

Evaluation Methodology of a Web-based Greenhouse Intelligent Management System

Michael T. Maliappis and K. P. Ferentinos

Abstract—Successful evaluation of a software system is of vital importance for its acceptance by end-users. This paper describes the evaluation process and characteristics of a system developed to introduce computer management into the cultivation process in low-tech greenhouses. The proposed system is implemented as a web-based application using open source technologies and subsystems comprised of modules that provide: (1) static information about the cultivation process and marketing of the supported crops, (2) simulation and forecast models of general interest, (3) a collaboration environment and (4) expert system capabilities and support. Modular construction of the system allows easy adaptation to new products and expansion into new languages. System modularity is a major factor affecting its evaluation. Although several characteristics, such as user interface, can be evaluated as a whole, individual modules have their own peculiarities affecting the evaluation process.

I. INTRODUCTION

FOR any system to be useful to its target audience, beyond its developers, it should be tested extensively and evaluated thoroughly. There are several kinds of testing and evaluations, such as unit testing, functional testing, acceptance testing, performance and environmental testing, validation and verification, formal analysis and so on. Some of them are considered necessary for any kind of system and others are target to specific kinds of software systems. Some of the tests are carried out by the developers, others by specialized personnel and others require extensive involvement from the end users [1].

There are two sets of requirements according which software quality is measured. Non-functional requirements which specify criteria that can be used to judge the operation of the system rather than specific behaviors, and functional requirements that specify specific behavior of functions. Non-functional requirements are often called qualities of a system or quality of service requirements. Quality of service requirements can be divided into two main categories: a) execution quality, such as security and usability, and b) evolution qualities, such as extensibility and scalability.

Software quality, as a measure of system usability, is a highly subjective term and is depended heavily on the users

of the system under consideration. What is valuable to the potential audience constitutes the criteria in measuring software quality. Of course there are criteria crossing all different kinds of software but their weight on the final result is generally different [2].

The present paper treats evaluation criteria measuring system usability for a Greenhouse Intelligent Management System (GIMS) that has been developed to provide computer management in low-tech greenhouses cultivating tomato, pepper, or aubergine. The system has been designed as a web-based application and integrates and extends previous efforts [3,4] by increasing their accessibility and applicability.

The cultivation of out of season vegetables in plastic-covered greenhouses is a major economic activity within the countries of the Mediterranean basin. In Greece, there are over 4000 ha of greenhouses, of which 95% are plastic covered and only 5% glass [5]. The principal greenhouse vegetable crops are tomatoes and cucumbers, but significant quantities of peppers, aubergines, melons and lettuce are also grown. Because of the mild climate, most greenhouse vegetable crops are either unheated or provided with a minimal level of heating, and the level of automation (e.g. for ventilation) is low.

The system implements several modules, which cover the collaboration, information and management needs of greenhouse growers by using web-based technology. All the modules are based on open source technologies and industry standards, thus achieving low cost and increasing portability, extensibility and communication with other systems. The GIMS contains modules offering static information about the cultivation process and marketing of the supported crops, simulation and forecast models of general interest and expert systems for irrigation management and pest control.

For each one of these modules a special set of evaluation criteria is needed to measure their quality as well as their user satisfaction. The satisfaction level of the end-user (student, grower or extension service) constitutes a basic index of performance of any information system. The quality of service and its improvement presupposes the measurement of satisfaction based on the necessities and desires of the end-user. The evaluation of satisfaction of the end-user is approached as a problem of analysis of satisfaction depending on a set of factors or characteristics of the offered system.

The following sections describe the design principles,

Manuscript submitted April 21, 2008. This work was supported in part by the "PYTHAGORAS-II" research project (EPEAEK II).

Michael T. Maliappis is with the Informatics Laboratory, Agricultural University of Athens, Greece (corresponding author, phone: +30-2105294451, fax: +30-2104294199; e-mail: michael@aua.gr).

K. P. Ferentinos, is with the Informatics Laboratory, Agricultural University of Athens, Greece, (e-mail: kpf3@cornell.edu).

system architecture and the modules constituting the system. For each module a special set of evaluation criteria has been defined and examined. The paper concludes with a discussion of the benefits and limitations of the system, the current stage of development and its future perspective.

