VICTORIA UNIVERSITY OF WELLINGTON

Department of Computer Science

PO Box 600
Wellington
New Zealand

Tel: +64 4 471 5328
Fax: +64 4 495 5232
Internet: Tech.Reports@comp.vuw.ac.nz

Process Representation for Diagnosis of Crop Diseases

Peter Andreae, Philip Plimmer

Technical Report CS-TR-93/11
December 3, 1993

Abstract

The paper describes a process based approach to representating and reasoning with expert knowledge for disease diagnosis. The central contribution is a representation language that supports multi-level, multi-stage descriptions of plant and disease processes. The paper also presents a simple reasoning strategy for using the process descriptions to perform diagnosis.

Author Information

Philip Plimmer is currently working for Comtex Group, Wellington

1 Introduction

Disease diagnosis was one of the first domains to which rule-based expert system technology was applied [Buchanan and Shortliffe 1984]. Experts doing diagnosis often seem to reason at a shallow level of heuristic rules as in rule-based systems. However, experts often have difficulty in articulating the rules that they appear to use and would rather express their knowledge in a more declarative form that describes the effects of the different diseases.

In 1984, a student in our department developed a rule-based expert system for diagnosing wheat and barley diseases in NZ [Turner 1984]. The student constructed the rules from descriptions of the diseases in a fact sheet produced by NZ MAF, and then modified them in consultation with an expert. Although the expert could happily read and understand the rules, it was not the way he preferred to describe his knowledge of the plants and the diseases [McCloy 1983].

This paper describes an approach to representing crop disease knowledge in a form closer to that used by the expert, and a reasoning method that uses this knowledge more directly than by translating it into a set of rules. It reports work done for an MSc by the second author [Plimmer 1990]. The key observation is that the experts describe both the plant and diseases in terms of a sequence of phases of the life-cycle of the organisms involved. Each phase is described as a process with preconditions for being active and observable effects. The descriptions are at a “surface” level in that they describe broad generalizations relating to observable features rather than constituting “deep” causal model of the processes that would enable a simulation (either qualitative or quantitative) of the processes or provide a detailed explanation of how the processes work (see [Patil 1981] or [Iwasaki 1988] for examples of an alternative “deep model” approach). However, the descriptions may involve more detailed descriptions of a phase in terms of a collection of sub-phases. More detailed comparisons of this work with other model based diagnostic systems can be found in [Plimmer 1990].

The central contribution of this paper is to provide a representation language that supports knowledge about crop disease diagnosis in terms of processes. The paper also presents a reasoning mechanism that uses these descriptions to perform diagnosis in a similar manner to a standard rule based system. We describe an implementation of the reasoning mechanism that exploits features of the KEEWorlds system that is part of the KEE knowledge engineering environment.

2 Representation.

We first present our representation of processes in the abstract, then present the representation of crops and diseases in terms of this representation.

2.1 Facts

To represent facts about the objects in the world (whether effects or precondition of a process), we use *[property, object, value]* triples, such as [colour leaves green] to mean that the leaves of the plant are green. Objects with components, such as a wheat plant, are represented by a collection of objects related by the distinguished part-of relation: [part-of wheat-plant stalk]. Since not all possible components of a compound object are always present, (the stalk of a wheat plant is not present during the tillering stage) we represent the presence of a component using the exists property: [exists stalk] (equivalent to the triple [exists stalk true]).

2.2 Processes

The processes we are concerned with are activities involving physical objects that span some duration of time. Our aim is not to describe the internal mechanism of a process, but to describe it as a black box. We specify three aspects of the process:

- the *effects* of the process on the objects involved,
- the *preconditions* required for the process to become active,
- the *stopping conditions* under which the active process becomes inactive.

Effects of Processes.

We distinguish two kinds of effects of a process: the effects that only last while the process is active (*e.g.*, the noise resulting from the operation of a pneumatic drill) and the effects that remain after the process is completed (*e.g.*, the hole in the ground and the deafness of the operator). We refer to the former as *during effects* and to the latter as *continuing effects*. These effects are described in terms of the existence of objects (*e.g.*, [exists hole]) and the properties of the objects involved (*e.g.*, [hearing operator poor]).

