Building a Web Advisor for Integrated Protection

Anna Perini[1]

Abstract. Basic issues to be addressed in a new project aimed at developing a web-based advisor for integrated plant protection which uses Case-Based Reasoning techniques are described and discussed. The integrated protection problem and the decision making process underlying the problem solving are presented.

Some of the critical aspects in applying case-based reasoning techniques to support this decision making problem are discussed.

Two more issues, dealing mainly with methodological aspects, are analyzed: system requirement specifications and maintenance problems.

1 INTRODUCTION

Integrated Protection (IP) consists of a set of practices aimed at favoring the set up of an agricultural development model characterized by a reduced environmental impact.

Among the basic principles of IP in plant diseases management:

- a tolerance threshold under which plant disease damages can be economically accepted;
- a classification of chemicals and techniques, with respect to their impact on the environment, including growers and consumers. A list of authorized chemical products is provided by the International Organization for Biological Control, further revised by the local government.

So, the application of IP practices in plant disease management by growers and agronomists requires:

- a large knowledge of plant diseases evolution, in order to be able to estimate the risk of overcoming the tolerance threshold disease;
- updated information on chemicals and techniques.

Our interest at IRST in this application domain dates back to 1990. The goal was that of verifying the usefulness of Artificial Intelligence (AI) techniques for setting up decision support systems for the use of agronomists and growers. This work was conducted in collaboration with the Istituto Agrario S. Michele all'Adige (IASMA) and, in particular, with the lab doing research on plant disease defensive techniques.

Our approach, following to the state of the art of knowledge-based applications to agriculture [1], was that of identifying a precise problem and of building a specific knowledge-based system that could help find solutions for it. We concentrated on Integrated Pest Management (IPM) problems separating the pest diagnosis problem from the remedy actions planning problem.

We developed a pest diagnosis advisor for apple orchards, for the use of farmers [2] and extended it with a module that gives advice to growers on seasonal pest control activities to be performed in order to monitor and control the presence of serious pests in the orchard [3]. The system was validated by thirty expert technicians of the Agriculture Advisory Service of our province [4]. Among the most relevant results of this work is a set of requirements for a decision support system for technicians.

Recent approaches of AI applications to agriculture, and more generally to environmental problems, tend to adopt a “systemic” approach (e.g. [5]).

That is to say a problem is considered for its impact on the involved organizations and the proposed applications aim to be integrated in larger information systems exploiting the fact that different organizations may manage information sources and resources that are relevant to problem solutions.

This basically has a twofold effect in applying AI techniques to environmental problems: first, an organizational analysis becomes a necessary step when specifying application requirements; second, different AI techniques need to be integrated resulting in hybrid systems and integrable with specialized software technology, such as DBMS, Geographical Information Systems and WEB technology [6].

The current project is aimed at developing a web-based advisor for the use of technicians when giving advice to growers in plant disease management.

In the following section I will describe the application domain by considering an example of IP application in the case of apple pests. The different knowledge and information sources that enter into the decision making process are also described together with the roles of different domain stakeholders: the farmer, the advisory center technician, the expert in IP directives and in plant diseases.

In section 3 I shall briefly recall the basic features of the Case-Based Reasoning (CBR) approach to problem solving and discuss the motivations that led to the choice of this technique as the most promising in supporting the IP problem discussed here.

In section 4 I shall point out some important methodological issues that will be addressed in the project: first the specification of system requirements, a critical aspect in developing effective software systems, second the CBR system maintenance problem.

2 THE APPLICATION DOMAIN

2.1 Applying integrated protection. An example in apple pest management

Plant pest control - according to IP directives - is to maintain pest damage on crop under the tolerance threshold. That is an economically acceptable number of damaged fruits.

For instance, for the "Aphis Pomi", an important pest for apple orchards in Trentino - a region in Northern Italy -, an acceptable tolerance threshold in May can be expressed as 10% of damaged blossom and a set of defensive actions should be performed in order to approach 0% of damaged fruit by the end of August. This approach requires controlling the orchard every 15 days.

