Le POD Mildium

Contents

Au cours du chapitre précédent (chap. 4), on a présenté la méthode de recueil de connaissances élaborée au cours de cette thèse. Ce chapitre sera plus précisément dédié à la présentation du résultat de ce recueil, à savoir le modèle formel du POD Mildium sous sa forme graphique en Statechart. Comme dans le chapitre 4, le corps du texte prend la forme d'un article scientifique (« *Working paper* »[a]) reproduit ici intégralement. Il faut souligner que certaines parties résument des éléments déjà présentés au cours des chapitres précédents.

5.1 Introduction

Citons les objectif de l'article qui suit : « L'objectif est de présenter le modèle de décision formalisé nommé POD Mildium et traduit par GrapeMilDeWS : Grapevine powdery and downy Mildews Decision Workflow System. Il s'agit de la présentation objectivée et exhaustive de cette conception PIC à base d'expertise. »

On a insisté dans les chapitres précédents sur l'intérêt de la formalisation pour communiquer. L'article ci-après met concrètement en pratique cette idée en soumettant l'intégralité du concept Mildium à la communauté PIC. Par cet exposé, on entend aussi illustrer les atouts des Statecharts, représentation graphique et sémantique formelle, pour modéliser des outils décisionnels en agriculture.

5.2 Matériel et méthode

L'article présente en section 5.4.2 les éléments théoriques nécessaires à la compréhension du modèle formel. Cette section est un condensé des développements du chapitre 2 section 2.3 et 2.6. Les principes de conception et l'architecture du modèle sont précisés en section 5.4.3.

5.3 Présentation de l'article

L'introduction de l'article replace ce travail dans son contexte (cf. chap. 1).

L'article présente dans une première partie le formalisme SED et la syntaxe des Statecharts, résumant ainsi la présentation faite au chapitre 2.

La seconde partie (section 5.4.3) expose les principes de conception de Mildium (vus au chapitre 3). Cette section introduit également les variables et les évènements avec lesquels les décisions sont prises. Elle se termine par la présentation de l'architecture du modèle : les échanges d'informations entre Mildium et l'environnement sont organisés selon trois niveaux de portée. Mildium au centre, puis le voisinage permettant la lecture et l'écriture des valeurs des variables décisionnelles, et enfin l'environnement n'échangeant avec Mildium que des ordres de traitement et des notifications d'exécution (voir fig. 5.4).

La troisième partie de l'article (section 5.4.4) précise le modèle en détail avec dans un premier temps, une vue générale du processus au cours de la saison, et dans un deuxième

---

[a] décliné le 09/10/2008 par l'un des éditeurs d'*Agricultural Systems* comme hors du domaine de la revue, sans évaluation par des juges arbitres

temps le raisonnement développé au sein de chaque étape ainsi que l'explication des différents Statecharts afférents à cette étape.

La discussion (section 5.4.5) se divise en deux parties. La première replace la démarche de conception de Mildium dans le cadre de la PIC et de la modélisation de la décision en agriculture (voir chap. 3). La seconde présente quelques arguments en faveur de la modélisation formelle pour la protection des cultures, en évoquant quelques exemples d'utilisation de ces méthodes en biologie. Cette argumentation sera développée plus en détail au cours du chapitre 7.

5.4 GrapeMilDeWS (part.1) un POD pour la protection intégrée du vignoble contre le Mildiou et l'Oïdium de la vigne

"Working paper" décliné le 09/10/2008 par l'un des éditeurs d'*Agricultural Systems* comme hors du domaine de la revue, sans évaluation par des juges arbitres .

GrapeMilDeWS (part.1) an integrated pest management (IPM) Decision Process against grapevine powdery and downy mildews

Bertrand Léger[$*] and Olivier Naud[*] Véronique Bellon-Maurel[*] Michel Clerjeau[•] Laurent Deliére[$] Philippe Cartolaro[$] Lionel Delbac[$]
[*]Cemagref - UMR ITAP - BP 5095 34196 Montpellier Cedex 5
[$]INRA - UMR Santé Végétale -BP 81 33883 Villenave d'Ornon Cedex
[•]ENITA Bordeaux - 1, cours du Général de Gaulle CS 40201 33175 Gradignan Cedex

Abstract
GrapeMilDeWS is an expert based approach for the integrated pest management (IPM) of two of the major pathogens of grapevine (*Vitis vinifera*): *Erysiphe necator* causing powdery mildew and *Plasmopara viticola* causing downy mildew. GrapeMilDeWS has been designed and experimented by a team of phytopathologists. It is presented here as a formal model in Statechart. We argue that formal modelling under the Discrete Event System paradigm (DES) is efficient to model this kind of Decision Workflow Systems. The formalism is introduced and the GrapeMilDeWS system thoroughly described. Experimental results and model validation are given in a "part.2" article.

5.4.1 Introduction

Today, vine growing represent only 3% of the land use in France. Yet, it still accounts for 20% of the national pesticides consumption (Aubertot et al., 2005).

The need to develop alternative cropping system was diagnosed as early as the 1950's. Since then, the concept of IPM evolved from these early works on integrated control. In his review Kogan (1998) counted 64 definitions of this concept. The FAO's one (FAO-UNEP, 1974) is the following:

> *Integrated Pest Control is a pest management system that, in the context of the associated environment and the population dynamics of the pest species, utilizes all suitable techniques and methods in as compatible a manner as possible and maintains the pest population at levels below those causing economic injury.*

On Grapevine, IPM against insects and mites is successfully implemented. In the case of fungal diseases, knowledge is still lacking or under construction. Recent work about biological control for downy and powdery mildews has been carried out using auxiliary mites (English-Loeb et al., 2007; Duso et al., 2005). Finding elicitors of the vine natural defences (Belhadj et al., 2006), or selecting resistant cultivars are other practical ways to develop IPM in vine.

Contribution to IPM on grapevine can also be made using the existing knowledge about the dynamics of the pathogens' development, the periods of risk (Thind et al., 2004), the known resistances against fungicides and the proper phytopharmaceutical product management (Matasci et al., 2008; Waard et al., 1993) as well as by technical expertise spanning from product choices that loosen the risks of resistance appearance to prophylactic measures, early symptoms sightings and "epidemics trend inference". This approach which proposes solutions to reduce the number of treatments, through observations, thresholds or risk models, can be illustrated by (Oliva et al., 1999; Hoffman et al., 2004).

Since 2001, The INRA santé végétale (plant health) laboratory has undertaken the design of pest management decision rules, based on observation and expertise, that come as close to IPM as can be. The target diseases were: gray mold (*Botrytis cinerea*), "insect" pests (*Scaphoïdeus titanus*, *Lobesia botrana*, *Empoasca vitis*, *Panonychus ulmi*), downy mildew (*Plasmopara viticola*) and powdery mildew (*Erysiphe necator*).

Up to now, the chosen scale of these decision rules has been the plot and the decision process for pesticide application is made individually for each pest. Although being a necessary step, this approach is not adapted to an IPM decision support system (DSS) at the farm level, where the growers is keen to have multiple diseases treated in a single application. The practice of coupling treatments is particularly widespread with powdery and downy mildew treatments. In France, these two diseases represent 80% of the treatments applied on grapevine (ASK, 2000).

The first part of the work presented here was therefore to move from a one-plot/one-pest approach to a more pragmatic approach that pairs treatments against a couple of diseases, i.e. powdery mildew and downy mildew. The aim is to be more compatible with common practices.

Our ultimate goal is to transfer an operational DSS that permits to significantly reduce the number of fungicides and yet guaranty that the production targets (both qualitative and quantitative) are reached.

However before designing a DSS, it was necessary to design, formalise and evaluate prescriptive crop protection decision strategies.

We abandon here the term “decision rule” in favour of the concept of *crop protection decision workflow system* (CPDeWS), which better accounts for the sequential and integrative structure of the crop protection decision system we developed. On the concept of workflow please refer to (van der Aalst and van Hee, 2002). Our work is grounded on the French agronomic tradition, which has developed the concept of “general model” since the late 1980’s to account for the way farmers take their decisions and manage their farms. Traditionally, this qualitative framework is targeted for diagnosis and accounts for the fact that decisions modify both the production system through “technical itineraries” (Sebillotte, 1978) and the farmers representations through model for action (Sebillote and Soler, 1988).

Indeed, we designed a decision system that organizes the collection of information, the decision making, and the treatment applications in time. We acknowledge that the tactical decision rules that may trigger a treatment action should be adapted during the season. Our CPDeWS models a process, beginning at bud break in spring up until harvest. The decisions are influenced by the decisions taken earlier, the phenological development of the plot and the evolution of the crop’s sensitivity to each pathogen. Finite state automata (FSA) (Black, 2008), under the discrete event systems paradigm, is a formalism that is well adapted to modelling our time/season dependant system. With FSA, the input situation of the pathosystem can explicitly be linked to the required decisions.

The originality of our model is that it emphasizes the sequentiality and temporality of the decisions. This approach differs from rule-based expert systems (ES), which do not focus on temporality (Travis and Latin, 1991; Shaffer and Brodahl, 1998). The hypothesis supported by (Girard and Hubert, 1999) is that emphasizing temporality forces the experts to give an exhaustive specification of their crop protection program.

The complexity of designing a multi-target with evolving priorities CPDeWS required a formal representation. This formal model should be both understandable by phytopathologists other than its designers and suited for computer simulations. Indeed computer simulation of crop protection seasons under climatic and epidemic scenarios, is thought to help design and test new cropping systems (Sebillote, 1987a). We chose the Statechart (Harel, 1987) formalism for creating this model.

The purpose of this paper is therefore to present the formalized decision model named GrapeMilDeWS, for grapevine protection against downy mildew and powdery mildew decision workflow system. It provides an exhaustive and explicit description of our expertise-based IPM design.

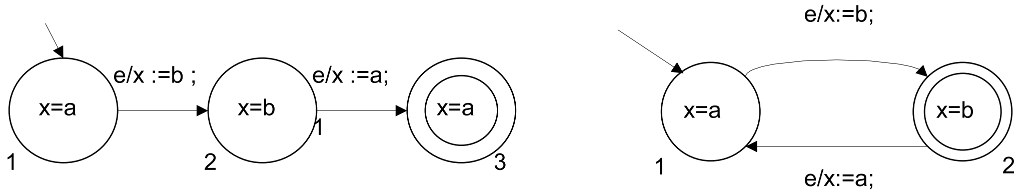
The first part of this paper consists in a description of the formalism used for the model. In the second part, the CPDeWS, named GrapeMilDeWS is presented in details. Extensive comments and explanations are made so as to allow understanding the contents of GrapeMilDeWS without *a priori* knowledge of the Statechart language. The assessment of the GrapeMilDeWS model decisions in comparison with the experiment ones and field agronomical performances is the subject of a second paper (Léger et al., 2008b).

5.4.2 Theoretical introduction to Statechart formal modelling

In this section, we introduce the formalism of Statechart and explain why it was chosen.

5.4.2.1 The Choice of Discrete Event Systems

The crop protection's decision system is modelled as a flow of decision leading to work operations. Our aim is to represent the temporal dimension of the CPDeWS, for the whole growing season. The continuous dynamics of phenology and of epidemics can be represented at the plot scale by differential equations. However, we have chosen to model the CPDeWS as a Discrete Event System (DES). Indeed the IPM experts take decisions based on thresholds defined on the epidemics and the phenological stage variables. The variables are therefore discretised according to the thresholds. The decisions are thus made according to these discrete values and the crossing of a given threshold constitutes an event. The combination of the epidemics and phenology variables compose a finite set of values for the input vector of the CPDeWS, together with a set of external events, such as rain forecast. Decisions, like "evaluate the diseases level" or "order a treatment" are output events of this system.



(a) "$x = a \ \wedge \ once \ x = b$" is true if state3 (b) "$x = a \ \wedge \ once \ x = b$" is never certain

Figure 5.1: State diagram : used to check a system's property

Among DES formalisms, we chose the diagrammatic language of Statechart. As they are depicted by readable graphs, statecharts are relevant mediation tools between the phytopathologist designers and the knowledge-management researchers for eliciting the formal model (Léger and Naud, 2007). Computationally speaking, Statechart can be assimilated to FSA. The later are presented in the next section.

