2010-OS-1

Decision Support Systems in Agriculture, Food and the Environment:
Trends, Applications and Advances

Basil Manos
Aristotle University of Thessaloniki, Greece

Konstantinos Paparrizos
University of Macedeonia, Greece

Nikolaos Matsatsinis
Technical University of Crete, Greece

Jason Papathanasiou
University of Macedeonia, Greece

Information Science REFERENCE

INFORMATION SCIENCE REFERENCE

Hershey · New York

Director of Editorial Content:	Kristin Klinger
Director of Book Publications:	Julia Mosemann
Acquisitions Editor:	Lindsay Johnston
Development Editor:	Christine Bufton
Publishing Assistant:	Travis Gundrum; Natalie Pronio
Typesetter:	Keith Glazewski; Travis Gundrum; Casey Conapitski
Production Editor:	Jamie Snavely
Cover Design:	Lisa Tosheff
Printed at:	Yurchak Printing Inc.

Published in the United States of America by
Information Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue
Hershey PA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com

Library of Congress Cataloging-in-Publication Data

Decision support systems in agriculture, food and the environment : trends, applications and advances / Basil Manos ... [et al.], editors.
p. cm.
Includes bibliographical references and index.
ISBN 978-1-61520-881-4 (hbk.) -- ISBN 978-1-61520-882-1 (ebook) 1. Agriculture--Decision making. 2. Agriculture--Data processing. 3. Agriculture--Management--Information technology. 4. Agricultural informatics. 5. Decision support systems. I. Manos, Basil, 1950-
S494.5.D3D39 2010
630.685--dc22
2009052438

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

All work contributed to this book is new, previously-unpublished material. The views expressed in this book are those of the authors, but not necessarily of the publisher.

Chapter 12
GrapeMilDeWS:
A Formally Designed Integrated Pest Management Decision Process against Grapevine Powdery and Downy Mildews

Bertrand Léger
Cemagref–UMR 1201 ITAP (Cemagref-Supagro), NRA–UMR1065 Santé Végétale & Institut des Sciences de la Vigne et du Vin, France

Olivier Naud
Cemagref–UMR 1201 ITAP (Cemagref-Supagro), France

Véronique Bellon-Maurel
Cemagref–UMR 1201 ITAP (Cemagref-Supagro), France

Michel Clerjeau
INRA–UMR1065 Santé Végétale & Institut des Sciences de la Vigne et du Vin, France

Laurent Delière
INRA–UMR1065 Santé Végétale & Institut des Sciences de la Vigne et du Vin, France

Philippe Cartolaro
INRA–UMR1065 Santé Végétale & Institut des Sciences de la Vigne et du Vin, France

Lionel Delbac
INRA–UMR1065 Santé Végétale & Institut des Sciences de la Vigne et du Vin, France

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 which causes powdery mildew and Plasmopara viticola which causes downy mildew. GrapeMilDeWS has been designed and tested 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 effective to model this kind of Decision Workflow System. The formalism is introduced and the GrapeMilDeWS system thoroughly described. Formal modelling is discussed as a representation of the dynamics of decision making in pest management and as an aid to large scale experiments.

DOI: 10.4018/978-1-61520-881-4.ch012

INTRODUCTION

Since 2001, The INRA[1] santé végétale (plant health) laboratory has undertaken the design of pest management "agronomical decision rules" in viticulture. Based on observation, scientific knowledge and operational expertise, these decision rules aim to come as close to "Integrated Pest Management" (IPM) (Boller, Avilla, Gendrier, Jörg, & Malavolta, 1998; Kogan, 1998) as possible and to allow for significant reductions in the number of pesticide applications.

After a few years of experimenting with single pest designs, the team chose to focus on downy mildew (*Plasmopara viticola*) and powdery mildew (*Erysiphe necator*). In France, these two diseases represent 80% of the treatments applied on grapevine (ASK, 2000) and the practice of coupling treatments is particularly widespread with these diseases.

The first aim of the work presented here is therefore to move from a one-plot/one-pest approach to a more pragmatic approach that pairs treatments against two diseases, as is common among vine growers. Furthermore, the team considered grapevine mildews to be a good case study for the design of multiple target (i.e. integrated) pest management strategies.

Besides a demonstrative case study, a desired outcome of the research would be to transfer an operational Decision Support System (DSS) at the farm scale that would help to reduce significantly the number of fungicide applications and yet would guarantee that the production targets (both qualitative and quantitative) are reached.