Natural plant protection techniques must be known - for example

[1] Istituto per la Ricerca Scientifica e Tecnologica, ITC-irst,
38050 Povo, Trento, Italy,
email: perini@irst.itc.it

the use of antagonists or low impact pesticides - in order to find the actions capable of reaching the 0% damage threshold goal.

2.2 Skills and information sources. The role of the advisory service technicians.

We analyzed the IPM problem by separating the disease diagnosis problem from the remedy actions planning problem. A further characterization of this second problem shows that it calls for two basic types of decisions.

The first one, corresponds to the choices, made by growers at the beginning of the year, concerning plant disease management strategies to be applied to specific orchards.

These choices rest on risk estimations of plant diseases, derived from a careful analysis of historical data, and on economical goals.

The technician of the advisory center supports the grower in making this decision. Strategic plans are defined taking into account evolution models of plant diseases provided by the IASM labs and tested on the field by the technicians.

These plans combine actions aimed at assessing disease extents and remedy actions.

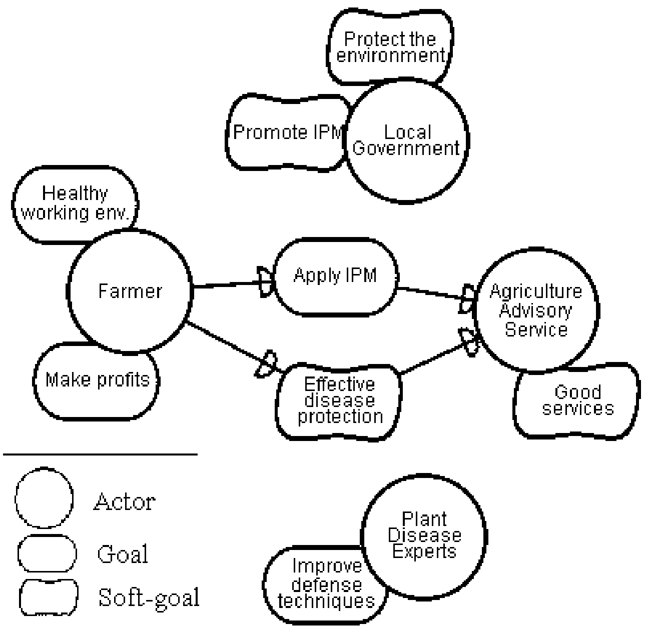

Figure 1: Stakeholders are represented as actors: the farmer that has the *goal* of making profits and having a healthy working environment; the advisory service that has a *soft goal* of providing good services to the farmer; the plant disease experts having the goal of improving their knowledge on diseases and defense techniques; the local government that has two related soft goals: that of protecting the environment and that of promoting a sustainable agriculture approach. Graphical symbols are used according to the i* Strategic Dependency model notation.

So, for this type of decisions, the most critical information are: a correct evaluation of historical data, disease evolution models and precise data on chemicals and on their side effects.

When applying a strategic plan growers and technicians are periodically required to evaluate the data gathered on the field and to find specific remedy actions in case of unforeseen, critical situations due to particular weather conditions or by anomalous disease manifestations. This concerns the second type of decisions.

In this case the most important information are: the weather forecast and the expertise on the application of agronomic techniques.

The technicians of the advisory center record their activities and the main events in reports that are discussed at the end of the season during a collective debriefing meeting.

These reports provide a valuable memory also for the next years activity.

Figure 1 depicts IPM problem stakeholders, their basic goals and dependencies, in i* notation, explained in section 4.1.

3 REASONING TASKS AND AI TECHNIQUES

So far I've described two types of decision making problems related to the IPM problem at issue: the disease diagnosis problem and the remedy actions planning problems.

In this section I'm interested in discussing AI techniques that suit the reasoning tasks that growers and technicians perform while solving one of the two problems.

I shall briefly analyze these reasoning tasks and focus on those that seem to better match CBR technology.