These effects are not always guaranteed to occur: some effects only occur sometimes. This uncertainty may result from known external factors — the process produces the effect only when certain conditions are true external to the process. These effects are represented as *conditional effects*, where the effects are conditioned on the external factors. For example, a disease might result in yellow pustules on a plant if the plant is young, but result in orange pustules if the plant is mature. The uncertainty may also result from unknown external factors, genuine randomness in the process, or ignorance of the internal workings of the process. Such uncertainties could be represented using probabilities. For example, the disease may produce black blotches on the leaves, but only in 30% of the cases. Although we can represent these uncertainties, we have chosen to ignore all but the conditional effects in the reasoning mechanism. This is an obvious area for future work.

Preconditions of Processes.

We distinguish several kinds of conditions for a process to become active. *Enabling conditions* and *disabling conditions* must be true or false, respectively, for the process to become active. For example, it may be necessary for the temperature to reach 10 degrees for a wheat plant to start sprouting, or a particular fungus may not enter the sporulation phase if the wheat has been sprayed with a particular chemical. A particular kind of enabler that we represent specially is the completion of previous processes. For example, the wheat plant cannot enter the stemming phase until the tillering phase is completed. These *transition conditions* are discussed below.

Enablers and Disablers are necessary conditions but may not be sufficient to activate the process. If the activation of a process is uncertain, there may be Predisposing and Inhibiting conditions that increase and decrease, respectively, the likelihood of the process activation. As in the case of uncertain effects, we allow the representation of Predisposing and Inhibiting conditions, but our reasoning mechanism ignores them.

Stopping Conditions of Processes.

The conditions required for a process to become active are not necessarily the same as the conditions for the process to remain active. Therefore, we represent the stopping conditions of the process explicitly. These will frequently include the negation of some of the preconditions. However, they may also include other conditions. For example, some processes run for a fixed time. In this case, the stopping conditions will include a time limit on the duration of the process.

Relations between Processes.

Processes may be related in several ways. A process may be described at a finer level of detail by specifying a collection of subprocesses of a single process. The representation makes the hierarchical