5.4.2.2 Finite state automaton

FSA are mathematically depicted by directed graphs where nodes are states (for instance, actions or observations in our case) and edges are transitions (Figure 5.1). Transitions are labelled with events. They may also bear a "guard condition". From the active state, a given transition can only be taken at the occurrence of the event specified by its label, if the guard, when present on the label, is evaluated to "true".

Consider now a system which holds track of the evolution of variable x. The "tracker" is modelled using a FSA. Here, labels mapping x's values have been added to the states. The event label is attached to each change of variable x. While taking any transition labelled with "e", the value of x is updated as stated by $/x := aNewValue$. The slash sign '/' indicates that an atomic action is carried out during the transition.

In Figure 5.1(a), state 3 is different from state 1 even though they both record the same property of the system: $x = a$. However, state 3 also holds the information that the system has been in state 2 at one point. This is where a modeller can choose the behaviour he needs to represent. If the monitoring of behaviour *"$x = a$ and once $x = b$ has been true"* is not relevant in the problem to solve, then the modeller can choose the simpler automaton in Figure 5.1(b).

Note that these two automata are not semantically equivalent: in Figure (a), the automaton can only accept one change of x from a to b, whereas the automaton represented in Figure (b) accepts an infinite number of changes.

We use FSA to monitor relevant phenomena during the crop protection season which we label and once they are identified, we act upon the system. For us the states are more than the combination of all input variables. They depict our output decision. At different time, similar input values may be repeated, but the state and the property associated to it will depend on the foregoing sequence of states that were reached. The combination of FSA with variable management and the possibility to label states so as to describe desired properties and generate actions accordingly, are called *State Diagrams* (Booth, 1967).

Yet, State Diagrams have a major draw back: the number of states becomes unmanageable. As soon as concurrent processes are modelled, the number of states is the

Cartesian product of each independent process's number of states. In our case, that combinatorial problem ("state explosion") occurs as soon as a rule in the model holds for the whole duration of the crop protection season. For example, monitoring the status of a product active period (AP) [b]is relevant during the whole season, although it is not relevant to all decisions. Statechart offers solutions (hierarchy and concurrency) to avoid the combinatorial explosion.

5.4.2.3 Statechart

The next section introduces the Statechart formalism. First introduced by Harel (1987), Statechart differs from the standard finite state automata formalism, by its following properties: in Harel's words:

Statechart=state-diagrams + depth + orthogonality + broadcast-communication

- Statechart allows depth: a hierarchical view of the system, each state can be composed of a substatechart.
- Orthogonality or parallelism, enhances the semantics, permitting to describe concurrent processes on the same chart.
- Finally, Statechart features broadcast communications. In a broadcast communication system, an event is available to every concurrent process simultaneously.

After this short introduction, the syntax and some semantic elements are presented. For accessible yet more complete presentation of Statechart refer to (Harel, 1987; Harel and Kugler, 2004). With its integration as a part of the Unified Modelling Language (UML 2.0) (OMG, 2007), Statechart is now supporting object oriented design. Under the Object Oriented modelling paradigm, broadcast communication has been restricted. The popularity of Statechart led to the design of many flavours. Comprehensive comparisons are presented in (von der Beeck, 1994; Maggiolo-Schettini et al., 2003).

Our implementation of the model is done using the Rhapsody software by Telelogic (Harel and Gery, 1996).

5.4.2.4 The graphical syntax of the Statechart

Reading tip: the words in capital are key concepts which are explained later in the section.

States Harel introduces 4 kinds of states in Statechart (see Figure 5.2). The *simple states* 5.2(a) which are close equivalent to the FSA states ; *the final states* 5.2(b) are the acceptor states in FSA and represent the completion of a Statechart or its substatechart.

Hierarchy (i.e. substatechart) is made possible by: the *"OR state"* 5.2(c) which includes exclusive substates inside a parent "OR state" and the *"AND state"* 5.2(d) which allows concurrent processes to run simultaneously. The concurrent processes of the AND-State are graphically divided by dashed lines.

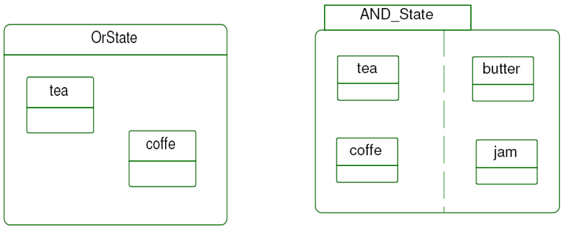
Entry and exit ACTIONS can be executed when the state activates or when it deactivates.

---

[b] The active period is the period during which the product efficiency is "guarantied", and after which we consider the plot has become susceptible again

(a) a simple state

(b) a final state marks completion of a process



(c) an OR-state contains mutually exclusive substates

(d) an AND-state contains simultaneous processes

Figure 5.2: Statechart's different kind of states

Transitions connect a set of origin states to a set of destination states. Events, guards and actions compose the label of a transition. A transition label is structured as follow:

$$evAnEvent[aGuard]/anAction$$

guards are denoted between brackets '[' *guard* ']' and actions are preceded by the slash sign '/'. In GrapeMilDeWS, all events are identified with the prefix '*ev*'. A transition is potentialized if its origin states are active. It is triggered by the event specified on its label and on condition the guard is "true". While the transition is taken, an action may be executed.

Each component of the label is optional. A transition with no triggering event is called a nul transition and is taken "as soon" as its origin states becomes active[c], provided the guard is true. Usually, transitions are instantaneous (Maggiolo-Schettini et al., 2003).

Events are instantaneous messages originating from the Statechart or from external sources. The occurrence of an event triggers the transitions referencing the event on its label, provided the transition is potentialized.

Guards are boolean conditions that control if a potentialized transition can be taken.

Actions are pieces of algorithm that modify the internal values of the system, for example: event generation or variable assignment. Actions may be executed during a transition or upon entry or exit of a state.

Readers interested in the formal definition of the UML Statechart semantic (that we use) should refer to (Damm et al., 2003).

Pseudostates are graphical symbols that have no transcription in formal semantics. They are: the initial states (also known as default transitions), the condition nodes, the

---

[c] This is Rhapsody object Statechart semantics, it requires the object which behaviour is described by the Statechart to have the focus. Focus while be given when a triggering event allows a transition to be taken from a stable configuration. Focus is lost when a stable configuration is reached, i.e. no more transitions can be taken)

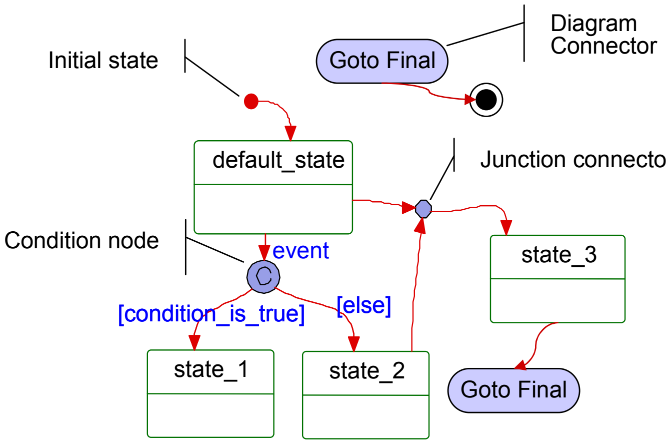


Figure 5.3: Pseudo-states (initial state, condition, junction and fork as well as "diagram connectors") have no mathematical existence but are graphically useful.

fork and junction (example in Figure 5.3). The diagram connectors are not part of the original Statechart syntax, but come handy to jump from one side of the diagram to the other. They help avoid cluttering the Statechart.

5.4.3 Principles and hypothesis for the design of GrapeMilDeWS decision workflow

Having established why the system at hand is modelled as a DES and having introduced the formalism, we present the crop protection principles of GrapeMilDeWS and then how the formal model was designed.

5.4.3.1 Crop protection design choices

GrapeMilDeWS aims at avoiding yield losses, not at avoiding disease symptoms. This is achieved (i) by controlling low epidemics (i.e. maintaining it at a low level) with a reduced number of systematic treatments applied at key phenological stages (2 mandatory treatments against downy mildew and 2 against powdery mildew), and (ii) by identifying the severe epidemics as early as possible, in order to apply additional treatments (5 optional sprayings are available against downy mildew and 3 extra treatments may be done against powdery mildew).

Adapting the number and the timing of the fungicides application to the plots' specific epidemic conditions is achieved through intensive use of various data sources, mostly from the plot itself.

When a treatment is required for a disease, the other will be dealt during the same application unless the risk in the plot (or in the area) is judged nil or low. This rule allows us to couple the treatment against powdery and downy mildews as often as possible. This heuristic simplifies the management of treatments against multiple pathogens which otherwise would impose strong operational constraints on the grower.

Still to alleviate the work load, GrapeMilDeWS is constrained w.r.t. the number of disease level evaluations in the plot. All plot observations lead to one or more treatment decision. Three field observations are done before flowering (one of them is optional), a third mandatory observation is done a month after flowering.

In a pragmatic approach, no treatment reduction is attempted during the period of highest susceptibility, the crop is systematically protected at the flowering and there is no need to estimate the level of infestation.

5.4.3.2 Observations and information generation

The treatment decisions are mostly made based on epidemic estimates, at the plot scale. These estimates are interpreted from sampled observations on the leaves as well as the bunches. The observation results are then translated into the three following discrete variables:

- O standing for the level of powdery mildew on the leaves (O for Oïdium: powdery mildew in French)
- Og standing for powdery mildew on the bunches (Og for Oïdium grappes: bunches powdery mildew)
- M for downy mildew on the leaves (M for Mildiou: downy mildew)

The number of modalities varies from 2 to 3 levels depending on the disease, and the observation date. These modalities encode the qualitative expert assessment as follows: ('0') for absence or low epidemic; ('+') for moderate to high epidemic; and ('++') for very high epidemic risk. The threshold values between these different modalities evolve with the phenology of the vine. This allows to adjust the consequences of an epidemic level to the evolution of the plant susceptibility during its development.

Field observations are the only information used as far as powdery mildew treatment decision is concerned. Two extra indicators are used for the decision making with respect to downy mildew epidemics:

- The local area risk level (ILM) gives information on the disease development risks at a geographical scale larger than the plot. It is based on a large disease monitoring network and on a climatic risk model. ILM is interpreted from the plant protection service advisory bulletins [d](SRPV-Aquitaine, 2007). It is encoded as a discrete variable, with two modalities: ('0') low risk and ('+') medium to high risk.
- The forecasted rain events from the MeteoFrance weather forecast service.

The variables (M, O and ILM) are built with thresholds which are modified during the season [d]. This has the effect of embedding some expertise on the dynamics and the dangerousness of the epidemics, into the three estimators. This provides GrapeMilDeWS and the end user, with data which are more easily interpreted.

5.4.3.3 Model's architecture

To fully understand how the system works and especially how information is managed, it is necessary to carefully describe the interactions of GrapeMilDeWS with the vineyard and the data flows between them. As shown on Figure 5.4, information generation and exchange can be organized in three scopes according to the information access rights.

The first scope is the environment of the decision system. Concretely, the environment is the vineyard plot with its phenology, its epidemics, as well as the weather forecasts and the local area epidemic pressure around the vineyard. The communication between GrapeMilDeWS and its environment are limited to exchanging event messages.

---

[d] see implementation details in GrapeMilDeWS part.2 (Léger et al., 2008b)

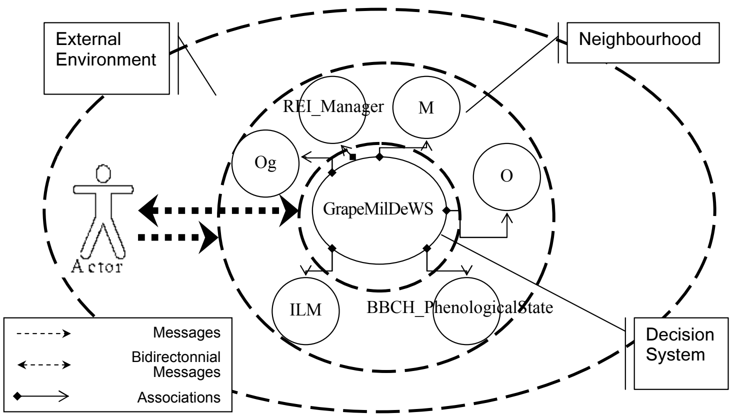


Figure 5.4: 3 Scopes are defines : the system, the neighbourhood, the environment

As events are not persistent information, a part of the communications are routed to the neighbourhood variables to make it perennial, the rest is interpreted directly by GrapeMilDeWS.