Plant disease diagnosis. Solving this problem requires facing two reasoning tasks that we respectively called *damage observations classification,* and *diagnosis.*

- Damage observations classification consists in associating a set of observed damage manifestations to diseases. This process results in a list of candidate diseases. In [2] we faced this reasoning task by exploiting a discrimination tree of damage descriptions whose leaves correspond to specific damage descriptions and identify a particular set of diseases that, without any further data, can all be responsible for the observed damages. This system can be seen as an electronic practical booklet for growers.
- Diagnosis consists in finding the disease responsible for the observed damages among those in the list. In [2] we exploited the Parsimonius-Covering theory of Peng and Reggia [7] to support an incremental abductive process where newly observed damages could be taken into account, once classified, resulting in a more precise diagnosis.

Planning remedy actions. From the analysis presented in section 2.2 we can reach the following main reasoning tasks: *defining a strategic plan,* currently the responsibility of the technician of the advisory service, *choosing a strategic plan* for a specific farm at the beginning of the year, *managing an unforeseen disease crisis*.

The last two reasoning tasks are the responsibility of each single farmer, but currently, they involve the advisory service technicians.

- Strategic plan definition. As discussed in section 2.2 strategic plans are defined taking into account evolution models of plant diseases and specialized defense techniques. We approached this problem in [3] by applying "skeletal plan" techniques [8].
- Choice of a strategic plan. A farmer has to choose, among a set of predefined strategic plans, the most appropriate to his/her case.
- Managing disease crisis. When an unforeseen event occurs and an appropriate remedy action must be search it can be critical to have information, as quickly as possible, by someone who has managed similar situations as well as by experts in specific defense techniques.

As already stated, the current project will focus on these last two tasks and one of the main efforts will be finding the right way to exploit CBR techniques in tackling them.

One of the reasons that make us confident in the choice of CBR technology is the availability of grower and technician reports on their seasonal activities which, in our opinion, represent an implicit model of the combined phenomena: plant disease (environment) and effects of applied remedy actions. In the following section I will discuss the current hypothesis of CBR utilization in the project.

3.1 Case-based reasoning

The CBR approach rests on the idea that a problem-solving experience can be stored, as a case, in a case memory (called case-base) and that it can be reused to solve a new, similar problem upon adaptation [9].

The CBR technology has been successfully exploited in different application problems ranging from help desk applications, to planning and design [10], [11].

Using CBR to a particular application problem requires to identify parameters relevant for decision making (sometimes called domain modeling), to acquire cases that are described according to these parameters and to maintain the quality of the case base over time.

Three main classes of CBR systems are described in the literature depending, basically, on the type of knowledge sources they use.

They are: Textual CBR systems that exploit collections of know-how documents containing elements useful for problem solutions; Conversational CBR systems typically used in help desk applications where a few questions are needed for decision making; Structural CBR systems that need a domain model definition.

In [12] these three classes have been characterized and compared respect to basic issues to be faced in their utilization, like initial modeling, case creation and knowledge maintenance.

In our project we are currently evaluating two main hypotheses:

- exploiting the available repository of technician reports on their seasonal activity on plant disease management to build a textual CBR system. This choice will require a considerable initial effort in defining domain terms, synonyms, and stop world. Measurements of recall and precision of the CBR system in use will be a critical quality factor. We will consider the opportunity to exploit existing tools for building textual CBR [13].
- defining a domain model of plant diseases and of their IP compliant management techniques and building a structural CBR system. In this case, a major effort is required in defining a suitable case model, the meaningful case features and the appropriate similarity metrics to be used in similarity retrieval. The case acquisition process has to be designed and managed in order to obtain a good case-base in a reasonable time.

In our opinion the first hypothesis will allow to overcome the acquisition problem, while the second should be pursued in order to have a system that allows to build solutions more efficiently.

The CBR technology will characterize the application logic level of the web advisor which rests on a standard distributed system architecture [14], briefly discussed in section 4.2.

4 SYSTEM DEVELOPMENT METHODS

At this point I shall mention two software development methodologies, born in the AI community, that will be used in the project: the i^* framework [15] aimed at supporting requirements specification and the INRECA CBR methodology [12].

4.1 System Requirements Specification with i^*

Requirements specification is a critical activity in developing an application, as largely recognized in software engineering [16].

Underestimating this activity could affect the success of the project. Moreover, when building innovative applications particular attention should be paid in considering their impact on the work organization, that is an analysis of the business process is required.