Before implementing a DSS however, the phytopathologists started by designing a *prescriptive crop protection decision strategy* to support the scientific evaluation of the various innovations that they blended together. This strategy involves a process, beginning at bud break in spring through to harvest. The decisions taken in a given period are influenced by previous decisions as well as the phenological development of the plot and the evolution of the crop's sensitivity to each pathogen. Designing a multi-target strategy is quite complex. In particular, response priorities between the two diseases evolve during the season, which makes decision rules difficult to write out. In order to carry out this program, computer scientists were brought into the team to help formalize, specify and evaluate the design. The model that follows is the result of this collaboration. This mathematically formal model representation had to be both understandable by phytopathologists other than its designers and suited for computer simulations. Indeed computer simulation is considered to be very helpful in the design and testing of new cropping systems (Aubry et al., 1997; Cros, Duru, Garcia, & Martin-Clouaire, 2004; Martin-Clouaire & Rellier, 2003; Sebillote, 1987). Moreover, a formalized model of the strategy should be easy to implement into a DSS once fully validated.

It must be noted that current knowledge of the vineyard pathosystem makes it difficult to simulate epidemics and damages at the plot scale and thus even more difficult to compute optimal crop protection strategies (see review of optimization techniques, in Dent, 1995). These techniques involve linear programming or dynamic programming (e.g. Feldman & Curry, 1982; Shoemaker, 1984), but these theoretical achievements are difficult to implement in practice. For instance, dynamic programming has been largely abandoned in recent years as building a realistic model for the pathosystem often requires a large number of variables which makes the problem quickly intractable.

In agriculture and particularly in pest management, decision making is generally modelled through a rule-based expert system (ES), (e.g. Mahaman, Passam, Sideridis, & Yialouris, 2003; Zadoks, 1989). Our approach is original in that it emphasizes the sequentiality and temporality of decisions. Our hypothesis, which is supported by (Girard & Hubert, 1999), is that emphasizing temporality forces the experts (i.e. the pathologists) to give an exhaustive specification of their crop protection strategy (Léger & Naud, 2009). We shall hereon abandon the term "agronomical decision rule" in favour of the concept of a *crop protection decision workflow system* (CPDeWS), which better accounts for the sequential and integrative structure of the CPDeWS. The workflow terminology also accounts for our attempt to encompass some of the managerial aspects of crop protection. The literature on workflows and workflow modelling is rich. One can refer to (van der Aalst & van Hee, 2002) for a broad presentation of the field. In agriculture, the concepts of business process management, workflow modelling and service oriented architecture have only recently emerged with an information centric point of view (Nash, Dreger, Schwarz, Bill, & Werner, 2009; Steinberger, Rothmund, & Auernhammer, 2009). Some authors emphasize on the need for modelling the farm business processes in order to cope with the complexity of agri-food supply chain networks (Wolfert, Matocha, Verloop, & Beulens, 2009).

Our work is also grounded in the French agronomic tradition, which has developed the concept of the "general model" since the late 1980's to account for the way farmers make their decisions and manage their farms. Traditionally, this qualitative framework is targeted at strategic planning (e.g. J. M. Attonaty, Chatelin, Poussin, & Soler, 1994; Chatelin et al., 2005; Cros et al., 2004; Martin-Clouaire & Rellier, 2009) and accounts for the fact that decisions modify both the production system through "technical itineraries" (Sebillotte, 1978) and the farmers' representations (i.e. his *a priori* plans through a "model for action" (Sebillote & Soler, 1988).

However, we have empirical evidence that *tactical decision making can be critical* while aiming at pesticide use reduction and we have concluded that *the tactical set of logical decision rules that may trigger treatment should be adapted during the season*. The CPDeWS we have designed is thus a system that organizes the collection of information, the decision making, and the treatment applications in time, for tactical use during the season. The model is to be implemented in a particular kind of DSS that would be called "CPDeWS management" system.

The purpose of this paper is to present a formalized CPDeWS named GrapeMilDeWS, and discuss the benefits and limitations of the chosen formalism. GrapeMilDeWS stands for Grapevine protection against downy and powdery Mildew Decision Workflow System.

The first part of this paper consists of a description of the Statechart formalism used for the model. In the second part, key features of GrapeMilDeWS are presented in detail. Extensive comments and explanations are made to allow those without *a priori* knowledge of the Statechart language to understand the contents of GrapeMilDeWS. The assessment of the model is not presented here, but can be found in (Léger, 2008, pp. 162-187). Finally, we discuss the choice of the modelling technique and some implications of our work in the design of CPDeWS and DSSs.