The second scope called the neighbourhood, is composed of the three field observations aggregated variables: O, Og and M (presented section 5.4.3.2 monitoring powdery and downy mildew epidemics, as well as the local downy mildew information (ILM), the phenological stage and the restricted entry interval manager[e](REI_Manager). They are modelled as associated objects to the GrapeMilDeWS system.

These five objects are GrapeMilDeWS' memory of the environment's status. They can exchange events with GrapeMilDeWS. For instance, the object "Pheno" which keeps track of the phenological state monitoring, sends a notification event each time the external environment (i.e. the actor in the first scope) updates its value. GrapeMilDeWS can also read the current state values of these variables whenever needed. The model is designed using the object oriented approach which has the advantage of built-in modularity. However the neighbourhood variable could be managed otherwise.

The third scope is the GrapeMilDeWS Statechart itself, inside of which the system's control over the data is total.

data flow The communications between the GrapeMilDeWS and the external environment is constrained by the boundaries of the different scopes. The environment is not directly observable. It is required that some actors run processes in that environment, which produce messages between the environment's continuous behaviour and GrapeMilDeWS. The main actor in the environment is actually the vine grower running GrapeMilDeWS. The processes are either permanent monitoring processes (phenology, weather forecast and local downy mildew risk) emitting status update information, or reactions to queries from GrapeMilDeWS (observation requests, treatment orders).

This architecture permits to build an asynchronous system that models well the reality of decision making in crop protection.

---

e. Restricted entry intervals (REI) are required by the French legislation on pesticides: depending on toxicity, access is forbidden from 1 to 3 days after an application.

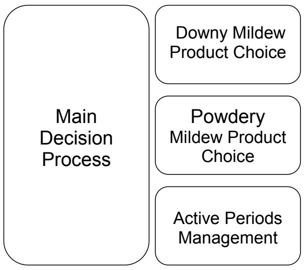


Figure 5.5: Four concurrent (simultaneous) functions compose GrapeMilDeWS

5.4.4 GrapeMilDeWS detailed presentation

5.4.4.1 GrapeMilDeWS Statechart's structure

GrapeMilDeWS is composed of four independent functional processes which run simultaneously (see Figure 5.5). Implemented in Statechart, these four functions are modelled as high level AND-States (see Figure 5.6). Along with the main process (section 5.4.4.2), are two product choice rules, one for each target disease (sections 5.4.4.11 and 5.4.4.12). The remaining AND-State is used to manage the active periods (AP) of the last treatment against each disease (section 5.4.4.3).

In the following, we start with the top level view of the main process. It represents the general organisation of the sequence of tactical decisions and the constraints controlling their timing. Then we will clarify the AP management key concepts. These preliminary given we will be able to detail the seven treatment decision stages. Finally we conclude with the two phytosanitary product selection rules.

5.4.4.2 Main process overview

GrapeMilDeWS' top level Statechart in Figure 5.6, abstracts from the details of the decision making which are hidden in the stages' substatecharts.

In the main process, each of the seven treatment decision stage state contains the intrinsic logic for a potential treatment against powdery mildew, downy mildew, or both, in the form of a substatechart (sections 5.4.4.4 to 5.4.4.10). In the following sections, we will refer to "treatment decision stage states" as "treatment stages" or just "stages". We will often use the following notation "Tx" when referring to a treatment ordered at `Stage_`x (i.e. from "T0" at `Stage_0` to "T6" at `Stage_6`). When referring to the treatment target is necessary, the variable name may also be added. For instance "T1O" stands for the treatment targeting powdery mildew (the O variable) at `Stage_1`.

At three key periods of the crop protection, treatment stages are interlaced with observation states. The strategy is built around securing the flowering period. Three treatment stages are positioned before flowering to control the early epidemics on the leaves as well as on the inflorescences and three post flowering treatment stages control the development of the diseases on the bunches and leaves.

The season starts with a monitoring as the first leaves unfold ([BBCH>10] tag ① in Figure 5.6). References to phenological stages in the diagram are given in the BBCH

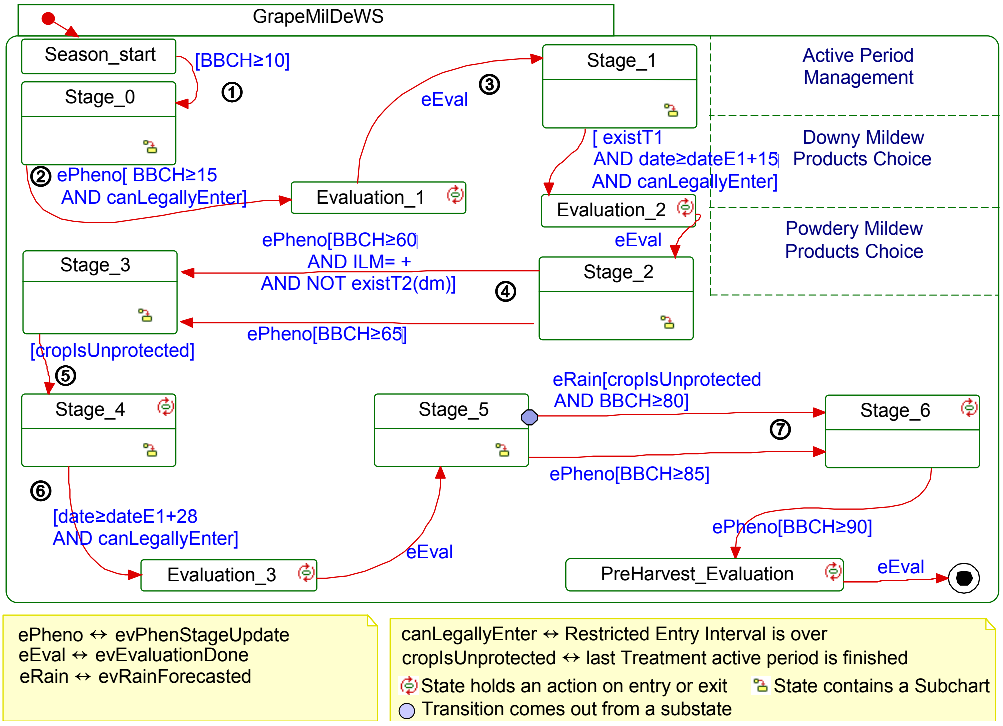


Figure 5.6: GrapeMilDeWS main process: 7 treatment decision stages and 3 observation states.

scale (Lorenz et al., 1995). The system remains in `Stage_0` (details in 5.4.4.4) until the phenology of the plot has developed to at least 5 leaves unfolded (② Figure 5.6). Downy mildew treatment is optional at that stage. `Stage_0`'s early monitoring is designed to control the extremely precocious downy mildew epidemic. If a treatment occurs during `Stage_0`, it is legally required that the restricted entry interval (REI) be elapsed before anyone enters the plot to perform the evaluation requested in GrapeMilDeWS' state `Evaluation_1` (the plants must also have developed 5 leaves). The REI test has been encapsulated in the boolean function *canLegallyEnter* shown at ② Figure 5.6.

GrapeMilDeWS will remain in `Evaluation_1` until the observation of the plot has been carried out and notified (evEvaluationDone) and the neighbourhood variables O and M (assessing both foliar epidemic levels) have been updated. (③ Figure 5.6) After that, `Stage_1` is entered.

`Stage_1` lasts two weeks after `Evaluation_1` during which carrying out a powdery mildew treatment is required. An optional downy mildew treatment may also be decided according to the epidemic estimators (M and ILM). The temporal positioning of the treatments during `Stage_1` is detailed in section 5.4.4.5. `Stage_1`, is aimed to last from '5/7' unfolded leaves to '8/10' leaves. Phenology being quite difficult to determine precisely, the designers have chosen to used a fixed time period of 15 days instead. They consider 15 days to roughly correspond to such a phenological development in the Bordeaux area where GrapeMilDeWS was experimented (see part.2 (Léger et al., 2008b)). At the end of `Stage_1` the second evaluation is ordered, provided the plot can be safely entered (i.e. the REI resulting from the first downy mildew application has elapsed).

`Evaluation_2` targets the same organs as `Evaluation_1` did. Upon completion of these observations, `Stage_2` has information on whether the first stage has efficiently managed to controlled the beginning of each epidemic or, if new symptoms are still surging. If the epidemic level of any the two diseases is worrying, `Stage_2` calls for an extra treatment in order to safely reach mid flowering. The precise decision logic of the transitions from `Stage_2` to `Stage_3` is quite complex and in-depth details are given in section 5.4.4.6. In our Bordeaux conditions the typical duration of `Stage_2` is again approximately two weeks.

The objectives of the two early observations at `Evaluation_1` ("E1") and at `Evaluation_2` ("E2") are to detect the severe epidemics by quantifying the early symptoms of the diseases on the foliage, before the period of high susceptibility of the bunches. This early detection mechanism allows us, when required, to apply treatments limiting the proliferation of the inoculum on the foliage thus "breaking" the epidemics before if reaches the explosive phase (under the Vanderplanck theory Segarra et al., 2001).

`Stage_3` (④ Figure 5.6) (see section 5.4.4.7) can be entered either at early flowering or at mid flowering. Depending on the decisions taken during `Stage_2` (if no treatment against downy mildew was ordered at `Stage_2` (T2M) the third treatment (T3) is done early, otherwise the plot is protected until mid flowering thus `Stage_3` is entered at the end of T2's AP). `Stage_3` simply triggers the third treatment: T3. This is *the key mandatory treatment* in the GrapeMilDeWS program. It targets both powdery and downy mildews.

`Stage_3` (⑤ Figure 5.6) ends when the shortest active period (AP) of the 2 product used for "T3", has elapsed (i.e. on Figure 5.6, the function *cropIsUnprotected* becomes true). At that time, the berries are at pea size. There is no evaluation of the epidemics in the field between `Stage_3`'s exit and `Stage_4`'s entrance.

No mandatory treatment is required at that stage. Any spraying that may be ordered in `Stage_4` is based on the values of O and M recorded during the first two evaluations (details in section 5.4.4.8). `Stage_4` is designed to give extra security only in the cases of high epidemic pressure, these disease scenarios are detected during "E1" and "E2".

`Evaluation_3` (⑥ Figure 5.6) is ordered 28 days after "T3", provided REI has elapsed after the optional "T4". `Evaluation_3` differs from the two previous evaluations. It monitors powdery mildew on the bunches and downy mildew on the leaves. It provides an early estimate of the sanitary status of the grape (before harvest) and support to decide on the opportunity for one more optional treatment.

At this time in the season, the bunches are beginning to close. The sprayings that may be ordered during `Stage_5` are based solely on the indication acquired during `Evaluation_3` (see section 5.4.4.9).

`Stage_6` (⑦ Figure 5.6) consists in a final mandatory treatment against downy mildew, positioned during the first half of ripening (see section 5.4.4.10). The bunches are no longer susceptible to neither powdery nor downy mildews, but the aging leaves can be destroyed by downy mildew. Therefore T6 is applied to ensure the stocks have enough foliage for the maturation of the grapes. When the grape is ripe, a `Pre_harvest_evaluation` is ordered to control the overall quality of the crop protection. This assessment leads to no spraying decisions and may be discarded in a production context.