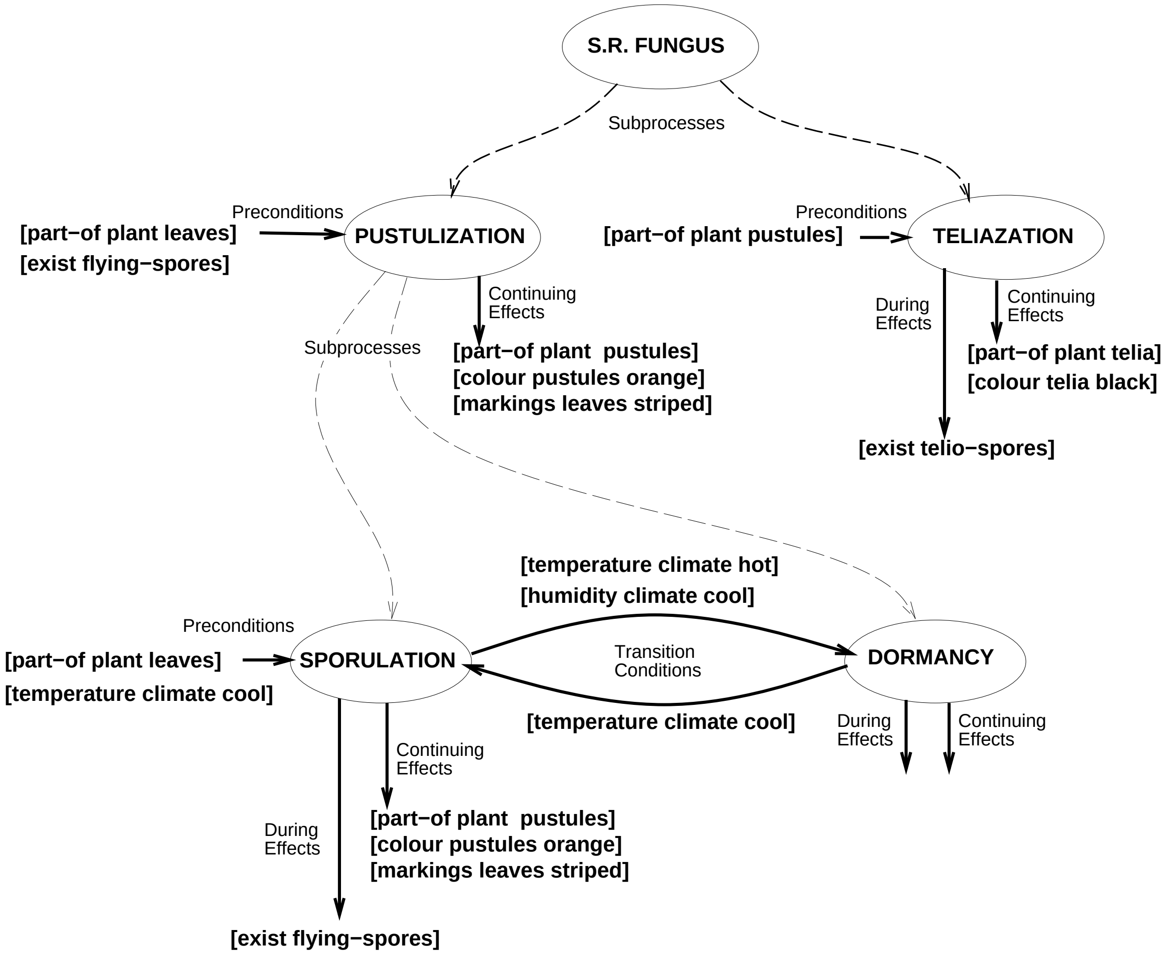

Figure 1: Process and subprocesses illustrating the representation scheme.

structure of such descriptions explicit by specifying the parent process of every subprocess. It may or may not be possible for two subprocesses of a single process to be active at the same time. If the subprocesses represent successive phases of a single parent process, then the activation of the later phase is very closely connected to the completion of the earlier phase. In particular, the stopping conditions of the earlier phase are part of the preconditions of the later phase. Rather than specify these twice, we represent them once as *transition conditions* that specify when the first phase will transition to the second phase.

Figure 1 shows the representation of a process with two subprocesses, illustrating the different components of our representation scheme.

2.3 Plants.

Although a crop disease will affect a collection of plants (*e.g.*, a field), the most important information for diagnosis involves a single plant that is taken to be typical of the collection. Therefore, the representation is focussed on a single plant.

A wheat plant grows from a single seed to a mature plant with root, stalk, leaves, and head. The experts distinguish many stages of growth. Each stage is characterised by the presence of different components (the stalk does not appear until the stemming stage) and/or different properties of the components (*e.g.*, the colour of the head may change). Each stage can be viewed a separate subprocess of the overall process of the life of a wheat plant. In some schemes, the stages are divided into shorter

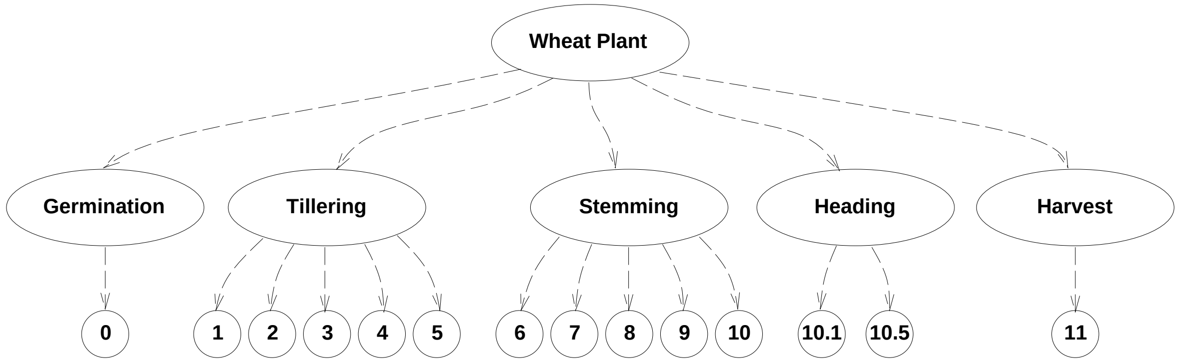

Figure 2: The Wheat plant life cycle at two levels of detail

sub-stages. We represent each stage (and sub-stage) as a process. Fig 2 shows a diagram of two versions of these stages, one being a refinement of the other, but does not show all the defining characteristics of the stages.