This aspect becomes particularly critical when building distributed software systems (in particular agent-based systems) and specific methods to support this requirement specification activity are being developed.

The so called i* framework [15] offers a graph based language to model stakeholders (called actors), their intentions in the business process, modeled as goals, and the network of interdependencies among actors (i* stands for “distributed intentionality’’).

Two basic models are provided: the Strategic Dependency model that describes the different types of relationships among actors: goal, task, resource dependencies; the Strategic Rationale Model that describes the reasoning that each actor goes through concerning its relationships with other actors.

These models have been formalized using intentional concepts from AI, such as goal, belief, ability, and commitment (e.g., [17]).

Figure 1 illustrates a strategic dependency model of the main stakeholders in our IPM scenario: the farmers (growers), the technicians of the advisory service, the experts in plant diseases working at the IASMA labs, the local government.

These stakeholders are modeled as actors (circles) which are the nodes of the strategic dependency graph.

The link between *farmers* and *agriculture advisory service* represents the dependency of farmers on the technicians of the advisory service in attaining their goals that they are otherwise unable to achieve on their own, or not as easily, or not as well.

In i* notation goals are represented by ovals and soft-goals by clouds.

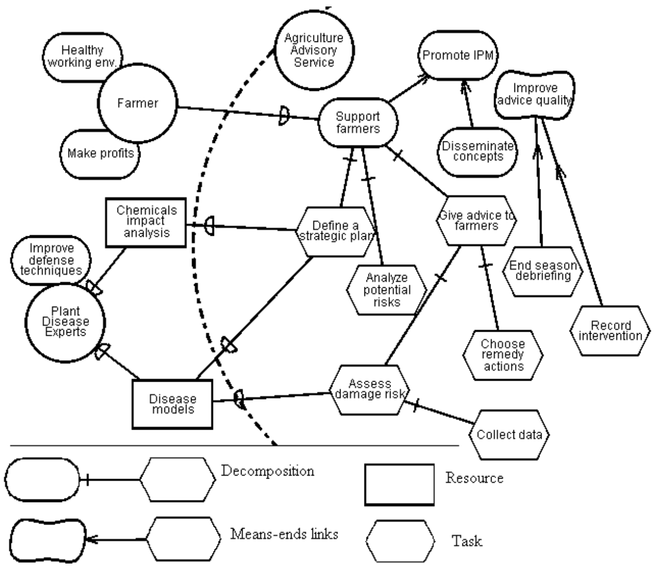

Figure 2: Means-ends analysis for *agriculture advisory service.*

Pursuing further the actor-goals analysis other dependencies can be highlighted, for instance between the *local government* and the *agriculture advisory service*, since the local government delegates to the advisory service its goal of promoting a sustainable agriculture approach and gives it the role of controlling that authorized chemicals only be used.

Once a goal dependency is established from, say, *farmer* to *agriculture advisory service*, the goal is now the *agriculture advisory service*'s responsibility and the goal needs to be analyzed from the *agriculture advisory service*'s perspective.

The recommended way to do this in i* is by analyzing each goal relative to the actor who is responsible for its fulfillment, by pursuing means-ends analysis inside a strategic rational model.

Figure 2 shows portions of such an analysis from the perspective of the *agriculture advisory service* actor.

Conducting means-ends analysis is, for example, to decompose goals into sub-goals and tasks needed for their fulfillment.

A goal may have several different tasks that can fulfil it. For instance the *agriculture advisory service* goal *Promote IPM* can be reach by pursuing the sub-goals *Support farmers* and *Disseminate IPM concepts*. Among the tasks that fulfill the *Support farmer* goal that of defining a strategic plan and of analyzing potential infestation risks, and that of giving advise to farmers on specific disease situations.

This analysis ends with a set of dependencies between the actor and other actors through which the goal can be reached. The final result of this phase is a set of strategic dependencies among actors.

Figure 3 depicts the resulting strategic dependency model that shows the dependencies between the *agriculture advisory service* and the *plant disease experts* for knowledge and data (resources) critical to an effective application of IPM, such as updated disease evolution models and data on the impact of chemicals to the environment.