5.4.4.3 Active period management

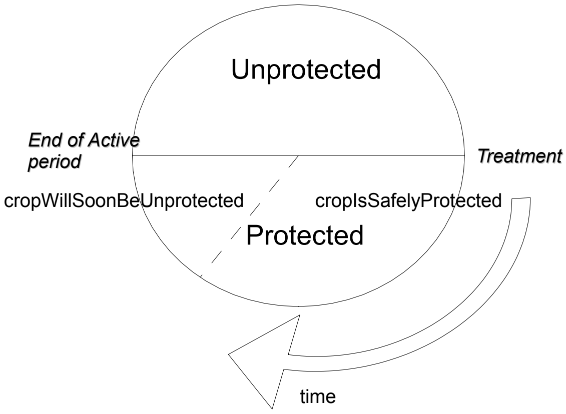


Figure 5.7: GrapeMilDeWS active period management is summarised by 4 States

The active period management consists of two symmetrical concurrent processes, used to keep track of the AP, one for each disease. As these statecharts are purely technical, they are not shown here, we rather provide a conceptual view of the AP management mechanism as shown in Figure 5.7.

When selecting a product, the active period duration is set. Once the treatment is done, A timer will keep the AP manager in the "Protected" state for the duration of the selected product's AP (Figure 5.7). The "Protected" state has 2 substates : "crop is safely protected" and "crop will soon be unprotected". Indeed the protection is considered safe until the delay before the end of the AP is less than EARLY_RENEWAL_INTERVAL (an expert parameter, we modelled as a constant value). Indeed, when the epidemic risks are high, the plant protection service often recommends to renew treatments a few days before the end of the active period. Typically the AP may be shortened by 2 to 3 days. After the end of the AP, the "Unprotected" state is activated. With these 4 States and the variable for the AP duration, we can model the protection provided by the treatment and ameliorate the positioning of the applications.

The active period protection can be queried using a set of boolean functions that map the set inclusions shown in Figure 5.7. Parameter *dm* allows to test only downy mildew protection, *pm* does the same for powdery mildew and no parameter tests both at the same time.

Following this overview of the crop protection process, the internal logics and sequentiality of each of the seven stages is described.

5.4.4.4 Stage 0

The substatechart of `Stage_0` is shown in Figure 5.8. All tags in the following section refer to Figure 5.8.

This early monitoring stage starts as soon as the first leaves unfolded (BBCH>10). During `Stage_0`, the driving neighbourhood variable is ILM. It is updated each time the risk of downy mildew in the area changes. Early in the season up to flowering, ILM will

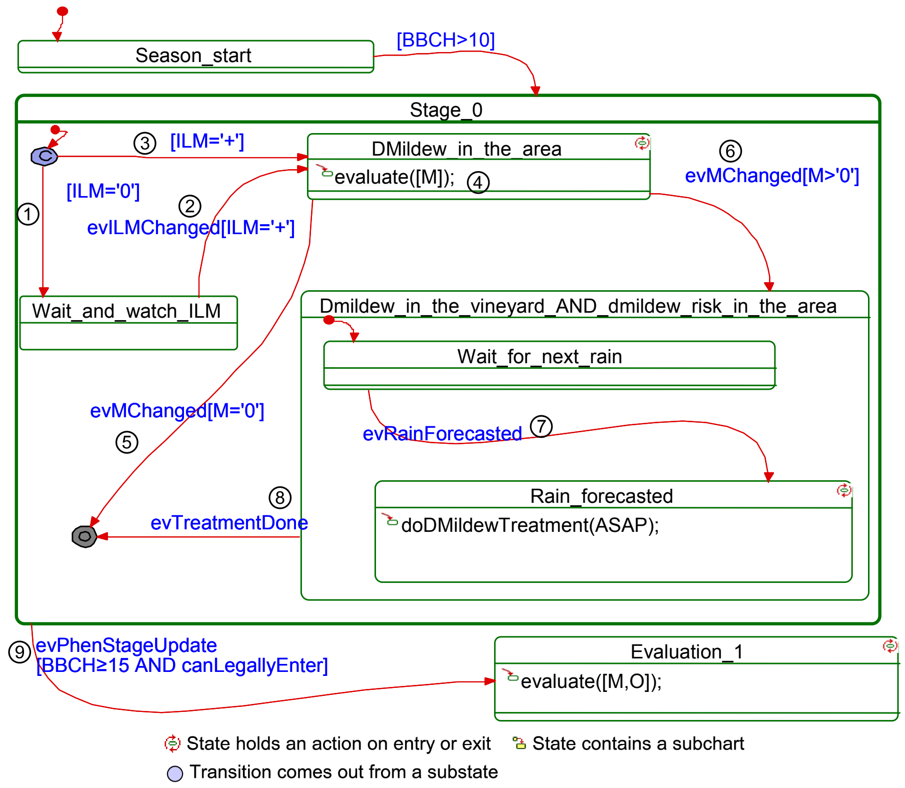


Figure 5.8: GrapeMilDeWS Stage_0

be set to '+' as soon as the first symptoms of downy mildew are found nearby the plot. This information is taken from the plant protection service advisory bulletins within the range of 10 to 25 km around the vineyard.

① If [ILM='0'] when `Stage_0` is entered, the `Wait_and_watch_ILM` substate is activated. It will remain so as long as ILM does not change. ② When ILM changes to '+', the transition towards `DMildew_in_the_area` is taken.

State `DMildew_in_the_area` can be activated when ILM is updated or if its value is '+' when `Stage_0` is entered ③ . Entry in state `DMildew_in_the_area` generates an order to evaluate downy mildew in the plot (④ *evaluate*([*M*])). Completing that evaluation will update the M variable. If no downy mildew is found, the final state is reached ⑤ and no further action is taken within `Stage_0`. Otherwise `DMildew_in_the_vineyard_AND_dmildew_risk_in_the_area` (for short: `S0.DVDRA`) is activated ⑥ .

The Substates composing `S0.DVDRA`, represent the behaviour that is generally applied for downy mildew management in GrapeMilDeWS. (i) First, the weather forecast watch is ordered upon entry into the `Wait_for_next_rain` state. (ii) Then, ⑦ when a rain is forecasted, the transition is taken, state `Rain_forecasted` is activated and treatment "T0" is ordered. Once the treatment has been done, the information is returned to GrapeMilDeWS with the event evTreatmentDone which triggers the transition between `Rain_forecasted` and the `Stage_0`'s final state⑧ . The final state indicates that no further activity will be carried out by the system while it remains in `Stage_0`.

When the field has 5 leaves unfolded, the phenological stage monitoring variable is updated. The update is notified to GrapeMilDeWS through the evPhenStageUpdate event. That event sets `Stage_0` to inactive, whatever its active inner substate ⑨ . However, the outgoing transition cannot be fired solely by the update event: the guard is composed of two mandatory conditions: 5 leaves must have unfolded ([$BBCH \geq 15$) and the plot can be entered (*canLegallyEnter*]) (i.e. the REI must have elapsed). Entry in `Evaluation_1`, generates the order for a field evaluation on the leaves of both powdery and downy mildew symptoms level.

5.4.4.5 Stage 1

The substatechart of `Stage_1` is shown in Figure 5.9. All tags in the following section will refer to Figure 5.9.

Once the first mandatory observation of powdery and downy mildew in the field has been carried out, `Evaluation_1` is exited and `Stage_1` entered. The variables M and O are then up to date. Entering `stage_1`, (O, M and ILM) are used to select a state in the decision path according to the sanitary status of the plot. A mandatory treatment against powdery mildew is done within each possible path. It aims at breaking the dynamics of the epidemic very early in the season. This mandatory treatment may be positioned differently within the 15 day period of `Stage_1`, based on the results of the observations made during `Evaluation_1`. An optional downy mildew treatment may be added when required by the epidemic conditions.

The initial state (default transition) of `Stage_1` leads to a first conditional node. This node has two branches separating the modalities of the powdery mildew variable O with the labels [$O =$'++'] and [$O <$'++']. We will structure the detailed presentation of `Stage_1` along this first conditional node as `Stage_1`'s decision making is driven by powdery mildew.

O <'++', low powdery mildew: The position of the spraying will then be driven by downy mildew through the second decision node ① . It, has 3 branches, one for each modality of the M variable.

If M='++', The high level of downy mildew requires immediate action. Therefore on entry of state `High_dmildew_in_the_vineyard`, mixed treatment of both powdery and downy mildews is ordered as soon as possible. Although powdery mildew is low, it is treated immediately so that only one spraying is done which makes the work easier to organise and is probably better for the operator's health and the environment.

`High_dmildew_in_the_vineyard` remains active until notification of the treatment (evTreatmentDone) triggers the outgoing transition to the final state.

If no downy mildew was found (M='0') then it is ILM which drives the decision. If no symptom has yet been found in the region (i.e. ILM='0'), `Wait_and_watch_ILM` is activated. As in `Stage_0`, the monitoring of ILM continues until either, (i) the 15 day period of `Stage_1` will *soon* be elapsed (transition labelled [$soon(DateE1 + 15)$]②[f]); or (ii) the ILM value changes. If ILM is set to ILM='+' then `Dmildew_risk_in_the_area_OR_dmildew_in_the_vineyard` (for short: `S1.DRADV`) is entered. This state is the start of a weather watch procedure very similar to the one in `Stage_0` (section 5.4.4.4). Details of the substatechart of `S1.DRADV` are given in Figure 5.10(a). If rain is forecasted, the

---

[f] In our simulator implementation of GrapeMilDeWS, the *soon*(*adate*) function returns a date 3 days earlier than *adate*.

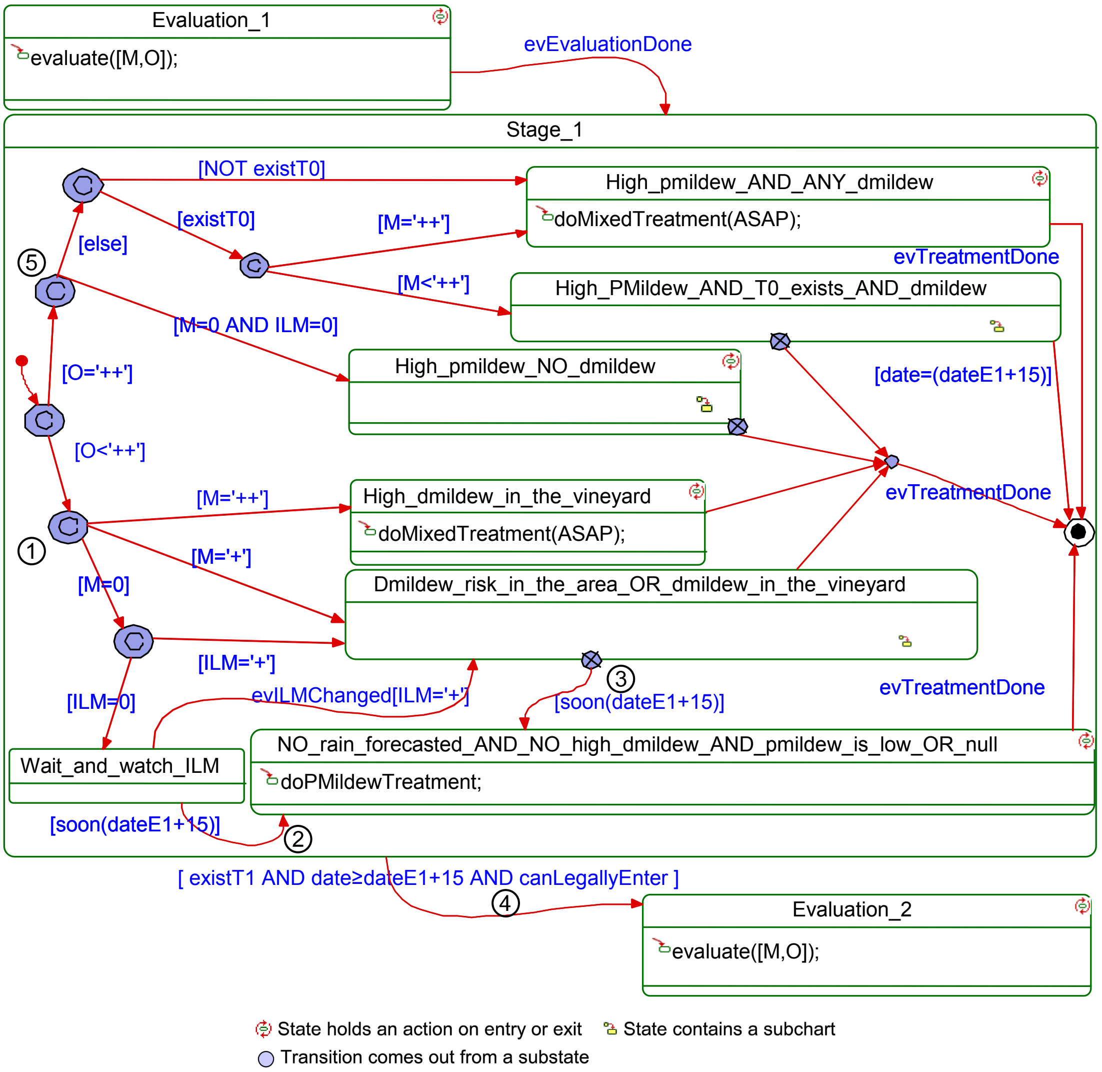


Figure 5.9: GrapeMilDeWS Stage_1

notification that the mixed application has been done triggers the transition to `Stage_1`'s final state(see Figure 5.9).