II. SYSTEM DESIGN PRINCIPLES AND OBJECTIVES

The main objectives of the system are to provide a collaborative environment and an assistance tool for greenhouse growers or students which could offer useful information and shared knowledge to practice their daily tasks. To reach these objectives the following set of design principles have been selected for system development:

- 1) *Web based application*: This kind of application is the only way to ensure system availability and maintainability. Furthermore, keeps user investment in hardware and software at the minimum level moving computational responsibilities to the server.
- 2) *Open architecture*: Based on open source software and open standards the system keeps its development cost low and can be benefited from other systems or sources of information and knowledge following these widely used and publicly available standards.
- 3) *Modularity*: It is the property of a system that has been decomposed into a set of cohesive and loosely coupled modules [6] and is perhaps the most widely accepted quality objective for software design [7]. Its adoption reduces system complexity and improves system manageability and maintainability. It allows incremental development of the system as well as easy extensibility, and promotes customization. Modularity allows the addition of new features, or modification of the old ones, without mix-up with the functioning of other modules preserving system stability.
- 4) *Central management*: Since one of the main goals of the designed system is to provide useful information, centralized management facilitates adaptation to a constant renewal of information and knowledge. Central management ensures that the provided information and knowledge is always up-to-date. Additionally, management actions are concentrated at the most appropriate place.
- 5) *Personalization*: Personalization provides users with an isolated operation environment. Authenticated users have their own separate storage space where they can insert data about their own greenhouses, process them and extract their own results. In addition, users can initiate a consultation session and invite other users or experts to participate in discussions to solve particular problems or exchange ideas.
- 6) *Internationalization*: A major attraction of the system is its ability to function in several languages. Language selection can be implemented at three points: presentation level, application level and knowledge level. Since the system interface is implemented using

Java language and J2EE technology, following J2EE guidelines for software internationalization is enough. Information material uses XML and XSL to accomplish the goal of internationalization. Expert systems are multilingual by design. Knowledge bases use a special multilingual structure, as described in [8]. This particular knowledge structure allows work with the automatically selected knowledge base in the specified language.

During system development cycle the development team tried to follow these principles, but the successful achievement of these can only be judged by the end users. System evaluation by end users should decide the degree of completion of these objectives.

III. SYSTEM COMPONENTS

GIMS has been implemented as a web-based application based on open source technologies and subsystems. It has been implemented using a J2EE (Java 2 Enterprise Edition) platform, a standard for implementing and deploying scalable, reliable and securely distributed applications. The J2EE technology is designed to support the rigorous demands of mission-critical application systems distributed on a large scale and provides support for multi-tier application architecture.

GIMS follows object-oriented design principles and is developed using Java programming language and related technologies. In developing a web-based application the use of a development framework facilitates the process and offers a strict discipline. For this specific purpose Struts¹ framework has been used. Struts is an open source framework, provided by the Apache Foundation, for developing web based applications. It uses and extends the Java Servlet API to encourage developers to adopt a model-view-controller (MVC) architecture. The goal of Struts is to cleanly separate the model (application logic that interacts with a database) from the view (HTML pages presented to the client) and the controller (instance that passes information between view and model). Struts provides the controller (a specialized servlet) and facilitates the writing of templates for the view or presentation layer, typically in JavaServer Pages (JSP). The web application programmer is responsible for writing the model code, and for creating a central configuration file which binds together model, view and controller.

The system uses a 3-tier architecture comprised of a presentation tier, a middle tier containing web and service components, and a storage tier. Business and data logic are usually partitioned into separate components and can be deployed on one or more application servers. This architecture, shown in Fig. 1, can be deployed using one or more application servers. One application server usually encapsulates web and service components, while the storage