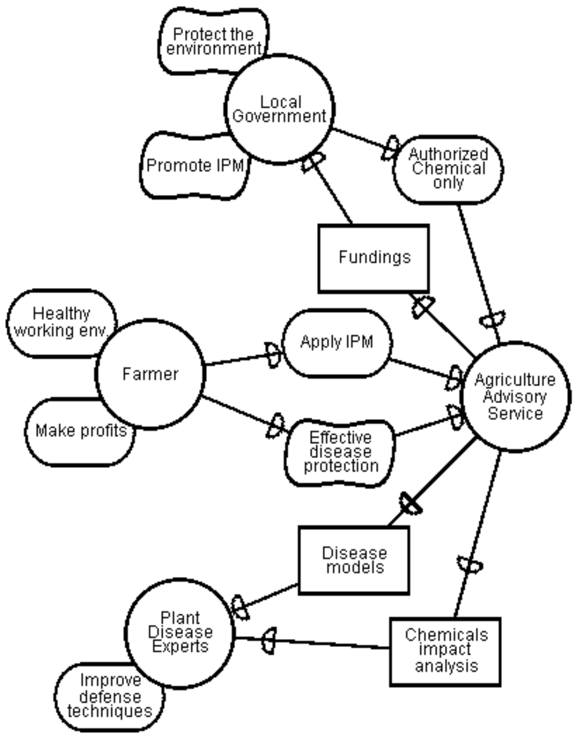

Figure 3: i* Strategic Dependency model resulting from early requirements.

This aspect will be taken into account when designing the web advisor for technicians.

Applying the i* framework in early requirements analysis allows to better understand the key needs of the application's users and to discover possible changes of their roles in the organization and of the way they will interact upon introducing the software application in their work environment.

This view is further expanded in a software development methodology called *Tropos* [18], where a key role is assigned to the analysis of the actors, their goals, and the actors' dependency relationships arising during goal satisfaction, along all the phases of software development, from early requirements, to late requirements, architectural design, detailed design, and implementation. In particular, it can be seen that pursuing a deep analysis of soft goals allows to model non-functional requirements.

4.2 System architecture (a preliminary sketch)

The system architecture will result from the analysis of both functional and non functional requirements.

Figure 4 depicts a preliminary sketch of the system architecture that takes the emerging requirements of a distributed architecture into account.

The system architecture, as discussed in [14], can be organized along the following levels:

- Presentation. That is queries and results pages. In a standard Web architecture it will rests on clients and web server nodes of the physical architecture.
- Application Logic. It may include specialized systems that support the reasoning tasks discussed in section 3.
- Data. It includes data bases of chemicals, activity reports, and disease models.

Moreover the system will need to access to services provided by external systems, such as the regional meteo Web site.

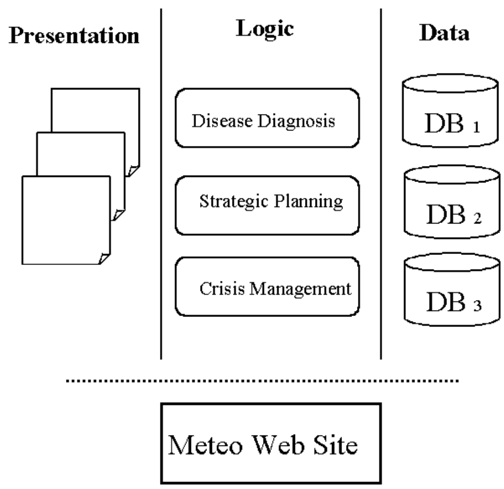

Figure 4: Sketch of the system architecture

4.3 Critical phases in CBR system development process

The INRECA methodology [12] offers a framework to set up a suitable development process of a CBR system, depending on the class of CBR system to be implemented.

It suits the needs met in building simple help desk applications as well as complex equipment maintenance systems based on a hierarchy of subsystems models.

It offers a set of guidelines for both the system development activities and the processes to be activated in order to guarantee a successful utilization of the system.