The individual wheat plant is not the only object that is important for crop diagnosis. We may need to represent the crop, as a collection of plants with properties such as density of planting, uniformity of maturation, average height, visual appearance, *etc.*. We may also need to represent the cultivar — the genetic variant of the plant — with properties such as disease resistance, average yield, etc. There would be no difficulty in adding either static or process descriptions of these objects, but it is sufficient for this project to focus on the single plant. For the kind of reasoning that we will be doing, we can treat important crop or cultivar properties as if they were properties of the individual plant with little loss. For other kinds of reasoning, such as planning and evaluating experiments on a variety of cultivars, this might not be sufficient.

2.4 Diseases.

Diseases are more complex than plants, at least at the level of this project. The diseases that we are considering involve a pathogen such as a virus, bacteria, or a fungi. The pathogens have a life cycle, just like a plant. For example, the stripe rust fungus has several phases: (see figure 1)

Pustulization: where the spores grow on the new plant forming pustules on the plant and stripes on the leaves;

Teliazation: where black telia appear on the plants and generate teliospores;

Sporulation: where the pustules grow and produce spores that spread to other plants;

Dormancy: where the fungus sits in the plant and does nothing.

The last two phases are actually sub-phases of the pustulization phase. Each phase can be described as a process with various preconditions and effects. For example, the dormancy phase will become active, (and then sporulation phase will stop) when the weather is hot and dry. The effect of the pustulization phase is to produce orange pustules on the plant and stripes on the leaves.

However, the pathogens cannot be described apart from the host plant that they depend on. For example, the leaves and pustules are part of the plant, not the pathogen, so that the effects of the pathogen include modifying the properties of existing parts of the plant and even creating new parts of the plant.

Unlike the plant model, it may be possible for two phases of a disease to both be active because the disease generally involves many instances of the pathogen on the same plant, which may be at different stages of the life cycle. In a model of a single instance of the pathogen, we would specify a strict sequence of subprocesses. However, in contrast to the plant where we chose to model a single plant rather than a crop, we choose to model the *collection* of instances of a pathogen that are infecting the plant and therefore we have a less specific model.

2.5 Environment

We also need to represent the environment in which the plant is growing, since factors such as the weather, season, and soil conditions all affect the plant and its diseases. The preconditions of the disease processes, in particular, will frequently refer to these environmental factors. Many of these factors are simple properties of objects in the environment and can be represented by triples. For example, the acidity of the soil can be represented by a triple such as [acidity soil 6.1]. Other factors are more complex, such as the seasonal cycle of weather, which we could model as a process with distinct subprocesses corresponding the significant seasons.

3 Reasoning Mechanism

Diagnosis in this domain involves identifying which disease, if any, a particular crop has on the basis of observable symptoms. As in most diagnosis problems, we cannot assume that the user will provide all the necessary information without being prompted, so that the reasoning mechanism must enable the system to ask the user for the results of observations during the consultation. Using the process-centered representation of the domain knowledge to do diagnosis involves attempting to fit the description of a disease to the observations.

To do this, the system must identify a disease process (or subprocesses of a disease process) that explains all the observations. Along the way, it will also need to identify the stage of growth of the wheat plant and any other active processes (such as environmental processes). Distinguishing between alternative hypotheses will drive the consultation because the system will ask the user for the values of any properties that it needs to know about to confirm or ruling out one or more hypotheses. As it finds out more facts, it will elaborate the description of the particular crop (the *specimen*), which will assist in confirming or ruling out later hypotheses.

The system uses a very simple strategy — it considers each possible disease process in turn until it finds one that fits well enough. In the process of fitting a disease, it will attempt to fit subprocesses if necessary. The order of the diseases is fixed, though it would be straightforward to extend the system to use a more intelligent strategy.

For each possible process that it considers, the system will ask the user for information about the specimen plant that might rule out the process. It needs to check several aspects of the process:

- It must check that all the effects (during and continuing) of the process of the process are present.
- It must check the presence of all continuing effects of processes that must have been active previously.
- It must check that the preconditions of the process were satisfied at some point in the past.
- It must check that the stopping conditions of the process are not satisfied.

For each fact that the system must check, it must determine whether it already knows the truth value of the fact, and if not, it must ask the user what value(s) the property involved in the fact has in the specimen. For example, if one effect of a process is to make striped markings on the leaves, then the system must ask the user what the markings on the leaves are, unless it already knows what the markings are. If the property involves a component of the the specimen plant that is not known to be present, then the system will first ascertain enough about the stage of growth of the specimen to determine if that component is present. Every fact that it discovers about the specimen should be added to its model of the specimen against to which it attempts to fit the hypothesised disease process models.