Alternatively (see ③ Figure 5.9), if no rain is forecasted and if the 15 day period is almost finished, then the transition leading to `NO_rain_forecasted_AND_NO_high_dmildew_AND_pmildew_is_low_OR_null` (for short: `S1.NONOPLON`) will be taken (Note the crossed circle near ③). It is a "drill-through" symbol showing that the origin of the transition is a substate. On the substatechart Figure 5.10(a), the same symbol indicates that the transition, which originates from state `Wait_for_next_rain`, targets an external state. Therefore, looking at Figure 5.9 ③ , when the guard condition ([$soon(dateE1 + 15)$]) becomes true, the "drill-through" symbol indicates that the substate `Wait_for_next_rain` has to be active for the transition to be taken. That is because when the other substate is active, a mixed treatment is already pending, their is no need to double the order. When all conditions are fulfilled `S1.NONOPLON` becomes active and powdery mildew treatment is ordered (no event is required to trigger it. Here, the transition is purely conditional). The final state is reached after reception of the evTreatmentDone event.

Going back to `Stage_1`'s default transition, left of Figure 5.9, we shall now focus on the management of high powdery mildew infestation observed at "E1".

[*O* ='++'], *powdery mildew is high:* As `Stage_1` targets specifically the early epidemic of powdery mildew, if powdery mildew level is already high at `Evaluation_1`, then a treatment should be performed as soon a possible.

The conditional nodes are structured as follows along the high powdery mildew branch. The second node ⑤ tests for absence of downy mildew in the plot as well as in the region.

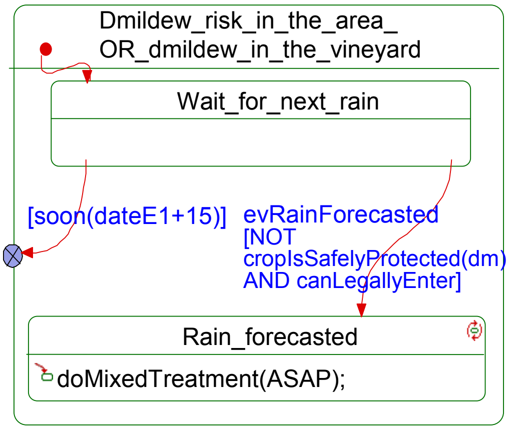
[*M* = *'0' and ILM* ='*0*']=*true:* An application solely against powdery mildew is required as soon as possible. State `High_pmildew_NO_dmildew` is activated ⑤ . The behaviour of its substatechart is presented on Figure 5.10(b). The ordered sequence of action is (i) order a powdery mildew treatment a.s.a.p (state `Pmildew_treatment_ASAP`); (ii) wait for a change of ILM (`Wait_watch_ILM`). (iii) If the ILM value changes to '+', `Wait_for_next_rain` will be entered and monitoring of the rain forecasts will be initiated. Finally when a rain is forecasted, a downy mildew treatment will be ordered (evRainForecasted[*canLegallyEnter*]). However, if ILM remains at '0' or no rain is forecasted the process will be stopped when `Stage_1` is exited.

The philosophy behind the `High_pmildew_NO_dmildew` substatechart (see Figure 5.10(b)) is (i) to execute the mandatory powdery mildew treatment a.s.a.p to control an epidemic that was estimated as '++' during the first evaluation; (ii) then to monitor the risk of downy mildew in the surrounding through ILM; (iii) if the risk increases then monitoring of the contaminating event is activated. This mechanism allows to disjoin powdery mildew and downy mildew treatments, allowing to spare downy mildew treatments when no epidemics was witnessed during the `Evaluation_1`, but also permitting to react when the epidemics of downy mildew is detected past the field evaluation.

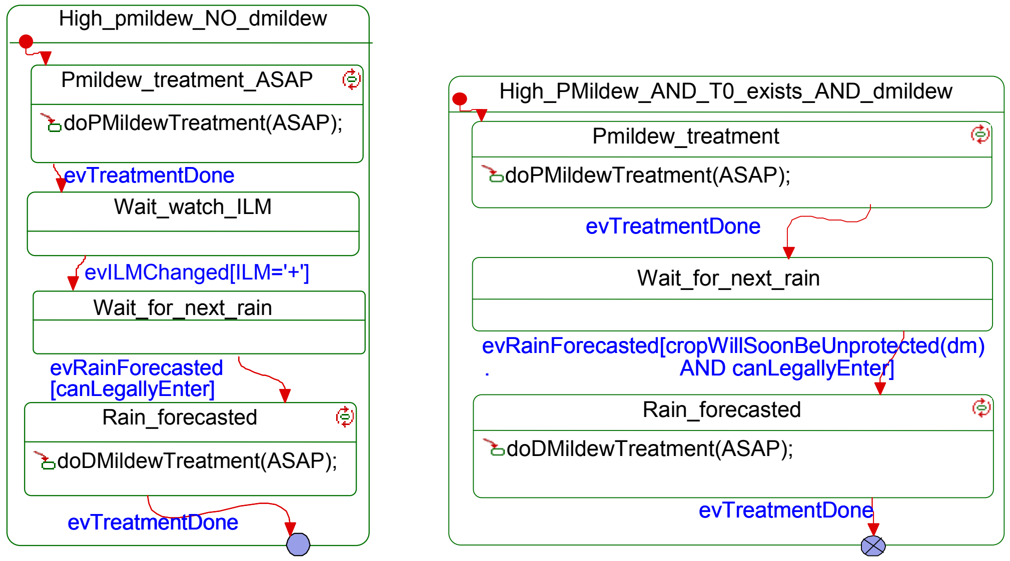
[*M* ='*0*' *and ILM* ='*0*']=*false* On the upper branch (noted [else]), downy mildew is either found in the plot or in the region.

The third node discriminates between existence and absence of a "T0" treatment. In practice, this optional treatment should be extremely rare. In the more general case: no previous treatment against downy mildew was done at `Stage_0`, a mixed treatment is ordered to be carried out as soon as possible, upon entering state `High_pmildew_AND_ANY_dmildew`. Indeed neither powdery nor downy mildew levels are negligible.

In the rare occurrence of "T0" (labelled [*existT*0]), we distinguish two behaviours. Either the downy mildew epidemic was not controlled by the previous treatment and M='++' was observed at "E1", in which case the situation is critical and immediate action is required for both diseases (state `High_pmildew_AND_ANY_dmildew` is entered). Or, on the contrary, "T0" has been efficient and the epidemic seems well controlled. In that case, state `High_PMildew_AND_T0_exists_AND_dmildew` (`S1.HPT0ED` for short) is entered. It may seem inconsistent that moderate downy mildew (M='+') leads to two different decision state. However the existence of "T0" imposes to take into account the remaining AP of that treatment. It permits to postpone the downy mildew application and eventually spare it if the weather is dry.



(a) `Dmildew_risk_in_the_area_OR_dmildew_in_the_vineyard` substatechart



(b) `High_pmildew_NO_dmildew` substatechart

(c) `High_PMildew_AND_T0_exists_AND_dmildew` substatechart

⟳ State holds an action on entry or exit ⊗ Transition comes out from a substate

Figure 5.10: Substatecharts from `Stage_1`

The substatechart of `S1.HPT0ED` (Figure 5.10(c)) shows the following decisions: (i) to treat powdery mildew as soon as possible, then (ii) to enter the weather watch and (iii) spray before it rains if the AP of "T0" is near its end ([*cropWillSoonBeUnprotected*(*dm*) *AND canLegallyEnter*]). `S1.HPT0ED` is exited when downy treatment is notified or when `Stage_1` period is finished.

Concluding with presentation of `Stage_1` (④ Figure 5.9), the transition going from `Stage_1` to `Evaluation_2` is guarded with the following condition: $[existT1\ AND\ date \geq dateE1 + 15\ AND\ canLegallyEnter]$, which guaranties before proceeding to the second evaluation that "T1" has been sprayed, that 15 days at least have elapsed since the previous evaluation and that the plot can be entered. The reason to test "T1's" existence is that GrapeMilDeWS has no other way of ensuring that the treatment orders have been carried out before the other conditions become true.