PRINCIPLES AND HYPOTHESIS FOR THE DESIGN OF THE GRAPEMILDEWS DECISION WORKFLOW

Overall Crop Protection Strategy

GrapeMilDeWS aims to avoid yield losses, not to avoid disease symptoms. This represents a shift from common practice in viticulture. This aim is achieved (i) by *controlling low epidemics* (i.e. maintaining them 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 carried out against powdery mildew).

The number and the timing of the fungicide applications are adapted to the plots' specific epidemic conditions through intensive use of various data sources, mostly from the plot itself.

When a treatment is required for one disease, the other will be dealt with 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 approach simplifies the management of treatments against multiple pathogens which otherwise would impose strong operational constraints on the grower.

To further alleviate the work load, the designers of GrapeMilDeWS have restricted themselves to a limited number of well-defined evaluations of the diseases levels in the plot. All plot observations lead to one or more treatment decisions. Observations are used to decide not to monitor the quality or to relieve the grower's anxiety as is too often witnessed in the vineyards (Léger, 2008, p. 236). Up to three field observations are done before flowering (one of them is optional) while a third mandatory observation is done a month after flowering.

For the sake of pragmatism and safety, the crop is systematically protected at the flowering, which is the period of highest susceptibility. There is then no need to estimate the level of infestation during this period.

Observations and Information Generation

The treatment decisions are mostly made based on epidemic estimates, at the plot scale: these estimates are based on observations on leaves and bunches, with a fixed sampling protocol. 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 the level of powdery mildew on the bunches (Og *for Oïdium grappes*: bunches powdery mildew)
- M for the level of downy mildew on the leaves (M for *Mildiou*: downy mildew)

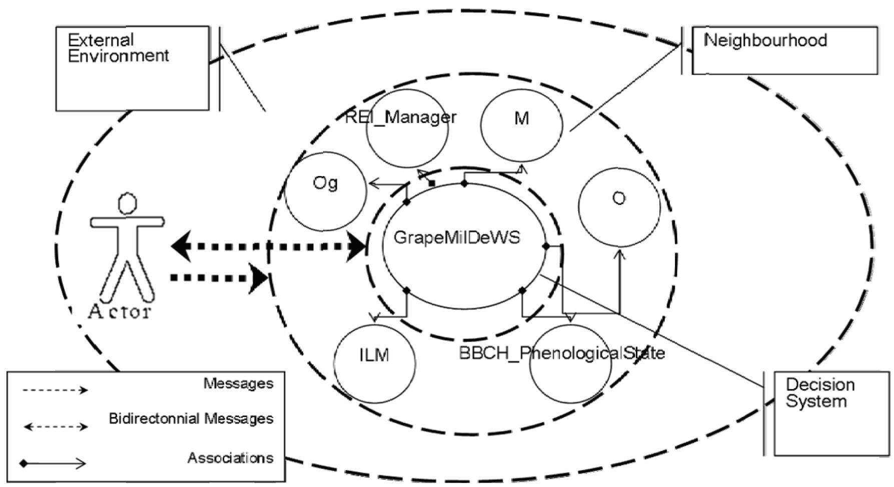
The number of discrete values for a variable varies from 2 to 3, depending on the disease, and the observation date. These modalities encode the qualitative expert risk assessment as follows: ('0') for absence or low epidemic risk; ('+') for moderate to high epidemic risk; and ('++') for very high epidemic risk. The threshold values between the different modalities evolve with the phenology of the vine. This allows adjusting the potential 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 risks of disease development at a larger geographical scale 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[2] for example (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 Meteo France weather forecast service.

The variables (M, O and ILM) are built with thresholds which are modified during the season. This has the effect of embedding some expertise on the dynamics and the dangerousness of the epidemics, into the three estimators. This provides

Figure 1. Three scopes are defined: the system, the neighbourhood, the environment



GrapeMilDeWS and the end user with data which are more easily interpreted.

Model's Architecture

Three Levels of Data Access

As shown in Figure 1, information generation and exchange can be organized in three scopes according to different levels of the information access rights.

The first scope is the environment of the decision system. More specifically, 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. As events are not persistent information, a part of the communications are re-routed to the neighbourhood variables to make it perennial, the rest of the events 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 above) as well as the ILM variable, the phenological stage and the restricted entry interval manager [3](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. The variables that compose the neighbourhood are data that may be used by other processes than a particular plot's GrapeMilDeWS instance (e.g. in a DSS at the vineyard estate level, the ILM variable may be shared by many plots).

The third scope is the GrapeMilDeWS Statechart model itself, inside which the system's control over the data is total and internal variables are of private use.

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 the decision making process in crop protection.