Dealing with observations that should be true in the present (the time of consultation) is straightforward; dealing with past observations is more difficult. For example, if the preconditions of the initial infection stage of some disease is that the soil was waterlogged, then it is important to know whether the soil was waterlogged at some stage in the past. However, the fact [water-level soil saturated] cannot be directly added to the model of the specimen since this may not be currently true. It is necessary, in fact, to construct a history of the specimen. This could be addressed simplistically by keeping two models of the specimen, one for the present and one of facts that were true in the past. In many cases, this simple, two-stage history will be adequate to fit the model. However, in more complex situations, where the sequence of past events is essential for diagnosis, we would need a richer model consisting of a sequence of states and what was true of the specimen in each state. This sequence of states would then be fitted to the hypothesised disease process, which would require a sophisticated matching algorithm. The most difficult problem with this approach is ascertaining from the user what time periods each of the historical facts should be fitted into and constructing a sequence of states of the appropriate level of detail. We have not addressed this problem in our implementation.

4 Implementation

Our implementation of this abstract reasoning strategy was designed to exploit some desirable features of the KEEWorlds system, part of the KEE knowledge engineering environment, which was based on an underlying ATMS system. KEEWorlds enabled us to simply and efficiently explore alternative hypotheses without any kind of backtracking and without having to explicitly check that each hypothesis was consistent with information about the specimen already obtained from the user.

Given the time limitations of a master's thesis[1] our implementation was limited in two major ways: The first is that KEEWorlds only supports boolean truth values, which prevents it from dealing with uncertainty. Although we explored the issue of uncertainty and designed an approach for dealing with it independently of KEEWorlds, the implementation did not address uncertainty in any way. The second limitation is that our implementation does not deal with time in any way, so that it does not distinguish between facts true in the present (the consultation time) and facts true in the past. Allowing the system to deal with a history of the plant as well as its current state would have required a more complex implementation that was beyond the scope of the project. Simple tricks, such as expressing some of the environmental properties as set in the past (*e.g.*, [past-water-level soil saturated]) are adequate for the simple situations that we explored.

4.1 Implementing the Abstract Representation in KEE

Facts

First, KEE provides a standard frame system that allowed us to represent the structure of a plant by a collection of frames, one for each component of the plant, connected into a part-of hierarchy. These components included all the possible objects such as pustules that are part of diseased plants. Triples are represented straightforwardly using slots in the frame of the relevant object. Constraints on the possible values of a property are represented using KEE's ValueClass and Cardinality facets. Each component frame is given a boolean valued slot exists that represents whether the component

[1]The time limitation was enforced rather drastically by the demise of the lisp machine that was our only platform for KEE. Further experiments and development of the implemented system would have required complete reimplementation not only of the diagnosis system, but also of significant fragments of KEE!

is present in the specimen. The default value of the exists slot is false for all the components that are only present on diseased plants.

In the wheat diagnosis domain, there are several logical dependencies between slot values that should be enforced. For example, there is an obvious relation between the value of the parts-affected slot of the wheat plant and the affected slots of the individual components. Logically, the parts-affected slot is unnecessary, but it enables the system to ask the user a single question about the plant rather than a sequence of questions about the individual components. There is also an obvious dependency between the parts-affected slot and the exists slots of the parts that are affected. These logical dependencies can be implemented by daemons (*ActiveValues*, in KEE terminology) on some of the slots. The first dependency was a result of a concern for making the user interaction more reasonable from the user's perspective rather than fundamental to the representation or reasoning strategy. The second dependency was a result of a limitation in the KEE frame system that does not support part-of hierarchies directly.

Processes

A *World* in KEEWorlds is a collection of facts (slot values of frames) that represent part of a possible state of affairs. The *background* consists of facts that are believed to be definitely true, and therefore must be true in all possible states of affairs.[2] Among other features, KEE will automatically (and efficiently) mark a world as inconsistent if any of the facts in that world are inconsistent with the background knowledge. We represent each process and subprocess by a world that specifies what will be true if that process is active. We represent information about the specimen by facts in the background. KEE will automatically rule out any processes that are not consistent with the the known facts about the actual crop.