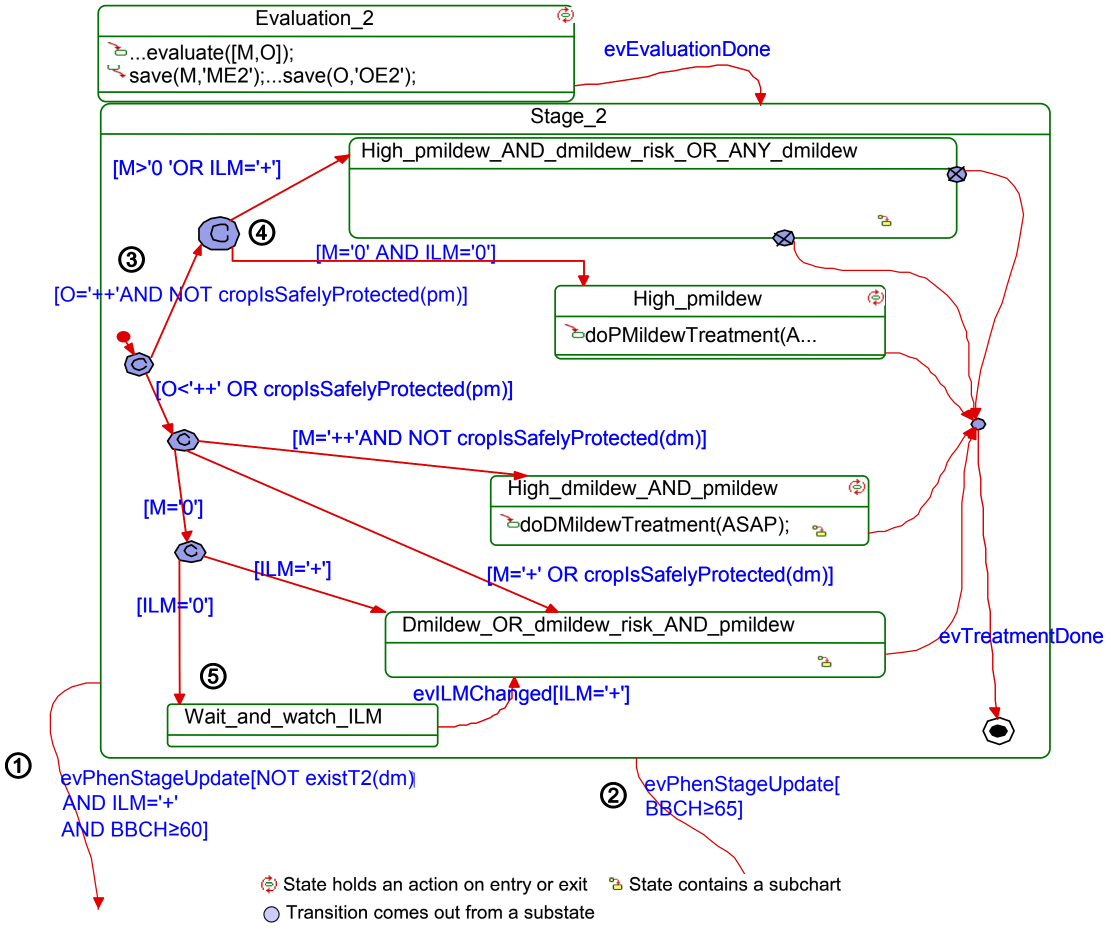


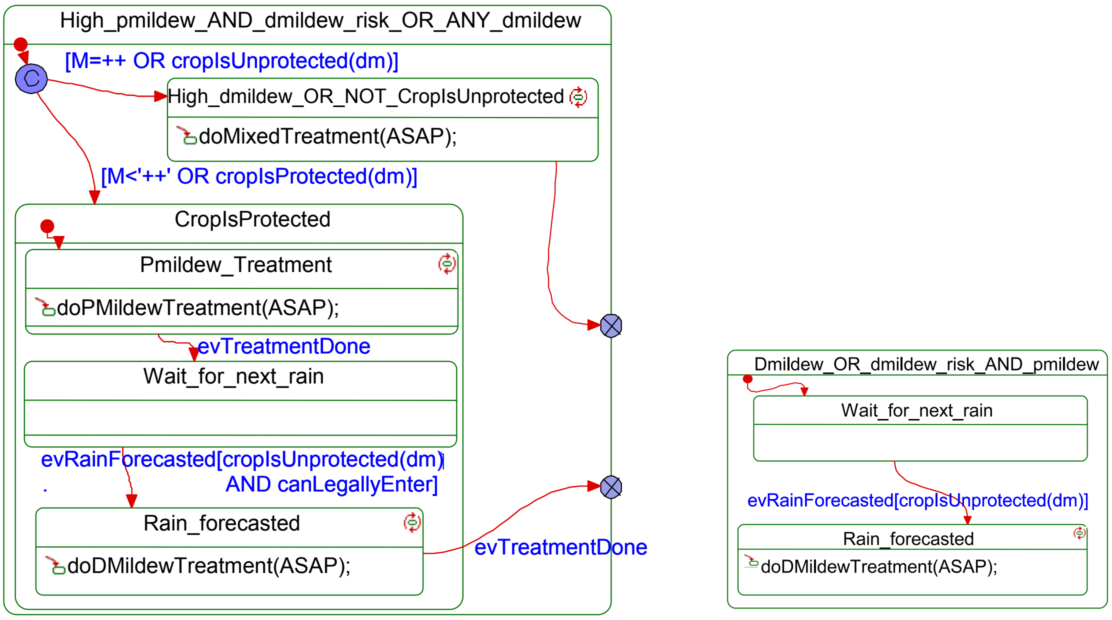
Figure 5.11: GrapeMilDeWS Stage_2

5.4.4.6 Stage 2

The substatechart of `Stage_2` is shown in Figure 5.11. All tags in the following section will refer to Figure 5.11. In the two previous sections, we have been extremely explicit with the Statechart notation. From hereon, we will explain the concepts in a more straightforward manner, the details are to find on the figure.

`Stage_2` is designed to schedule an optional treatment for either powdery or downy mildew or both. The decision is based on the level and evolution of both foliar epidemics observed during the second evaluation. "E2" allows a second assessment of the level of both epidemics, 15 days after "E1". All treatments are optional at `Stage_2` and may be spared if both epidemics are lulled.

Renewing the treatment is not systematic. It requires both the end of "T1's" AP and high epidemic risks (assessed by O, M at "E2" as well as the current ILM and rain forecasts). Under low downy mildew epidemic pressure, the decision would be to withhold the second treatment up to the beginning of the flowering. In which case, the third treatment "T3" would be done a little earlier than what we consider normal (i.e. mid flowering)① and that would permit to spare "T2". Otherwise, when "T2" is actually needed against downy mildew or against both diseases, thanks to the protection provided by "T2", `Stage_3`'s entry will be postponed until mid flowering ② .



(a) High_pmildew_AND_dmildew_risk_OR_ANY_dmildew (b) Dmildew_OR_dmildew_risk_-AND_pmildew

State holds an action on entry or exit State contains a subchart
○ Transition comes out from a substate

Figure 5.12: Substatecharts from `Stage_2`

Having presented the general context of `Stage_2`, we shall give some of the interpretation keys of the decision making in `Stage_2`. Starting from the default transition, following the upper branch ③ , implies that the powdery mildew treatment ("T1O") was done early (i.e. $[NOT\ cropIsSafelyProtected(pm)]$) and that the powdery mildew epidemic is high. Then powdery mildew treatment should be renewed right away. The level of downy mildew will distinguish the substate that activates (2nd conditional node ④).

The same type of reasoning is applied in the lower branch: either powdery mildew was well controlled thanks to "T1O" or the treatment was applied recently w.r.t. `Stage_2`'s entry and the plot is protected (i.e. $[O <'{+}{+}'\ OR\ cropIsSafelyProtected(pm)]$. This justifies that no immediate action be taken against powdery mildew.

When both observed disease levels are low, or well protected, `Wait_and_watch_ILM` is activated ⑤ . Then, when ILM turns to '+', there are two possible behaviours: (i) state `Dmildew_risk AND_pmildew` is activated and the weather watch procedure begins; however (ii) if the flowering has started ($BBCH \geq 60$) then `Stage_2` is exited right away, and treatment "T3", in state `Stage_3` , will be applied in lieu of "T2".

The rest of the behaviour of `Stage_2` is left to read on Figure 5.11. It follows the same "process vocabulary" presented in stages 0 and 1. Details of the substatechart of state `High_pmildew_AND_dmildew risk_OR_ANY_dmildew` is found on Figure 5.12(a). There, the AP management will cause either to order a mixed treatment immediately or to have first the powdery mildew treatment done and then if required by the forecasted weather conditions the downy mildew treatment is ordered. This setup permits to adjust precisely on the level of downy mildew risk. For instance, when the plot is protected or

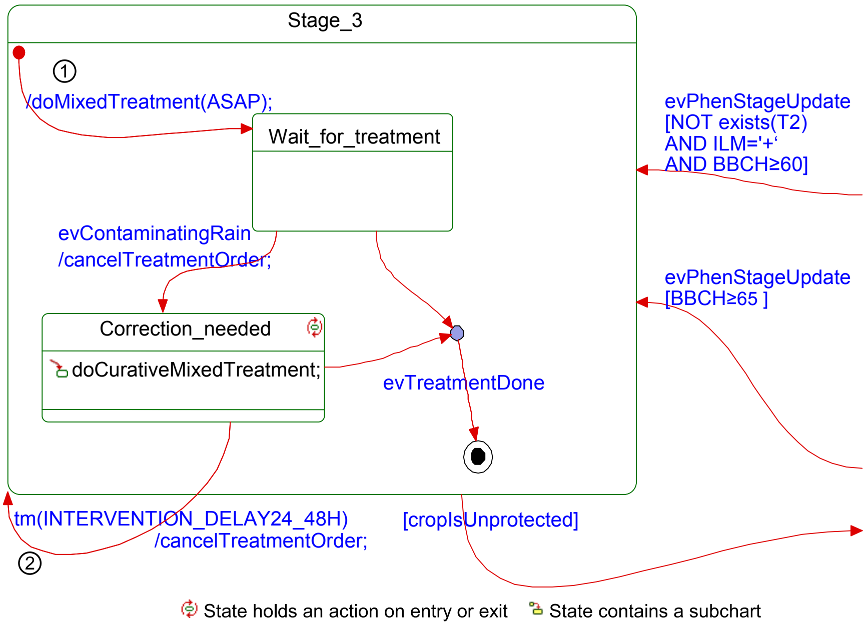


Figure 5.13: GrapeMilDeWS Stage_3

the epidemic is moderate. Our aim is to attempt to spare a downy mildew treatment, but at the risk of having to spray twice.

Figure 5.12(b) is again a version of the weather watch procedure. However here, the AP is taken into account before ordering the downy mildew treatment.

5.4.4.7 Stage 3

The substatechart of `Stage_3` is shown in Figure 5.13.

`Stage_3` has already been partly discussed in the previous sections. This stage holds the third and key treatment "T3" in the GrapeMilDeWS strategy. It protects the flowering period which is recognized as the most critical time during the season w.r.t. powdery and downy mildews. "T3" can be ordered according to two different modalities. First, as "T2M" is skipped "T3" is done early in the flowering([$NOT\ exist T2\ AND\ ILM = `+'\ AND\ BBCH \geq 60$]). Otherwise, if "T2M" was done, it still protects the early flowering, and "T3" will be done at mid flowering. The end of the active periods are synchronised after "T3" since products with long lasting effects against both diseases(see section 5.4.4.12 and 5.4.4.11) are used at `Stage_3` which simplifies the scheduling of the work later on.

error recovery procedure `Stage_3` substatechart shown in Figure 5.13, represents the error recovery procedure which is also associated with every other states ordering a downy mildew treatment or a mixed one. In GrapeMilDeWS, we try to avoid downy mildew treatment when no rain is forecasted. Eventually the rain may fall before the treatment can be carried out. In this case, the procedure is identical to that of `Stage_3` (except for the actions target which may be either downy mildew or both diseases).

On entry, "T3's" mixed treatment is ordered ① (with the product selected according to

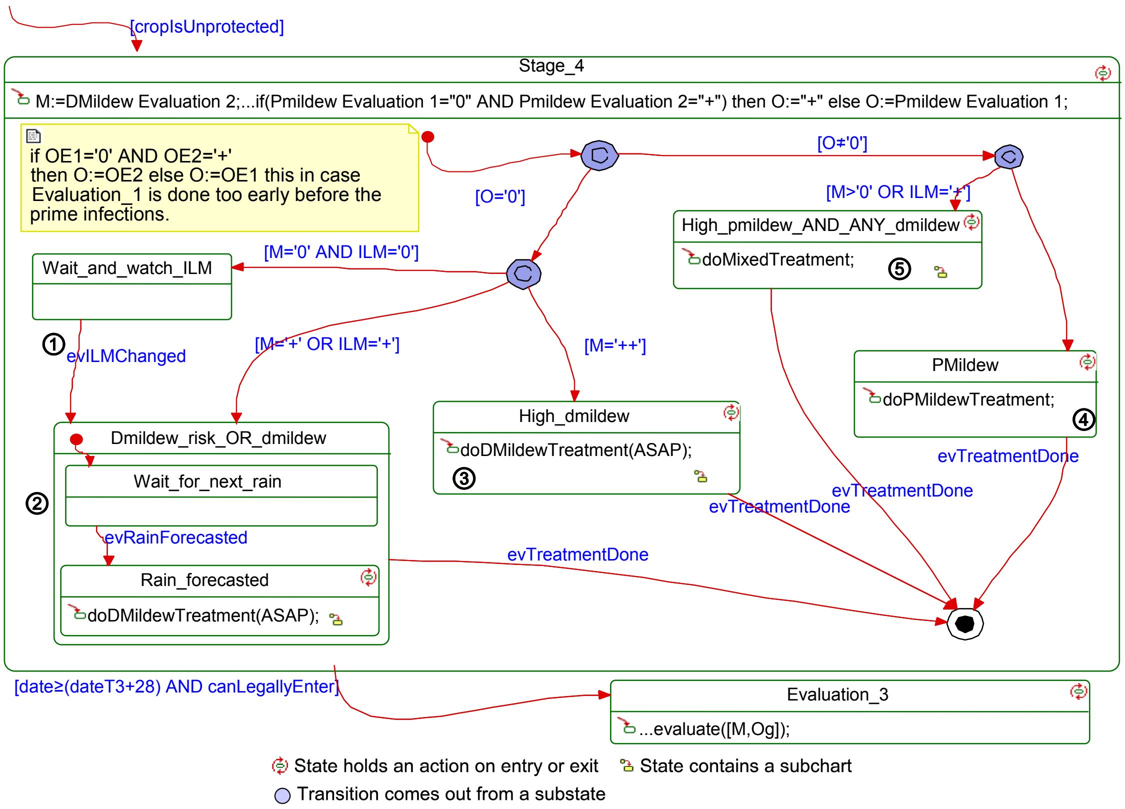


Figure 5.14: GrapeMilDeWS Stage_4

the product selection rule, see section 5.4.4.11). State `Wait_for_treatment` is activated. Eventually, if a contaminating rain occurs before the initial order is executed, emergency actions will be needed to limit the proliferation of the inoculums.