¹ <http://struts.apache.org>

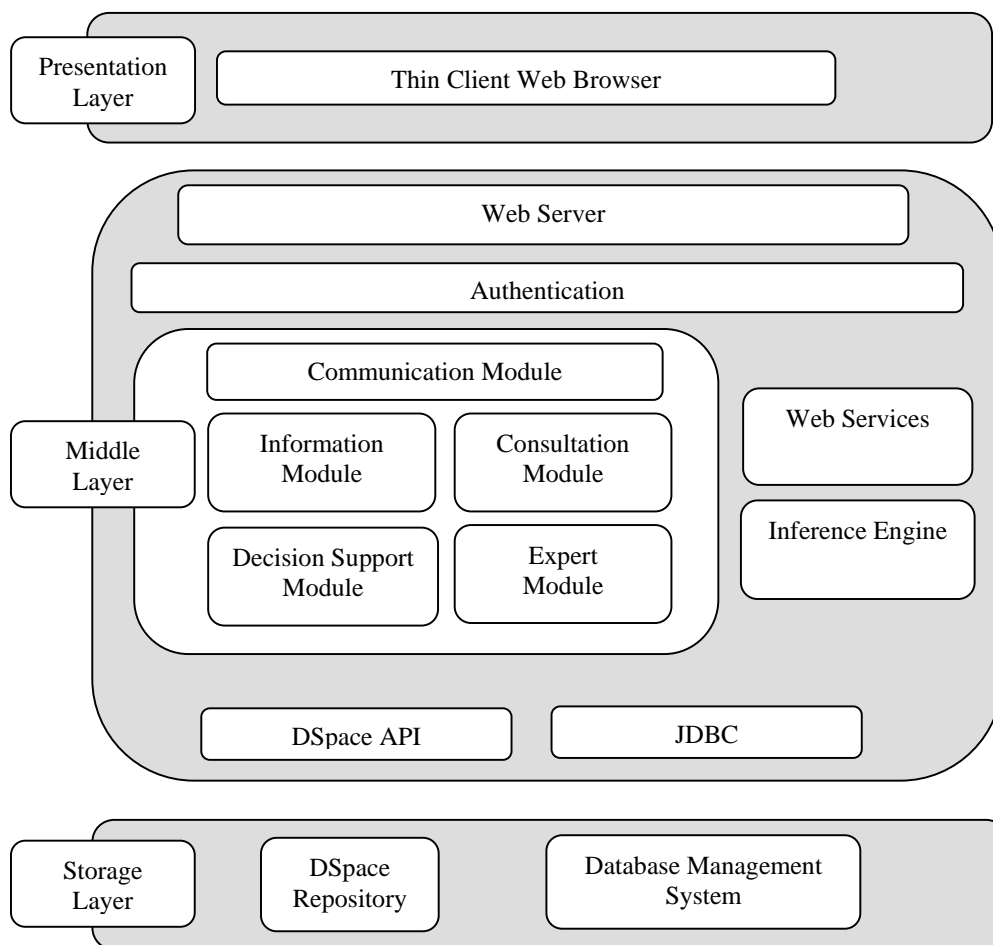


Figure 1 System Architecture

tier can be deployed on the same or a separate server. This partitioning of the application into multiple server components allows components to be easily replicated and distributed across the system, ensuring scalability, availability, reliability and performance.

The client tier is a very “thin” tier that contains only presentation logic. Communication with the web server, seated at the middle layer, is accomplished using HTTP protocol through a simple web browser. Each user is able to connect into GIMS using the web browser available on his/her machine. Thus, on the client site a computer is needed with an Internet connection and a web browser such as Firefox, Netscape or Microsoft Internet Explorer. If Firefox is used no other software is needed since it incorporates an RSS reader, otherwise an RSS reader is necessary for RSS feeds to be exploited.

In the middle layer the most important component is the application server, a web server with intrinsic capabilities to execute Java servlets and support JSP pages, which is able to support the development of integrated web-based applications. The web server handles communication with

the external environment and routes external calls to appropriate GIMS components. In this particular case Apache Tomcat² has been used as a web and application server. It is an open source application server provided by the Apache Foundation. Tomcat implements the Java Servlet and the JSP specifications, providing an environment for Java code to run in cooperation with a web server. Tomcat includes its own internal HTTP server.

The storage level contains the database of the system. It can be separated into two distinct components. The first is an ordinary database management system (DBMS) used for data, information and knowledge storage, manipulation and retrieval and is implemented by PostgreSQL³ DBMS, an open source system with professional capabilities in data manipulation. Database is used to store data concerning the cultivation procedures for each greenhouse and the cultivation period, as well as the knowledge needed by the expert systems. The second component is the DSpace repository (see Consultation Module) which is used as a

² <http://tomcat.apache.org/>

³ <http://www.postgresql.org/>

storage medium in the consultation module. The DSpace repository is implemented over the PostgreSQL DBMS.