THEORETICAL INTRODUCTION TO STATECHART FORMAL MODELLING

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

The Choice of Discrete Event Systems

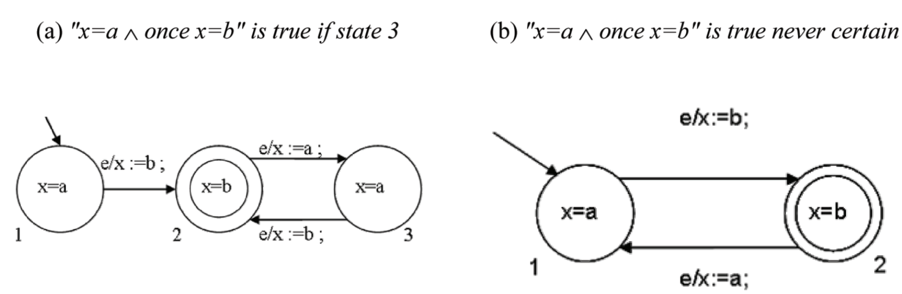
The crop protection's decision system is modelled as a flow of decision leading to work operations. Our aim is to represent the temporal dimension of the CPDeWS, for the whole growing season. The continuous dynamics of phenology and of epidemics can be represented at the plot scale by differential equations. However, we have chosen to model the CPDeWS as a Discrete Event System (DES). Indeed, the IPM experts make decisions based on thresholds defined on the epidemics and the phenological stage variables. The variables are therefore discretized 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.

Among DES formalisms, we chose the diagrammatic language of Statechart. We showed in (Léger & Naud, 2009) that Statecharts are relevant mediation tools between the phytopathologists designers and the knowledge-management researchers for eliciting the formal model, as they are depicted by readable graphs. Computationally speaking, Statecharts can be assimilated to Finite State Automata (FSA). The later are presented in the next section.

Finite State Automata

FSA are mathematical machines which model the evolution of a discrete event system in time. They are mathematically depicted by directed graphs where nodes are states and edges are transitions (Figure 2). Transitions are labelled with events.

Figure 2. State diagram: used to check a system's property



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* and is modelled using a FSA. Figure 2 gives two representations of this system. We have labelled the states with *x*'s values. The event label '*e*' 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. An action is said "atomic" if it cannot be interrupted. This is the "Mealy machine" semantic of FSA, note that there is a second canonical semantics, the Moore machine semantic where actions are executed during the state.

In Figure 2(a), state 3 is different from state 1 even though they both record the same property of the system: *x*=*a*. 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 shown in Figure 2(b).

We use FSA to monitor relevant phenomena and to describe the appropriate response (observations, treatments) during the crop protection season. The states are then more informative than the combination of all input variables. States depict the progression of the decision process and decision outputs. 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 drawback: the number of states becomes unmanageable as soon as concurrent processes are modelled. The number of states for a system with multiple processes is the Cartesian product of the number of states of each independent process. 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)[4] is relevant during the whole season, thus each decision state is multiplied by the number of states in the AP monitoring process. Statechart was invented to avoid combinatorial problems during design thanks to hierarchy and concurrency constructs.

Statechart

The next section introduces the main syntactical features and some semantic elements of the Statechart language. For accessible yet more complete presentation of Statechart refer to (Harel, 1987; Harel & Kugler, 2004). The popularity of Statechart led to the design of many flavours of Statechart (Maggiolo-Schettini, Peron, & Tini, 2003; von der Beeck, 1994). Statechart is part of the Unified Modelling Language (UML 2.0, see OMG, 2007), it thus supports object oriented design. We have implemented the GrapeMilDeWS model using the Rhapsody® software by Telelogic-IBM (Harel & Gery, 1996).

The Graphical Syntax of the Statechart

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

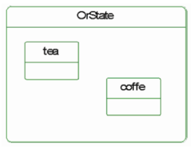
States Harel introduces 4 kinds of states in Statechart (see Figure 3): the *simple states,* Figure 3(a), which are close equivalent to the FSA states; *the final states,* Figure 3(b), are called the acceptor states and represent the completion of a statechart or substatechart.

Figure 3. Statechart's different kind of states

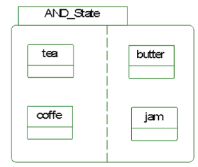
(a) *a simple state*

(b) *a final state marks completion of a process*

(c) *an OR-state contains mutually exclusive substates*



(d) *an AND-state contains simultaneous processes*



Hierarchy (i.e. substatechart) is made possible by: the *"OR-state"*Figure 3(c) which includes exclusive substates inside a parent "*OR-state*" and the *"AND-state"*Figure 3(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 de-activates.