The recovery procedure is to cancel the preventive treatment and replace it by a curative treatment. Depending on the temperature, the curative treatment must be done in less than 24 to 48 hours. Otherwise, the damages are irreversible and the curative treatment becomes useless, therefore the normal process is resumed, `Stage_3` is re-entered ② and the mixed treatment ordered again.

5.4.4.8 Stage 4

The substatechart of `Stage_4` is shown in Figure 5.14.

`Stage_4` is entered after the end of “T3’s” AP. In years with intense early epidemics, the goal of this stage is to protect the growth of the berries, when they are still green, growing and susceptible (pea size: $BBCH \approx 73$). This optional treatment stage should not yield treatment applications on low epidemics years. As for `Stage_3`, the variables O and M are not updated before entering `Stage_4`. Nonetheless, their values are refreshed according to the following rules:
$if\ OE1 = \text{‘0’}\ and\ OE2 = \text{‘++’}then\ O := OE2\ else\ O := OE1$
$M := ME2$
O will take the value it had after `Evaluation_1` (‘OE1’), except when no powdery mildew was found during “E1” and yet high powdery mildew was observed 15 day later during

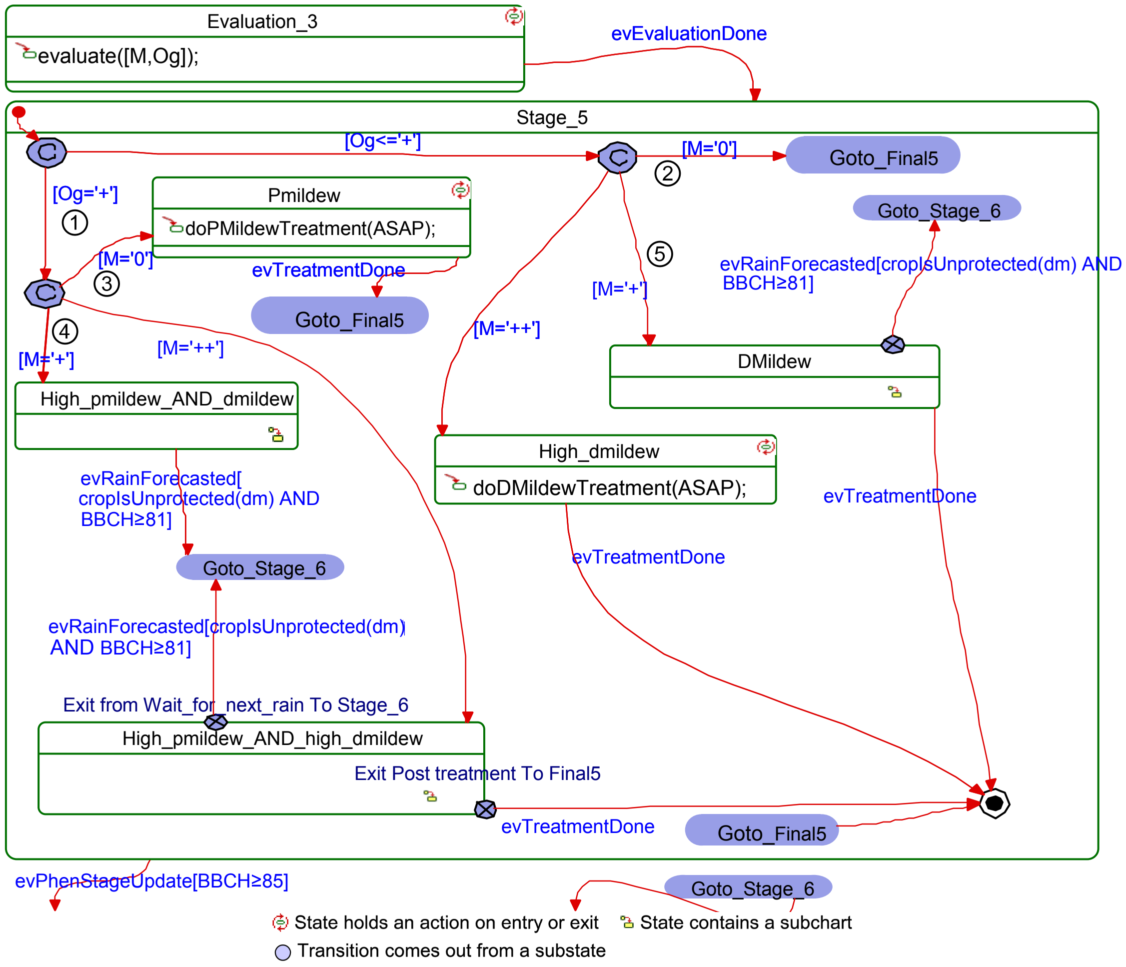


Figure 5.15: GrapeMilDeWS Stage_5

"E2"("OE2"='++'). Downy mildew variable M is always reset to the value found at "E2"("ME2"). ILM during this stage is the only variable that represents the current epidemic conditions.

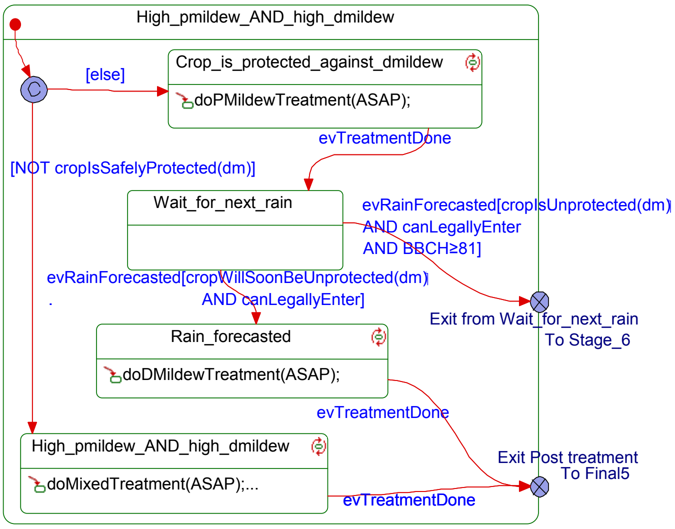
Stage_4 is most illustrative of the general logic of the decisions taken in GrapeMilDeWS (Except for the variables re-assignment on entry). When the pre-flowering epidemics have been low, then only ILM monitoring is done ① . If downy mildew increases in the area, then the weather watch will be started ② . If only one estimator is high, then only its target disease will be treated ③ ④ . Finally, when both powdery and downy mildews estimators are above nil or when risks of downy mildew exist in the area, then a mixed treatment is ordered ⑤ .

Stage_4 is followed by Evaluation_3, 28 days after "T3". That is approximately 2 APs after "T3's" application. Eventually exit may be postponed until the plot becomes accessible again after a late "T4". The targets of Evaluation_3 are downy mildew epidemics on the leaves and powdery mildew on the bunches.

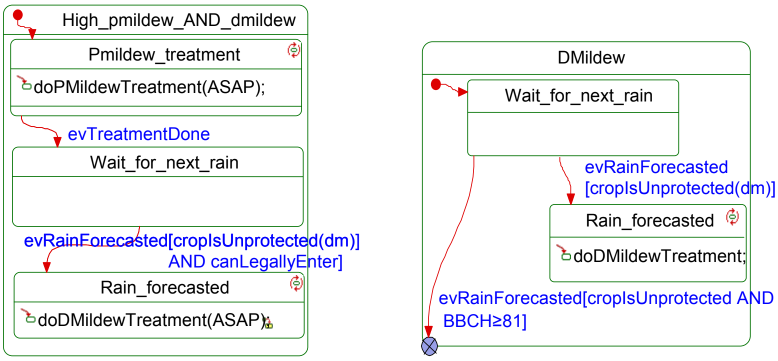
5.4.4.9 Stage 5

The substatechart of Stage_5 is shown in Figure 5.15.

Following Evaluation_3 the system receives a refreshed view of the plot's sanitary



(a) High_pmildew_AND_high_dmildew



(b) High_pmildew_AND_dmildew

(c) DMildew_OR_dmildew_risk

State holds an action on entry or exit State contains a subchart

Figure 5.16: Substatecharts from `Stage_5`

status. Variable Og is used to estimates the intensity of the powdery mildew epidemic on the bunches. Based on this information, if Og is high, ① a treatment is ordered in all cases, whereas no powdery mildew treatment is requested when low. If M is low, then no downy mildew treatment is needed ② ③ . Weather watch is activated when at `Evaluation_3`, M='+' ④ ⑤ . When M='++' a downy mildew treatment is requested as soon as possible.

In state `High_pmildew_AND_high_dmildew` (for short: `S5.HPHD`) is more complex than most substates because powdery mildew on the bunches was found high at "E3" and protection against that disease must therefore be renewed on entry of `Stage_5`. However, protection may still be active against downy mildew (from late "T4M"). Thus sparing a treatment becomes possible if the weather remains dry.

This translates on `S5.HPHDs`' Statechart (Figure 5.16(a)) into the following procedure: if downy mildew treatment "T4M's" AP is nearly over ([$NOT\ cropIsSafelyProtected(dm)$]) then (i) the treatments are mixed. Otherwise, (ii) the powdery mildew treatment is done first, then (iii) the weather watch procedure is started and the downy mildew treatment done only when the rain is forecasted. After the beginning of ripening ([$BBCH \geq 81$])

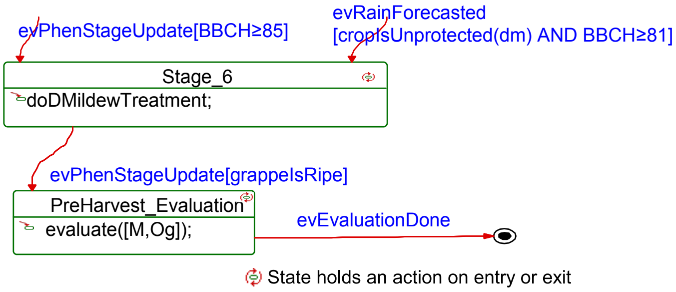


Figure 5.17: GrapeMilDeWS Stage_6

the advent of a rain will force "T6" to be done in advance and thus "T5M" is spared. (iv) `HPHD` is exited through the GOTO_Stage_6 "shortcut" (drill-trough Figure 5.15 and Figure 5.16(a)).

In general at `Stage_5`, the weather watch procedure are all modified as presented above, to attempt to spare a downy mildew treatment by applying "T6" at early ripening instead.

5.4.4.10 Stage 6

The Statechart part of GrapeMilDeWS containing `Stage_6` is shown in Figure 5.17.

`Stage_6` is essentially a mandatory "Bordeaux mixture" (copper) treatment. Its aim is to protect the leaves against late downy mildew epidemics which can cause defoliation, thus damaging the maturation of the grapes. As explained in the above section it may be advanced from mid ripening ($BBCH \geq 85$) to early ripening ($BBCH \geq 81$) to allow sparing "T5". The advantage of copper treatment is that it is not photo reactive. Therefore it offers a long lasting protection. Only rain can actually wash it off. Yet "T6" is sufficient to protect the maturation of the fruits which are not susceptible to powdery and downy mildews. This treatment will not be renewed, even if washed. Indeed partial defoliation is acceptable when it occurs late enough, because the stocks have reserves to carry out maturation to its term (Candolfi-Vasconcelos et al., 1994; Hunter et al., 1995). However, systematic defoliation may, in the long run, tire out the stock and jeopardize production. As a consequence `Stage_6` should be entered as late a possible whenever the weather permits.

We recommend before harvest a last optional evaluation to assess the quality of the GrapeMilDeWS program, no management decision is associated with `Preharvest_evaluation`. This concludes the main process of GrapeMilDeWS. The dormancy period is not taken into account and the program needs to be reset each spring.

After going through the main decision making process of GrapeMilDeWS, the next two sections present the product choice mechanisms involved when treatments are ordered.