The web and service tier communicates with the data tier through the JDBC (Java Data Base Connectivity) protocol.

The contents of the system are organized as separate modules, which can be divided into four major categories

The GIMS allows registered users to upload their own files and information documents according to the structure of domain ontology. Ontology [9] is a formal explicit description of concepts in a discourse domain that provides a common understanding for a particular domain. Ontologies can be used to describe knowledge domains and

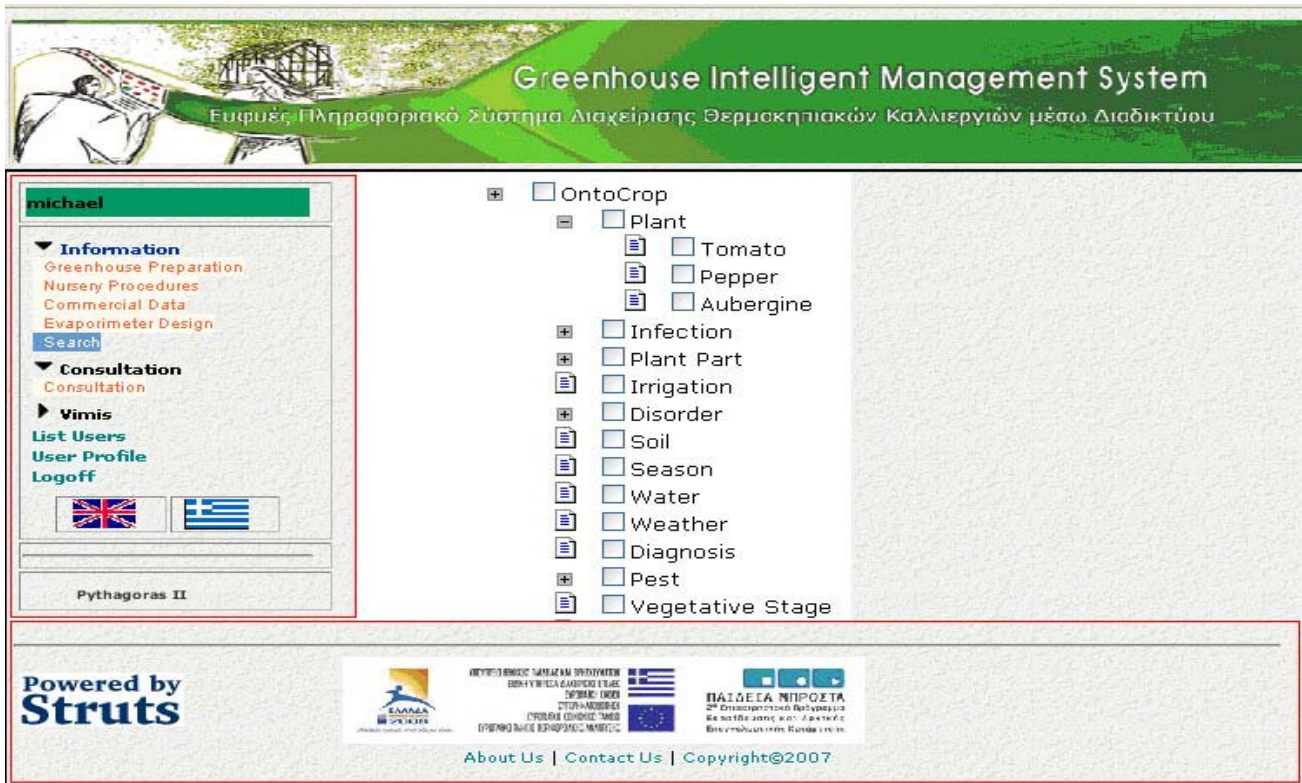


Figure 2 Searching Through Ontology

providing several facilities: a static and dynamic information module, a consultation module, a decision support module and an expert module. Apart from these four major modules there is also a communication module. The contents and basic characteristics of these modules are described in the following paragraphs.