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 follows:

*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 said "potentiated" 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 null transition and is taken "as soon" as its origin states becomes active, provided the guard is true[5]. 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 potentiated.

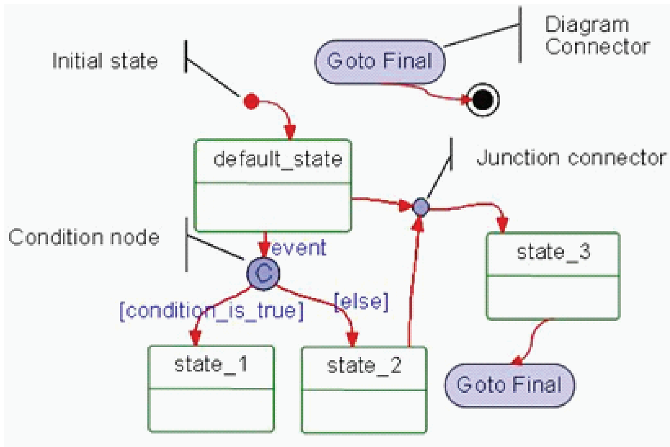
Guards are boolean conditions that control if a potentiated transition can be taken.

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

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 fork and junction. 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. See example in Figure 4.

Readers interested in the formal definition of the UML Statechart semantic (that we use) should refer to (Damm, Josko, Votintseva, & Pnueli, 2003). Having established why the system at hand is modelled as a DES and having introduced the formalism, we now present details of GrapeMilDeWS' formal model

Figure 4. Pseudo-states (initial state, condition, junction and fork as well as "diagram connectors") are useful graphical abbreviations



GRAPEMILDEWS DETAILED PRESENTATION

Statechart's Structure

GrapeMilDeWS is composed of four independent sub systems or functional processes which run simultaneously (see Figure 5). Implemented in Statechart, these four functions are modelled as high level AND-States. Along with the main process, are two product choice rules, one for each target disease. This is because the choice of phytosanitary specialty depends on the vine development stages. In experimental practice some flexibility is given; the model only gives the expert designers' best choice as classes of molecule. The last AND-State is used to manage the active periods (AP) of the last treatment against each disease. For various reasons, including the leaf growth, the efficiency of any product decreases over time. To account for this, AP management provides such functions as *IsSafelyProtected* and *IsProtected*, that are used in the main process, for example in stage 2. When *IsSafelyProtected* is *false* and *IsProtected* is *true*, this means that the product has lost part of its efficiency.

In the following, we will describe the main decision process. We will carefully detail a significant part of the model so that the reader can check the expressiveness of the formalism in regards to

Figure 5. Four concurrent (simultaneous) sub-systems compose GrapeMilDeWS. Each sub system is modelled by a substatechart and communicates with the other processes through event messages.

