사용으로 원하는 정보를 안정성있게 제공할 수 있다.
● 실시간 감자 역병 예찰 시스템이 실제 많은 AWS 정보가 입력되어도 잘 작동하는가?	● Java Web Server의 Servlet을 사용하였기 때문에 시스템의 스레드로서 안정성있게 동작한다.
● 감자 역병 예찰 결과에 대한 신뢰성이 보장되는가?	● BRITECAST법, 이동평균법을 사용하였기 때문에 신뢰성이 보장된다.
● 감자 역병 예찰을 수행하는데 사용자가 알기 쉽고 현실감이 있는가?	● 실시간 AWS 데이터를 사용으로 사용자가 알기 쉽고 현실감을 줄 수 있다.
● 현실적인 AWS 정보(강수량 및 습도, 온도 등)들을 입력자료로 활용하는가?	● 모든 raw-data들을 직접 AWS로부터 받아 데이터베이스에 반영하였기 때문에 매우 현실성이 있다.
● 다른 인터넷 서비스 정보도 효율적으로 검색, 사용할수 있는가?	● 농업 정보 관련 사이트를 링크시켜줌으로써 타 농업 관련 서비스를 쉽게 사용할 수 있도록 하였다.
● 사용자의 이용이 과다해도 서버의 안정성이 보장되는가?	● Java Web Server의 Servlet을 이용함으로써 많은 사용자의 요구를 받아들일 수 있다.
● 시스템의 실용성이 있는가 ?	● 현실적인 연구가 되기 위하여 다음의 연구를 수행하였다. - AWS 관리와 AWS 정보 입력기 개발 (단일 인터페이스로 많은 AWS들을 관리 할수있다.)
● 시스템이 확장성이 있는가 ?	● Java 기반이기 때문에 다른 이기종 시스템과 직접 연동이 가능하다. ● 일반적인 DBMS를 사용하였기 때문에 다른 작물에서 본 시스템을 사용하고자 할 경우 알고리즘만 수정하면 바로 사용할 수 있다. (추가 연구)

3. 관련분야의 기술발전예의 기여도

가. Java 기술을 이용한 C/S 개발

본 과제에 가장 큰 공헌은 향후 구축될 시스템들간에 잘 연동 할 수 있는 기본 소프트웨어 개발이었다. 이는 인터넷 확장성과 이식성이 뛰어난 Java언어로 시스템을 구축하였기 때문에 가능한 것이다.

본 과제에서 사용된 Java는 현재 많은 벤더들의 호응에 힘입어 다양한 플랫폼으로 그 유용성이 지금도 확대되고 있다. 그러나 아직 국내에서는 이렇나할 Java 기반 농업 관련 정보시스템 소프트웨어 관련 제품이 출시되지 않고 있는 실정이다. 본 과제는 이러한 국내의 Java에 대한 관심을 불러일으키기에 충분할 뿐 아니라 Java로 개발된 시스템이 다양한 이 기종 시스템과의 상호 연동될 수 있다는 가능성을 보여주어 이와 관련된 분야에 대한 기술 발전에 기여할 수 있다.

나. AWS 정보 관리 시스템

기존에는 데이터베이스에 데이터를 저장하기 위하여 데이터베이스의 스키마를 알아야하거나, SQL 문에 대한 지식을 미리 습득하고 있어야 하는 등의 단점이 있었으나 본 시스템에서는 관리자가 비주얼한 그래픽을 통하여 강우량, 습도, 온도 등의 AWS 정보 raw-data를 쉽게 저장할 수 있도록 하였다. 이러한 것은 직관적이고 사용하기 쉬운 사용자 인터페이스와 결합하여 그 유용성이 배가되며 이러한 관련 기술의 발전을 촉진시킬 수 있는 기회가 될 것이다.

제4절 연구 개발 결과의 활용 계획

1. 추가 연구의 필요성

본 과제 의 가장 큰 공헌은 향후 구축될 농업 관련 정보 시스템과 잘 연동 할 수 있는 소프트웨어를 클라이언트/서버 개념으로 개발한 것이었다. 이는 인터넷 확장성과 이식성이 뛰어난 Java언어로 시스템을 구축하였기 때문에 가능하였다. 그러나, 이를 좀더 확장하여 CORBA가 가지는 분산객체 개념을 잘 응용하고 향후 농업 관련 정보 시스템에 곧바로 적용할 수 있는 CORBA기반 분산객체 중앙 관리 시스템 소프트웨어를 개발해야 할 것이며, 본 과제의 결과물이 그러한 연구의 기초가 될 것이다.

2. 타 연구에의 응용

본 과제의 연구 결과를 활용하기 위한 방안으로는 크게 실용적인 면과 연구적인 측면으로 나눌 수 있다. 실용적인 활용 면에서는 주관연구기관에서 변경된 AWS 정보의 수정을 거쳐 강원도 일대에 대한 감자 관련 정보 서비스를 웹 환경에서 일반 사용자들에게 제공해 줄 수 있다. 그리고 기존의 농업 관련 방송, 하이텔이나 천리안에서 제공하는 농업 정보 등과 연계하여 사용 할 수 있다.

연구적인 활용 면에서는 본 과제에서 획득한 인터넷, WWW, 클라이언트/서버, Java, JDBC, CORBA 프로그래밍, 멀티미디어, 인공지능과 실시간 컴퓨팅의 통합, 시스템 통합, 그리고 객체지향형 DBMS 기술을 활용하여 향후 국가적 차원에서 농업 관련 정보 인프라 구축에 적극 응용할 수 있다.

참 고 문 헌

- [1] A. V. Hof, S. Shaio, O. Starbuck, Hooked On Java, SUN Microsystems, 1996
- [2] B. F. Rasmussen, A Web to Database Interface. Sybase WWW Home Page, 1995
- [3] IBM, DB2 World Wide Web Connection, IBM, 1995
- [4] ORACLE, "오라클과 인터넷", ORACLE White Paper, 1996
- [5] T. Ritchey. Java, New Rider, 1995
- [6] 신명기, 최희창, 박경순, "인터넷 에스페란토 자바", 마이크로소프트웨어, 1996, 1월호, pp. 262-297
- [7] M. Breeding, Mecklermedia's Official Internet World : World Wide Web Yellow Pages 1996, Mecklermedia Corporation, Bk&Cd-Rom Edition, 1996.
- [8] K. Milburn and J. Burdman, Designing Web Pages with FrontPage 97, New Riders Publishing, 1996.
- [9] WebWise!: The Cyberia Guide to Smart Web, Cyberia, Publishing, McGraw-Hill, 1996.
- [10] Java WWW Home Page, <http://java.sun.com>, 1998
- [11] <http://jserv.javasoft.com/products/java-server/webserver/fcs/doc/servlets/api.html>
- [12] <http://jserv.javasoft.com/products/java-server/index.shtml>
- [13] <http://jserv.javasoft.com/products/java-server/toolkit/>
- [14] <http://jserv.javasoft.com/products/java-server/sdk/>

- [15] <http://jserv.javasoft.com/products/java-server/ncserver/>
- [16] <http://jserv.javasoft.com/products/java-server/toolkit/index.shtml>
- [17] <http://java.sun.com/products/jdbc/>
- [18] Art Taylor "JDBC Developer's Resource" INFORMIX Press
- [19] Ashton Hobbs "Database Programming with JDBC in 21days"
Sams Net
- [20] <http://java.sun.com/products/jdbc/jdbc.drivers.html>