A. Information module

The static information provided concerns the crops' cultivation procedures and marketing information. Information concerning the cultivation and marketing of tomato, pepper and aubergine is provided using static pages. Information is stored in properly structured XML files, which are converted to ordinary HTML pages by the use of proper XSL files. This kind of information covers greenhouse preparation, nursery and cultivation procedures, post-harvest and commercial data. Because some crops have certain pieces of information in common with other crops, the XML files used are considered jointly so as to enable their integration and keep redundancy to a minimum.

to support a search, or to navigate through a body of knowledge to assist knowledge mining. Ontologies are useful for knowledge sharing and re-use, and are used to homogenize heterogeneous sources of knowledge and provide a common level of presentation. Ontologies can be used to achieve taxonomy of knowledge. A taxonomy is a hierarchical structure for organizing a body of knowledge; it gives a framework for understanding and classifying that knowledge – how to group it and shows how the various groups relate to each other. In content management, the purpose of taxonomy is to organize information so that users can navigate their way through it more easily.

For this particular situation, OntoCrop ontology [10] is used as a classification and navigation vehicle. OntoCrop covers a portion of the horticultural domain concerning the cultivation of vegetable crops in low-technology greenhouses. It contains knowledge concerning cultivation techniques, pest management and crop physiology. It has also been extended to include knowledge about propagation, post-harvest physiology, consumption and marketing.