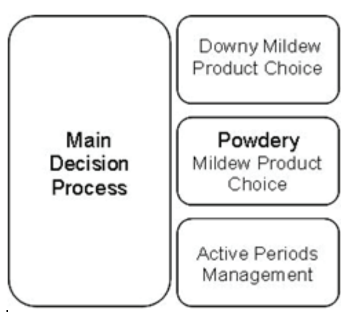
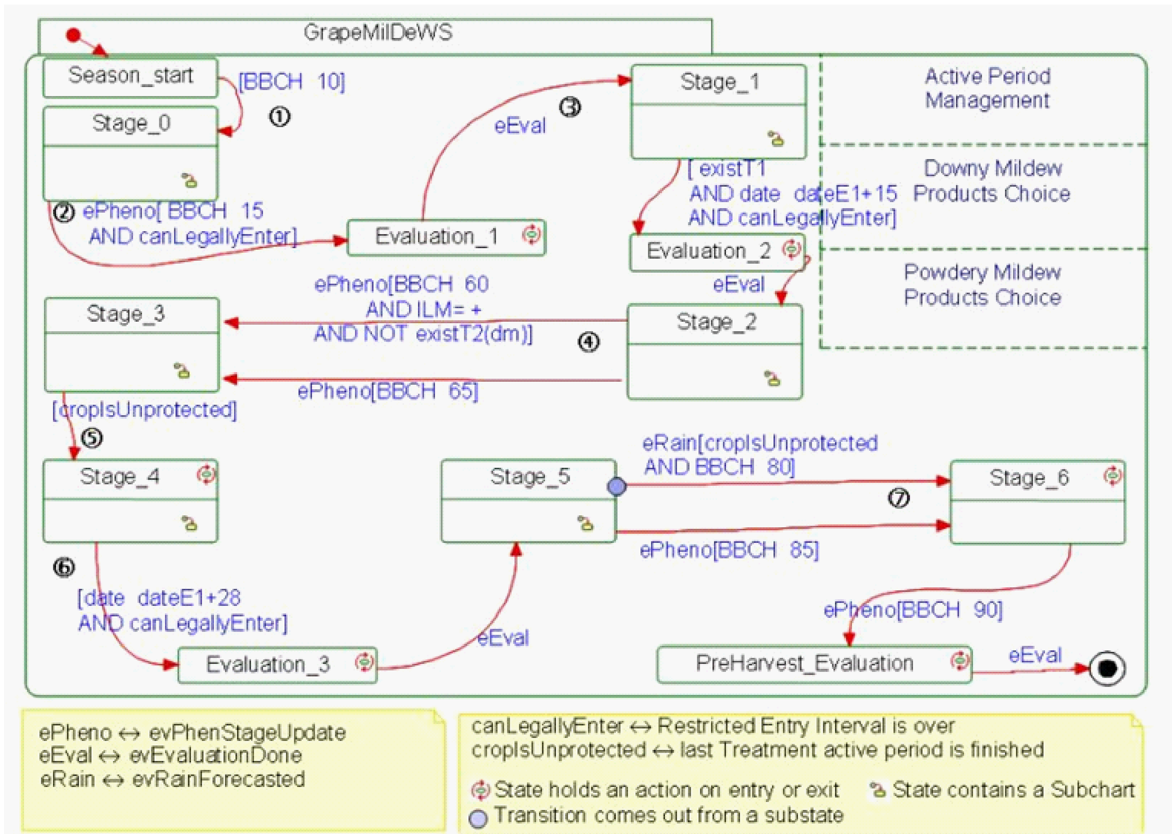


Figure 6. GrapeMilDeWS main process: 7 treatment decision stages and 3 observation states.



the decision problem at stake. After a description of the top level view of the main process, which represents the general organisation of the sequence of tactical decisions and the constraints controlling their timing, we will detail three treatment decision stages[6]. Stage_0's presentation will introduce the GrapeMilDeWS Statechart vocabulary (how we modelled our ideas into Statechart features). With Stage_4, we will present the general tactical decision logic of a stage. We will conclude the section with Stage_2, and show how using the Statechart language provides a better description of the required actions than the initial tabular format that the phytopathologists were using.

Main Process Overview

GrapeMilDeWS' top level Statechart in Figure 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. 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 "T*x*" 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 for that target 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 bunches and leaves.

The season starts with a monitoring as the first leaves unfold ([BBCH>10] tag ① in Figure 6). References to phenological stages in the diagram are given in the BBCH scale (Lorenz et al., 1995). The system remains in Stage_0 until the phenology of the plot has developed to at least 5 leaves unfolded (tag ② in Figure 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 6.