Cur. Stage	Conditions		Product Type
`Stage_0`		→	Contact or Prevading
`Stage_1`	$M = ++$	→	Curative
	↳*closetoE*1	→	Systemic
		↳	ANY Protectant
`Stage_2`	*closetoE*2	→	Systemic
		↳	ANY Protectant
`Stage_3`	$M = ++$	→	Curative
		↳	Systemic
`Stage_4`		→	ANY Protectant
`Stage_5`		→	ANY Protectant
`Stage_6`		→	Bordeaux mixture

(a) Targeting downy mildew

Cur. Stage	Conditions		Product Type
`Stage_1`		→	SBI type 1
`Stage_2`		→	Strobilurin
`Stage_3`		→	Strobilurin
`Stage_4`		→	Quinoxifen
`Stage_5`	*NOT existT*2	→	Strobilurin
		↳	SBI type 2

(b) Targeting powdery mildew

Table 5.1: GrapeMilDeWS product selection rules

5.4.4.11 Downy mildew product choice

In GrapeMilDeWS, the types of products are structured according to their mode of action and properties, as inclusive sets (Figure 5.18). When the larger set is recommended for an application, any type of products belonging to that large set may be chosen. The specific choice is left to the vineyard manager.

By default, protectant fungicides are chosen against downy mildew. If the error recovery procedure is activated, then curative products are selected.

In the case of a treatment during `Stage_0` (see Table 5.1(a)), the choice of a contact or a pervading product is made, both for their partial systemic properties and their short active period. Thus "T0" will have a small disruption effect on the timing of "T1".

If M='++' in `Stage_1` then a curative treatment will be ordered as soon as possible (section 5.4.4.5). Otherwise, the product choice will depend on the desired length of the active period: (i) the application is done early (*closeToE*1) then a systemic treatment will offer a longer lasting protection than (ii) the protectant fungicide chosen by default. That same reasoning holds for "T2".

"T3" should be systemic in most cases, except when during a second evaluation M='++'. Then a curative treatment is preferred, as the epidemic started on a high trend. Finally, the last treatment in `Stage_6` will use copper based products for its long lasting efficiency.

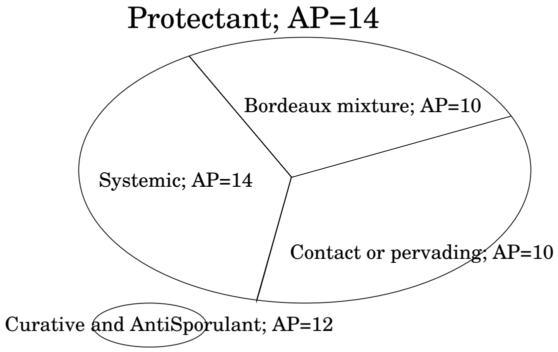


Figure 5.18: Partition of the families of products targeting downy mildew

5.4.4.12 Powdery mildew product choice

Products need to be diversified in order to lower the risk of selecting a resistant stock of *Erysiphe necator*.

Four kind of active molecules may be used during the season. The mandatory treatments should be done using Sterol Biosynthesis Inhibitor (SBI) (at `Stage_1` see Table 5.1(b)) and Strobilurin (at `Stage_3`). The optional `Stage_4` treatment will be done using either Quinoxyfen or Metrafenon. An extra Strobilurin treatment is allowed in the limit of two applications overall ("T3"+("T2" Or "T5") Thus "T5" will use Strobilurins only if no powdery mildew treatment was applied at "T2" otherwise (see Table 5.1(b) last line) "T5" should use a SBI of the second group.

This concludes the section dedicated to presenting GrapeMilDeWS.

5.4.5 Discussion

The discussion will be restricted to the innovation of the approach in the field of IPM and to the advantages of the formal representation. The quality of the IPM solution, with respect to the experimental results and the accuracy of the modelling shall be discussed in a second article (GrapeMilDeWS part 2 (Léger et al., 2008b)).

5.4.5.1 GrapeMilDeWS an IPM decision Workflow

General practice in IPM is to establish a reliable model of the pathogen development and then discover the decision thresholds which control the epidemics. During our preliminary experiments (2001-2004), risk models where used, EPI (Tran Manh Sung et al., 1990) and Milvit (Rouzet and Jacquin, 2003). It led to over protection as these models are not adapted to the plot scale (epidemics would be predicted when no epidemics developed at our plot scale). These false positive predictions showed that some factors were missing in these models. Since, we have preferred field observations, although the Milvit model is still taken into account through the plant protection service advisory bulletins used to update the ILM variable. The plant protection service uses Milvit to build the forecasts included in the disease information bulletins.

Choosing observations, introduced operational problems. The designers had to take into account multiple criteria: efficiency of the IPM program, workload and cost. Farmers know well these type of multiple criteria strategy design problems (Girard and Hubert, 1999). The chosen solution is both pragmatic and scientific: (i) make a limited number of field observations positioned at strategic time and cut the crop protection decision making in sub goals: the stages, designed according to the sensitivity of the plant. This approach is consistent with the way farmer organize their activity and make decisions (Papy, 1998).

Coupling the control of both Powdery and Downy mildews is again consistent with common practices; and is quite original in IPM: no IPM strategy targeting multiple pathogens was found in the grapevine literature. Originally our concern was focused on the operationnality of the process and led us toward this multiple pathogen management decision workflow.

Compared to having two strategies and trying to manage the conflicts, our solution leads to a few tradeoffs between the optimal management of both pathogen taken separately. The advantage is that these tradeoffs are rationalised and can be quantified by the experiments.

The whole project is influenced by the "decision process and action model" theory (Sebillote and Soler, 1988). However, the action model theory is a diagnosis tool of farmers' practices. In our case, the designers are pathologists and the goal is to represent the decision process they have created. Here the formal modelling language served as a tool for an exhaustive and explicit transfer of process knowledge toward other researchers.

Further work will be dedicated in transferring this tool to the professionals. In order to achieve that goal, a DSS will be designed. This formal model will also be used for controlling the scalability of the solution. We are involved in a design loop, modification will be made if the present design proves too costly. However before moving onto this path, GrapeMilDeWS will be agronomically evaluated at a wider scale. These experiments justify an exhaustive model for knowledge transfer between R&D participants, as the designers will not be able to manage all these experiments.

5.4.5.2 Formal graphical modelling

In this section, we shall investigate why formal process modelling is novel and promising.

Building simulation models for decision making is quite common in systems agronomy (Attonaty et al., 1999, 1994; Cros et al., 2001; Wauchope et al., 2003), however, formal modelling using the engineering approach as proposed in Zheng (2006), is not developing in agriculture. Although the trend is rapidly acquiring momentum in the fields of systems biology (Webb and White, 2005), manufacturing (Baresi et al., 1997; Castillo and Smith, 2002) and medical research (ten Teije et al., 2006).

As Harel (2004) advocates, the tools developed in computer science to model and verify the behaviour and properties of complex real time systems are now mature and can be used in other systemic sciences. Systems such as an immune system (Cohen, 2007) or a complete *Caenorhabditis elegans* nematode worm (Kam et al., 2003) have been modelled formally integrating the available knowledge from the literature.

Simulations associated with animation (Harel et al., 2002; Philippi and Hill, 2007) permits the expression and step by step observation of emerging behaviours which in turn can be researched either for complement in the literature (under-specification of the model) or through new in vivo experimentations. This approach permits to represent knowledge at different scales, levels of details and of abstraction. Yet the most interesting feature of systems formal modelling is the ability to do "Model Checking" (Alur and Dill, 1994).

Model checking is a set of technique using modal logics to prove temporal or even real time (Yovine, 1993; Penczek and Pólrola, 2006) properties of systems: reachability, vivacity and safety. Safety for example permits to control that a forbidden state of the system can never be active. This is achieved without the need to test all possible configuration in simulations. In our case a safety property that can be checked would be to guaranty that no curative treatment will be applied later than 48h after a rain event on an unprotected plot. In agronomy, Model checking has been used by Largouët (2000) on a land use problem and by Hélias (2003) in organising the use of pork effluent for the fertilization of sugar cane fields.

In order to apply model checking, a formal model is required (i.e. the model should have an equivalent formulation in a finite state automaton formalism). Here, we focus on the graphical representation of such formal model. There are several DES graphic modelling languages. But the majority refer to either Petri net's or state based modelling (Mosterman and Vangheluwe, 2004).

We chose Statechart for its intuitiveness. Based on higraphs (Harel, 1988; Grossman and Harel, 1997), it is efficient for the visual representation of union and conjunction. Among the variety of Statechart, we chose Rhapsody's (Harel and Kugler, 2004) which is UML compliant, but Glinz (he proposes truth tables for complex triggering conditions) may prove more adapted to readers unfamiliar to the notation as Cruz-Lemus et al. (2005) indicate in their study of the ergonomic of the Statechart language. Our experience working with experts is that Statechart requires some learning time, but are able to represent both logical rules, and sequentiality. The nesting capabilities allow to focus on different mater at different scales.

5.4.6 Conclusion

We presented two innovations in this paper. The first is methodological, with the use of a formal graphical language for the modelling of processes and decisions, that is workable in agriculture, with the advantage of being mathematically sound and of producing executable software, with a wide variety of implementations available (Harel et al., 1990; Telelogic, 2007; IBM, 2007; Gentleware, 2007; Mathsworks, 2008) producing implemented code in different programming languages.

The second point of this article, was to present the GrapeMilDeWS decision workflow itself. Its novelty is both due to its expert workflow based approach at reducing the number of application to grapevine, and to the proposition of an IPM solution that is very flexible in the number of treatments, reducing the number of applications when possible, and protecting when required, while taking into account the farmers operational constraints.

5.5 Discussion du Chapitre

Il s'agissait ici de donner une vision exhaustive du modèle Mildium tel qu'il a été recueilli au cours de l'année 2006 auprès des experts concepteurs grâce à la méthode présentée au chapitre 4. Dans le cadre du processus de conception itératif dans lequel ce travail se situe, le modèle présenté correspond à une version.

Mildium a été expérimenté de manière experte (c'est à dire sans le modèle) pendant les saisons 2005 et 2006 à Bordeaux. Puis au cours de l'année 2007, le modèle a été utilisé par les experts comme référence pour leurs expérimentations à Bordeaux. Enfin en 2008, ce modèle a servi de base aux expérimentations menées dans le cadre du projet " Conception et transfert de systèmes décisionnels pour une réduction des traitements phytosanitaires sur vigne " (SyDéRéT) [g] dans divers vignobles méridionaux..

Grâce à l'établissement d'une version, un travail d'étude systématique qui comprend à la fois la validation informatique et l'expérimentation permet de produire un corpus de connaissances qui pourra être associé à la phase de « conception & amélioration » suivante. Ainsi, il a été possible au cours des années 2005 à 2008 d'expérimenter le POD dans divers contextes épidémiques annuels et dans plusieurs régions et donc d'accumuler des observations. En 2009, une nouvelle version du POD devrait être reformulée prenant en compte l'expérience acquise.

Séparer nettement les phases d'innovations des phases d'études et d'expérimentations, relève de la même démarche qui m'a poussé à refuser la réalisation d'un système expert à portée générique. Il me semblait qu'un système trop versatile eut été difficile à étudier alors qu'avec l'approche processus adoptée, l'ensemble des comportements reste plus restreint. Cependant, l'étude d'un procédé comme Mildium dans ses composantes décisionnelles, temporelles et biologiques, reste encore le sujet d'un questionnement méthodologique.

Le choix de fixer et d'identifier les versions permet simplement de documenter les évolutions, d'associer une modification à une difficulté due aux expérimentations faites aux champs ou à l'analyse du modèle (par simulation par exemple). Cette volonté de produire une conception incrémentale, si elle est répandue dans le domaine du génie logiciel, est finalement assez rare dans le domaine de la protection des végétaux.

Rappelons que la démarche de conception de POD menée à Santé Végétale est innovante en ceci qu'elle propose une abstraction du raisonnement de la protection par rapport à son instanciation réelle sur la base d'indicateurs concrets.

5.6 Conclusion du Chapitre

Le modèle présenté ici est issu du recueil de connaissances. Il s'agit d'une version intermédiaire du procédé Mildium à replacer dans un processus de conception itératif. Le modèle a été accepté par les experts lors du recueil (voir chap. 4). Une approche quantitative restait nécessaire pour estimer la qualité du modèle. La méthode de validation fait l'objet du chapitre 6.

---

[g] retenu en 2008 à l'Appel à Projets Protection Vigne (A2PV) du Ministère de l'Agriculture