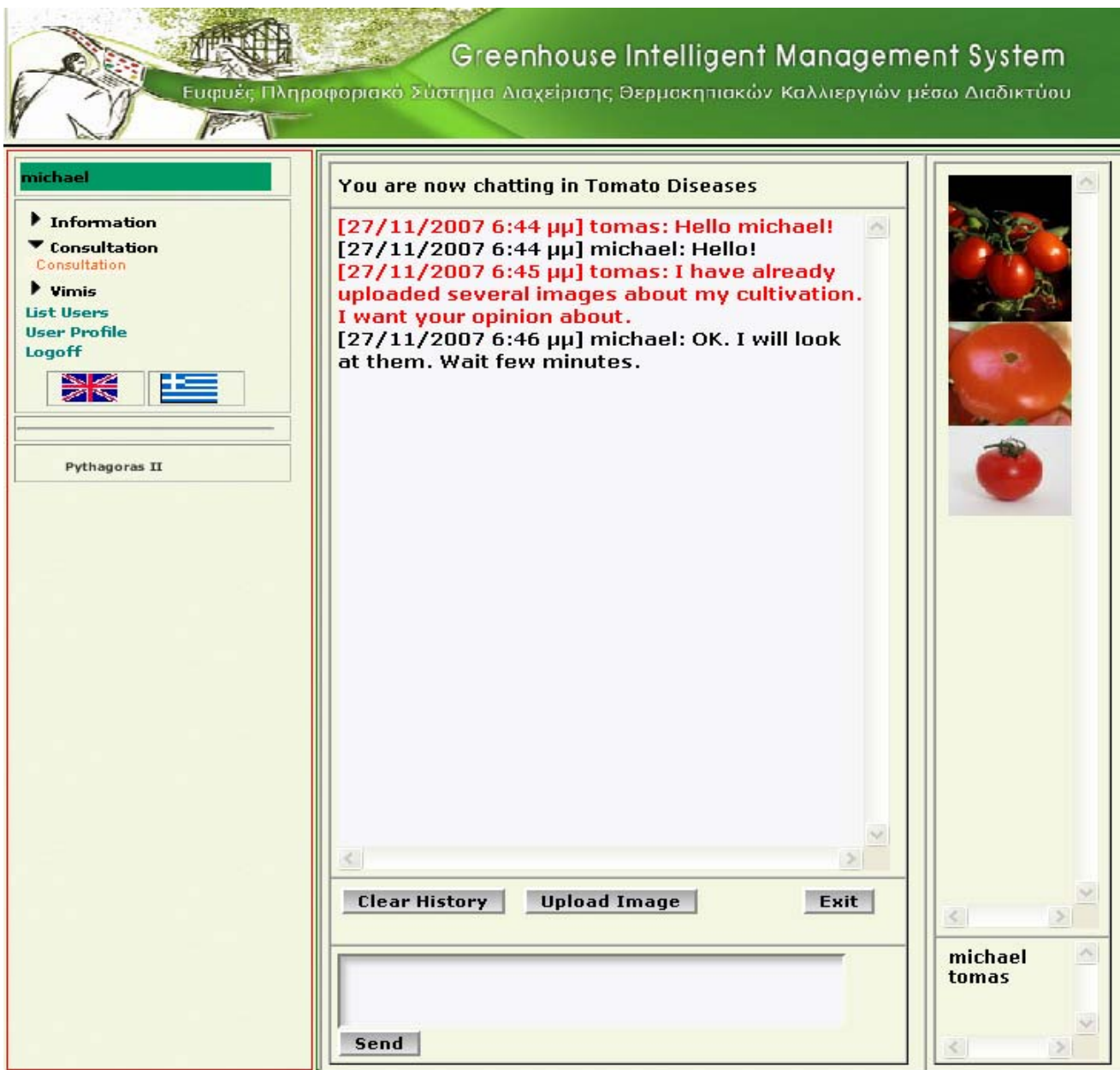


Figure 3 Consultation Module Snapshot

Using OntoCrop, users are able to upload their own documents and specify their position in the ontology structure. At the same time they have the opportunity to navigate through the stored documents using the provided ontology. Fig. 2 shows a screenshot of the search process through ontology usage.

B. Consultation module

The consultation module is a chat-like application enriched with image uploading facilities. Any registered user can initiate an individual private or public session concerning a specific problem or a subject of general

interest. Private sessions can be joined by users having an invitation from the owner. Public sessions are open to any authenticated user. During sessions users are able to exchange written messages and upload multimedia files visible by anyone participating in the session.

Fig. 3 shows a snapshot of a consultation process between two users discussing the topic of tomato diseases. During the discussion several images have been uploaded, shown on the left side of the screen as thumbnails. Users are able to download and inspect images in full resolution.

The whole conversation process, including uploaded files,

can be saved by the session owner in a DSpace⁴ repository for future reference. DSpace is an open source software platform that enables organizations to capture, describe and preserve digital assets and provides distributed access to these assets through a search and retrieval subsystem. It is written in Java and JSP and provides its holdings primarily via a web interface.

Session documents and multimedia files can be saved into a DSpace repository as best practice examples and could be useful should similar situations arise in the future. The use of DSpace provides for longer viability and dissemination to a larger audience.

C. Communication module

One of the goals in the system design has been to develop a collaborative environment for its users. For this reason the system has been equipped with components such as an RSS feeder and an email alert subsystem. The communication module distributes news using RSS feeders and mail alerts in the case of something concerning a specific area, such as extreme natural phenomena or some disease problem within the neighborhood.

The RSS (Really Simple Syndication) feeder is a kind of live bookmark and is used for the automatic feeding of news. The system's server site uses XML to format summaries for important news or announcements incorporating the actual URL of the source. Clients using special RSS readers or aggregators are able to exploit these sources of information. Instead of having to remember to access the web site and look for altered information, the RSS feeder forwards the summary of the updated information to the client's RSS reader and after a review the client can access the full subject through the incorporated URL.

Mail alerts can be used to forward urgent messages directly to the growers using an ordinary email server. For this service the user needs a valid email address. In addition, by using the system's capabilities an authenticated user is able to schedule his/her activities by setting specialized alerts.

D. Decision support module

This subsystem provides a way to schedule, supervise and analyze the cultivation procedures, keeping records of the grower's activities and having the system remind the grower with special alerts about essential actions. The grower inserts his daily activities and events into the system according to their occurrence. The system is able to suggest subsequent actions within a certain future horizon and at the same time to show any problems in the sequence of the previous actions. In case of deviations from the schedule, the system alerts the grower and suggests possible corrective actions. The subsystem uses simulation and statistical techniques, expert system analysis as well as check lists to

accomplish its tasks.