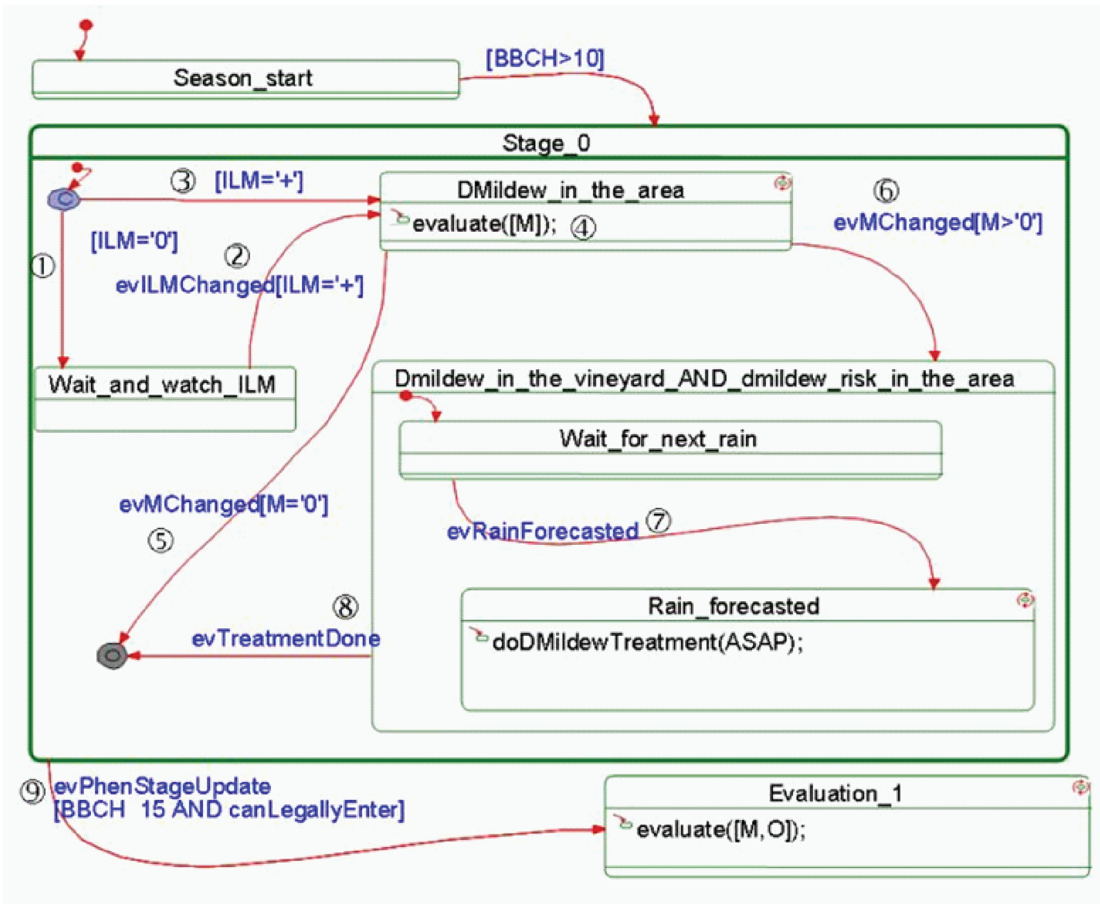
The active state will remain in Evaluation_1 until the observation of the plot has been carried out, notified (evEvaluationDone) and the neighbourhood variables O and M have been updated (③ Figure 6). After that, Stage_1 is entered.

Stage_1 lasts for two weeks past Evaluation_1 ("E1"). It is required to carry out a powdery mildew treatment within Stage_1. 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 managed in the stage's substatechart. The typical phenology for Stage_1, should be between '5/7' unfolded leaves (Evaluation_1 done) to '8/10' leaves. Phenology being quite difficult to determine precisely, the designers have chosen to use a fixed time period of 15 days instead, which is consistent with the desired phenological development in the Bordeaux area where GrapeMilDeWS was experimented (Léger, 2008, pp. 162-187). 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 ("E2") targets the same organs as Evaluation_1. Upon completion of this observation, Stage_2 has information on whether the first stage has efficiently controlled the beginning of each epidemic or, if new symptoms are still surging. If the epidemic level of any of the two diseases is worrying, Stage_2 calls for an optional treatment in order to safely reach mid flowering. In our Bordeaux conditions the typical duration of Stage_2 is again approximately two weeks.

The objectives of the early observations "E1" and "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 bunches. This early detection mechanism allows, when required, to "break" with treatments the dynamics of the epidemics before if reaches the so-called "explosive phase" (under the Vanderplanck theory, Segarra, Jeger, & van den Bosch, 2001).

Stage_3 (④ Figure 6) manages the flowering period. Depending on downy mildew local information (ILM) and occurrence of treatment during Stage_2, Stage_3 is entered either at mid-flowering or early flowering. Stage_3 simply triggers the third treatment: "T3". This joint application is *the key mandatory treatment* in the GrapeMilDeWS program. It targets both powdery and downy mildews because flowering is the most critical period in the season: the fruiting zone is then most susceptible and the year's revenue of the plot is at stake. Stage_3 ends when the shortest active period (AP) of the 2 product used for "T3", has elapsed (i.e. ⑤ Figure 6, the function *cropIsUnprotected* becomes true). At that time, the berries are at pea size. There is no evaluation

Figure 7. GrapeMilDeWS stage_0



of the epidemics in the field between Stage_3's exit and Stage_4's entrance.

No mandatory treatment is required at Stage_4. Any spraying that may be ordered during that stage is based on the values of O and M recorded during the first two evaluations. Stage_4 is designed to give extra security in years of high epidemic pressure. Such disease scenarios are detected during "E1" and "E2".

Evaluation_3 (⑥ Figure 6) is ordered 28 days after "T3", provided REI has elapsed after the optional "T4". Evaluation_3 differs from the two previous evaluations. Powdery mildew's monitoring is then aimed at the grape bunches. Downy mildew's monitoring remains focused on the leaves because downy mildew dries up the flowers or young fruits which then fall and thus cannot be used to estimate the disease level. At the time of "E3", bunches are beginning to close and there is no more downy mildew symptoms left on bunches. Evaluation_3 provides an early estimate of the sanitary status of the grape (before harvest) as well as support to decide on the opportunity for one more optional application.