KEE allows one world to be a specialisation of another world so that all the facts in the parent world are also treated as facts in the child world without having to explicitly list those facts in the child world. KEE will automatically mark as inconsistent all the child worlds of an inconsistent world. We represent subprocesses as child worlds of the world representing the parent process, which reduces the number of facts that must be explicitly stated of the child worlds.

In general, several worlds may be believed at the same time as long as they are consistent with each other. KEE can determine whether two worlds are consistent with each other. It is also possible to explicitly mark a set of worlds as inconsistent with each other, even though the facts in the worlds are consistent. This can be used to enforce a choice between alternative hypothetical situations. We could use this feature to ensure that only one phase in a sequence of phases (*e.g.*, the stages of growth of a plant) can be believed at one time. However, the system cannot exploit this feature since its reasoning strategy never actually confirms belief in a world, but only rules out worlds. Therefore, we use a different mechanism for enforcing the mutual exclusion of phases in a sequence using flags in the background (*e.g.*, the stage of growth of a plant is a property of the specimen in the background that is deduced from other facts).

4.2 Translating the Abstract Representation to KEE

The translation from the abstract process based representation described in the previous section into the worlds based representation is, in the main, straightforward and mechanical. The translation is adequate for the reasoning that the system does, but does lose some information that would be required for other kinds of reasoning (in particular, any kind of temporal reasoning, such as sophisticated matching of histories).

[2]The exists slot is necessary because of a restriction on the kind of facts that can be placed in a world — all objects and member-of relations must be in the background.

A world is created for each process and subprocess. The world of a subprocess is made to be a child world of the world of the parent process. Preconditions of a process (enablers and disablers) are stored as facts in the corresponding world, except for the special precondition of the completion of a previous process, in which case the stopping conditions of the previous process are also stored in the the world. The negation of the stopping conditions of the process are also stored in the world.

This translation is not adequate to deal with preconditions whose negation will not stop a process once it has become active (*e.g.* very wet weather may activate the pustulization phase of a fungus, but dry weather may not stop the phase once started) since this translation will insist that the precondition is true at the time of consultation rather than at an appropriate point in the past. The simplest "solution" is to represent the preconditions as past facts as discussed above.

During effects are stored in the corresponding world, and Continuing effects are stored in the corresponding world and in all worlds corresponding to succeeding phases. For every fact that involves a property of a component of an object, an additional fact is added to the world stating that the component exists.

Where the effects are conditional on some other fact, the translation is a little more complex. Essentially, we need child worlds of the process world that have the condition and the conditional effects. These worlds must be identified as representing the same process as the parent world rather than subprocesses of it.

The order in which the facts are listed in the world governs the order in which the questions will be asked of the user (as described in the following section). The only necessary constraint is that the exists facts are listed before the facts about the properties of the corresponding objects. We have determined the remaining fact order by hand. Other possible approaches are outlined below.

For every sequence of subprocesses (processes whose preconditions include the completion of a previous process), a unique property is created and given a different value in each world. For each world, it will construct a rule that asserts (in the background) the appropriate value of the property if all the facts in the world corresponding to the preconditions are ascertained to be true. For the sequence of stages of growth of the wheat plant, the experts had already named this property — the Feekes' scale has a distinct number for each stage of growth.

4.3 Implementing the Reasoning in KEE

Given this world based representation of the plant and disease processes, the system searches sequentially through a list of the worlds corresponding to the different diseases. For each world that is still consistent, the system will consider the facts in the world in order. If the truth of the fact is unknown, then the system will ask the user for the value of the property involved in the fact. All the user's answers are added to the background, and KEE rules out any hypothesis worlds that are not consistent.

The system will explore each world until either it is rule out or the user has been asked about all the facts in it. If the world is inconsistent, the system continues with the next hypothesis. If the world is consistent and has child worlds, the child worlds will be explored. When all the child worlds have been explored, the system will report the hypothesis as a possible diagnosis and either halt or continue by considering the remaining consistent hypotheses. The current system continues until all hypotheses have been considered; alternative approaches are discussed below.

Fact and Question Ordering

It would be possible to automatically construct good orderings on the facts. One approach is put early in the list facts involving the most distinguishing properties — the ones that would rule out the most number of hypotheses. This could be readily determined by an analysis of the hypotheses with facts about each property, and counting the number of different values that the properties can have in those hypotheses. With statistical information about the likelihood of the hypotheses, one could do a more sophisticated analysis that would determine the fact order that would result in the least number of questions to the user on average.