The system handles each greenhouse as a distinct unit of reference. The relevant data are stored and retrieved according to this unit of identification, the year of cultivation and the cultivation period in the year. For each greenhouse, data concerning the type of the construction are kept, as well as the geographic location, the crop and variety under cultivation, the prevailing climatic conditions etc. Furthermore, the system keeps track of any treatments during the cultivation period, noting the quantities and time of application of insecticides and pesticides. The irrigation needs of the crop, the fertilizer requirements and the detection of any nutritional imbalances are all achieved through the operation of the expert module, described below. Analysis of data takes into consideration the stage of the cultivation process and the growth level of the crop under investigation.

E. Expert module

The expert module contains two expert systems which are provided to help growers with nutritional deficiencies and pest management, as well as to regulate irrigation.

The first expert system is an adaptation of the VEGES expert system [4] to be used as a web based application. It can be used for the identification of pests, diseases, and nutritional disorders. As a result, VEGES identifies the disorder according to the input data provided by the grower and suggests methods of confrontation. The second expert system extends IRNMA, an irrigation and nutrition management system [3]. This defines the daily requirements of the plant for irrigation and fertilizer application relative to the stage of growth, plant population, soil type, and the amount of water evaporating from the soil, as measured daily by a simple, manually operated evaporimeter.

For each expert system, the same knowledge base can be used by all users of the system. The consultation process is guided by input data provided separately by each user. The centralized management of the knowledge base provides better quality knowledge.

The inference engine [11] is accessible through web services. The user provides the symptoms to the system using special forms. The system converts the data provided into XML format and submits them to the inference engine using the appropriate web service. The results follow the reverse procedure and are displayed to the user using HTML. Every inference trial is stored in a database for future reference.

IV. EVALUATING GIMS

The system provides a common user interface and the individual components are incorporated into this interface providing their functionality. Evaluation of the user interface can be considered as a unified task judging the common characteristics of the whole system and the special characteristics of the modules.

⁴ <http://www.dspace.org>

The system, being a web-based application, should be evaluated against its look and feel, readability and performance. Characteristics concerning navigation, functionality, control, feedback, consistency, error prevention and correction and visual clarity should be taken into consideration.

Since the system has a modular structure, each component or module should be evaluated separately. The evaluation process should take into consideration individual characteristics and peculiarities of each module and apply the proper process.

Information module: This module can be checked for the completeness and accuracy of the provided information. Module upload and search capabilities are depended on OntoCrop ontology which needs extensive evaluation for its completeness.

Consultation module: Basic characteristics of the module are the availability of a rich set of capabilities and its response speed in heavy load.

Communication module: The module is responsible for instant delivery of crucial messages and information. Its performance should be evaluated against its response speed and the way it communicates its messages.

Decision support module: The module provides guidance for actions based on data provided by the user. It should be evaluated against its capability to give accurate and useful instructions. Additionally, its potential to act in case of incomplete information is a factor under consideration.

Expert module: The evaluation of expert module has two scales. The first is inference engine evaluation and the second is knowledge base validation. Inference engine should be accurate and should provide alternative inference mechanisms in case of incomplete knowledge. Knowledge base should be complete and consistent [12, 13].

During the implementation of a program of measuring users' satisfaction, quantitative and qualitative searches of satisfaction are used. When developing a questionnaire of a satisfaction search, the dimensions of satisfaction are determined; that is a consistent family of criteria characterized by monotony, completeness and non-redundancy. Each search of satisfaction should be characterized by reliability, validity, lack of systematic errors, and accuracy.

V. CONCLUSION

The users' satisfaction constitutes the most important factor for the development of a total quality process based on the necessities, the desires and the standards of the end-user. With the goal of overall quality, the continuous improvement of the system is pursued. The user observes the level of performance of the system and acquires with certainty whether the system owns some characteristic of quality. A comparison analysis of the user's expectations with the offered services by the system takes place.

The system described provides a web-based application

with centralized knowledge and information management having extended personalization and internationalization characteristics. Its potential accessibility from anywhere and integrated environment with collaboration capabilities, provide a way to increase information technology penetration into growers of low-tech greenhouses.

The system, at its present stage of development, handles three important greenhouse crops (tomato, pepper and aubergines), which belong to the same family (*Solanaceae*) and present similarities in their cultivation methods. The system architecture exploits these similarities at the knowledge structure of its intelligent components and in the way it stores static information concerning each crop.