Among the technical processes discussed by the INRECA framework are those related to generic IT-system development (i.e. system specification, implementation, integration, verification) and those related to the knowledge repository development (i.e. initial knowledge acquisition, core knowledge acquisition).

Moreover a specific process, related to the knowledge repository, is defined for the system use, i.e. the continuous knowledge acquisition and maintenance process.

Particular care should be taken in designing this process in order to take into account changes due to the domain knowledge evolution (for instance new chemicals, disease models updating) and changes induced by the learning mechanism related to the case acquisition process.

5 CONCLUSION

I have presented a project aimed at developing a web-based advisor for plant disease management according to Integrated Protection directives.

A peculiarity of this system will be the integration of current technologies for distributed software systems with a reasoning module resting on CBR techniques.

The paper discusses project motivations and basic issues that will be addressed in the project both at the CBR research level and at the software development methodological level.

ACKNOWLEDGMENTS

The Project is partially funded by the Italian Ministry of Scientific and Technological Research.

I would like to thank the specialists of the plant disease defense of IASM for their collaboration and Paolo Avesani for his precious suggestions.

REFERENCES

[1] R.E. Plant, D. N. Stones, *Knowledge-Based Systems in Agriculture*, McGraw-Hill Inc., 1991.

[2] A. Gerevini, A. Perini, F. Ricci, ' POMI: An Expert System for Integrated Pest Management of Apple Orchards', *AI Applications J.*, **6**:3, 51-62, (1992).

[3] A. Perini, P. Gobbi, D. Forti, M. Delaiti, L. Mattedi, 'A knowledge Based System for Integrated Pest Management', *EFITA Workshop*, Rome, (1996).

[4] A. Perini. et al., ' Valutazione di un prototipo di sistema di supporto alle decisioni per la gestione dei fitofagi del melo', Internal report IRST#9706-02, (1997).

[5] See for instance the DESSAC project at IACR - Rothamsted, UK, http://www.iacr.bbsrc.ac.uk/res/depts/bid/tdessac.html.

[6] P. Avesani, A. Perini, F. Ricci, 'The Twofold Integration of CBR in Decision Support Systems' , *AAAI98 Workshop on Case-Based Integration.*

[7] Y. Peng, J.A. Reggia, *Abductive Inference Models for Diagnostic Problem-Solving*, Springer-Verlag, New York, 1990.

[8] P.E. Friedland, Y. Iwasaki, 'The concept and implementation of Skeletal Plans', *Journal of Automated Reasoning* **1**, 161-208, (1985).

[9] A. Aamodt, E. Plaza, ' Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches', *Artificial Intelligence Communications*, **7**:1, 39-59, (1994).

[10] K. D. Althoff , R. Bergmann, L.K. Branting, *Case-Based Reasoning Research and Development*, ICCBR 99, Springer, 1999.

[11] P. Avesani, A. Perini, F. Ricci, ' Cases on Fire: Applying CBR to Emergency Management ', *Applied Expert Systems*, **5**,175-190 (1999).

[12] R. Bergmann, S. Breen, M. Goeker, M. Manago, S. Wess, *Developing Industrial Case-Based Reasoning Applications. The INRECA-Methodology*, Springer-Verlag ,1999.

[13] I. Watson, *Applying Case-Based Reasoning: Techniques for Enterprise Systems*, Morgan Kaufman Publishers, 1997.

[14] Yen-Ping Shan, R.H. Earle, *Enterprise Computing with Objects*, Reading, Massachusets, 1998.

[15] E. Yu, *Modelling Strategic Relationships For Process Reengineering*, PHD Thesis, 1995.

[16] M. Jackson, *Software requirements and specifications*, Addison-Wesley, 1995.

[17] Dardenne, A., van Lamsweerde, A., and Fickas, S., 'Goal–directed Requirements Acquisition', *Science of Computer Programming*, **20**, 3-50, (1993).

[18] Mylopoulos, J., and Castro, J., 'Tropos: A Framework for Requirements-Driven Software Development', in Brinkkemper, J. and Solvberg, A. (eds.), *Information Systems Engineering: State of the Art and Research Themes*, Lecture Notes in Computer Science, Springer-Verlag, (to appear.)