However, such “optimal” orderings may generate “unnatural” and unacceptable question sequences. In particular, the questions are likely to jump from asking a question about the disease itself to asking questions related to the stage of growth of the plant or the environmental factors. One way of reducing this problem is to require the system to recursively elaborate the worlds describing the stage of growth or the environment whenever one question Although such analysis might be useful to provide a starting point for the knowledge engineer, in practical systems, it would probably be essential to do some modifications by hand in order to create acceptable performance.

There are also other kinds of information used to control the user interface, such as a facet in which the knowledge engineer can place a hand constructed question format to override the default, system generated question. These issues are not significant for this paper.

Fact Uncertainty

When presented with a question, the user may not be able to answer with certainty. Many systems allow the user to express their confidence in their answer; this system merely allows the user to either provide and answer or state that they do not know the answer. If the user does not know the value of some property, the system must record this so that it does not ask the user again. We use a facet of each property to record the distinction between “not known because not asked”, and “not known because user does not know”.

Terminating the Search

Once the system has fully elaborated an hypothesis that is consistent with all the information from the user, it is not clear when the system should stop considering further hypotheses. Stopping immediately might result in ignoring an alternative hypothesis that was just as good an explanation of the symptoms. If the hypotheses are ordered by prior probability, then we might assume that the first consistent hypothesis would be the most likely diagnosis. However, if there is any uncertainty associated with the facts provided by the user, or if any of the effects of the processes are uncertain, then this assumption may be false. The implemented system has very restricted measures of uncertainty, and therefore could not make informed decisions about search termination. At the very least, it would need to take into account the fraction of of facts in the hypothesis world that they user could not answer — a large fraction would indicate a very uncertain hypothesis. Dealing correctly with quantitative uncertainties was beyond the scope of this project, so the system current adopts the simplistic approach of finding all possible consistent hypotheses.

5 Conclusion

We have presented a representation scheme that allows knowledge about plant and disease processed to be represented at a level of detail appropriate for disease diagnosis and in a manner that is more consistent with the way experts like to express their knowledge. The key idea is to make explicit the different phases of the life cycles of biological organisms, representing each phase as a process with preconditions, effects, and stopping conditions. We also make explicit the transition conditions between phases in a strict sequence. We have described a reasoning mechanism that does diagnosis by matching descriptions of diseases to a description of the specimen, constructing the specimen description as necessary by asking questions of the user.

We have described an implementation of part of this reasoning mechanism that exploits the features of the KEEWorlds system in KEE to produce an efficient and simple search of the hypotheses. This implementation is quite limited, but can be viewed as a “proof of concept” for this kind of representation. Future work based on this approach would need to address two issues: dealing with uncertainty and dealing with histories.

Incorporating uncertainty could be done accomplished using probababilities, but the straightforward method of ruling out hypothesis when inconsistencies are found would have to be replaced by a more sophisticated best-first search of the hypotheses, always exploring the hypotheses that have the highest likelihoods.

Dealing with histories of observations of the specimen would require a way of representing observations of the specimen at different times, perhaps as a sequence of states covering qualitative intervals, and then a more sophisticated matching mechanism that could match a history against a process description of a disease.

References

Buchanan, B. G., and Shortliffe, E.H., editors, [1984] *Rule-Based Expert Systems*, Addison-Wesley.

Iwasaki, I., [1988] *Model Based Reasoning of Device Behaviour with Causal Ordering*, PhD Thesis, Carnegie Mellon University, Pittsburg, Pensylvania.

McCloy, B.L., [1983] *1983/84 Wheat Management Package*, Technical Report,, Ministry of Agriculture and Fisheries, New Zealand.

Patil, R. S., [1981] *Causal Representation of Patient Illness for Electolyte and Acid-Based Diagnosis*, PhD Thesis, Lab for Computer Science, M.I.T., Cambridge, Massachusetts.

Plimmer, P., [1990] *Diagnostic Reasoning from a Causal Model*, MSc Thesis, Department of Computer Science, Victoria University of Wellington, New Zealand.

Turner, K. D., [1984] *A Computerised Diagnostic System for Wheat Crop Damage*, Technical Report, Department Of Computer Science, Victoria University of Wellington, New Zealand.