For such a tool to be useful, the active involvement of the growers is needed so as to share their knowledge, communicate their problems and ask for solutions. The ideal operator of such a system is a dynamic union of producers in cooperation with specialists in the field to provide useful information and added value services to their members. Emphasis has been given to the easy management of the system without the need for special technical skills. Unions are in a position to use the system for communication among their members and to communicate important news using the system's technical facilities.

Because it evolved from a research program in an agricultural university environment, the system has an immediate application within the field of horticultural education. The students can use the system within the classroom to acquire information and solve practical problems (e.g. disease assessment under virtual reality). However, in addition of being helped in their own educational tasks, the students will also act as front line testers of the system, improving its capabilities.

Modular construction of the system allows ease of expansion. Features such as forums can be easily established for communication, the exchange of ideas, sharing of knowledge and for expressing problems. Since the system contains information about certain crops, it is possible to be used as a source or a point of negotiation through an eCommerce component. The grower will be able to select whether he/she wants his/her production to be monitored and included in this process.

GIMS is under an evaluation process following the criteria and principles described in this paper. A hybrid method of evaluation has been selected containing questionnaires and direct involvement of end-users. The first results are expected early in the summer of 2008.

The main characteristics of the system are the simplicity of use of the system succeeded with the software use and the effective handling of the data, the friendliness provided to the users, the effectiveness of extraction of certain conclusions and definition of propositions of improvement by the users.

REFERENCES

- [1] Pressman, Scott. *Software Engineering: A Practitioner's Approach*. McGraw-Hill International, Sixth Edition International, 2005.
- [2] Dumas, J. S. and Redish, J. C., *A Practical Guide to Usability Testing* (revised ed.), Bristol, U.K., Intellect Books, 1999.
- [3] Passam, H. C., A. B. Sideridis, C. P. Yialouris, and M. T. Maliappis, "Improvement of Vegetable Quality and Water and Fertilizer Utilization in Low-Tech Greenhouses through a Decision Support Management System." *Journal of Vegetable Crop Production*, vol. 7, no. 1, 2001, pp: 69-82.
- [4] Yialouris, C. P., H. C. Passam, A. B. Sideridis, and C. Metin, "VEGES: A Multilingual Expert System for Diagnosis of Pests, Diseases and Nutritional Disorders of Six Greenhouse Vegetables." *Computers and Electronics in Agriculture*, vol. 19, 1997, pp: 55-67.
- [5] Mavrogiannopoulos, G. N., *Greenhouses*. Athens: Stamoulis Publications, 3rd ed., 2001, (in Greek).
- [6] Booch, G., R. A. Maksimchuk, M. W Engel, B. J Young, J. Conallen, and K. A. Houston, *Object-Oriented Analysis and Design with Applications*. 3rd Edition. Addison-Wesley, 2007.
- [7] Card, D. N., and R. L. Glass, *Measuring Software Design Quality*. Prentice-Hall, Inc, 1990.
- [8] Maliappis, M. T., and A. B. Sideridis, "A Framework of Knowledge Versioning Management." *Expert Systems*, vol. 21, no. 3, 2004, pp: 149-56.
- [9] Beck, Howard W., and Helena Sofia Pinto, "Overview of Approach, Methodologies, Standards, and Tools for Ontologies, Agricultural Ontology Service", UN FAO, 2002.
- [10] Maliappis, M. T., "Investigating Ontology Development and Usage in Horticultural Production", In the Proceedings of the EFITA/WCCA 2007 Conference, Glasgow, Scotland, 2007.
- [11] Maliappis M. T., "Design and Implementation of a Knowledge Base Versioning Management Framework", Ph.D. dissertation, Informatics Laboratory, Agricultural University of Athens, Greece, 2006.
- [12] Preece, Alun D. "Validation of Knowledge-Based Systems: The State-of-the-Art in North America." *The Journal for the Integrated Study of Artificial Intelligence Cognitive Science and Applied Epistemology* vol. 11, no. 4, 1994.
- [13] Preece, Alun D., "Building the Right System Right: Evaluating V&V Methods in Knowledge Engineering". *KAW'98*, 1998.