The sprayings that may be ordered during Stage_5 are based solely on the indication acquired during Evaluation_3, ILM is not included in the decision process.

Stage_6 (⑦ Figure 6) consists in a final mandatory treatment against downy mildew, positioned during the first half of ripening. Bunches are then neither susceptible to powdery nor to downy mildew, but the aging leaves can be destroyed by downy mildew. Therefore T6 is applied to ensure that 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 last evaluation is meant to assess the overall quality of the crop protection and leads to no spraying decisions.

Stage 0

The substatechart of Stage_0 is shown in Figure 7. All tags in the following section refer to Figure 7.

This early monitoring stage starts as soon as the first leaves have 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 be set to '+' when the first symptoms of downy mildew are found within a range of 10 to 25 km around the vineyard and bioclimatic models indicate risk. This information is provided by the plant protection service advisory bulletins.

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. ②When 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, ⑦when 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 state ⑧. The final state indicates that no further activity will be carried out by the system while it remains in Stage_0.

When the field's stocks have developed 5 leaves, 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*≥15…) and it is safe to enter the plot (…*canLegallyEnter*]) (i.e. the REI must have elapsed). Entry in Evaluation_1 generates the order for a field evaluation of both powdery and downy mildew symptoms level on the leaves.

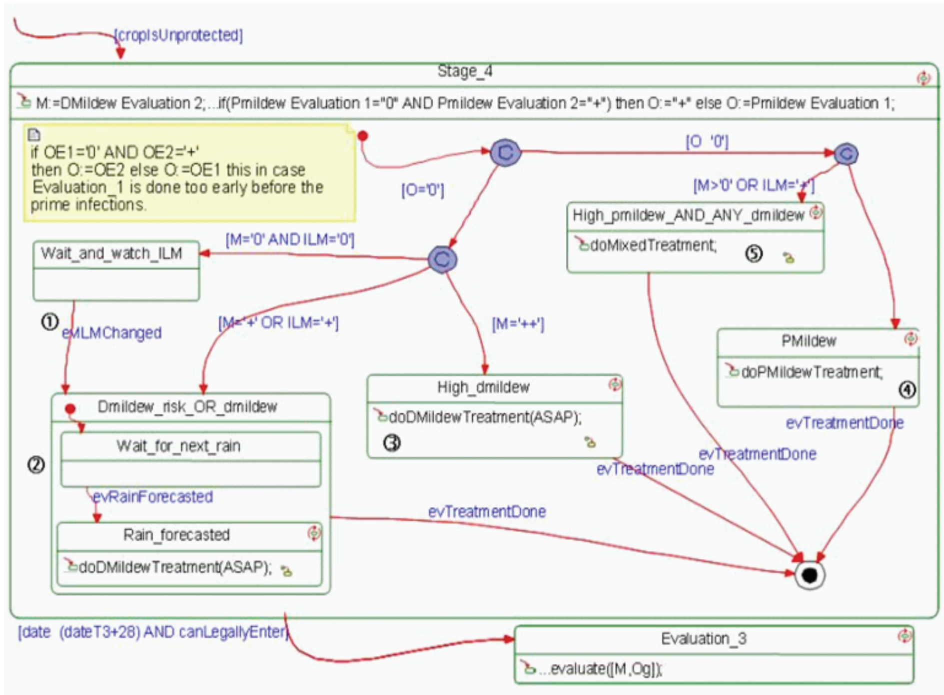
We have seen with Stage_0 how static decision logic (*[ILM='+']* or *[ILM='0']*) is interleaved with dynamic behaviours such as rain monitoring, which leads to a treatment application upon forecast of a rain event.

Stage_0 is rather simple because it is not necessary then to manage powdery mildew. Nevertheless, Stage_0 introduces the two tactical management features of the downy mildew:

- Wait and watch ILM when the epidemic risk is low and
- When the risk is greater or increasing, wait for a forecasted rain before doing any treatment against downy mildew.

We shall see hereafter that high downy levels on the plot change this typical behaviour of the decision system. The following stage is typical

Figure 8. GrapeMilDeWS stage_4



of combined management of the two different mildews.

Stage 4

The substatechart of Stage_4 is shown in Figure 8.

Stage_4 is entered once the active period (AP) of the third treatment ("T3" mandatory treatments at mid flowering) has ended. The goal of this stage is, during the years of intense epidemics (e.g. early powdery mildew), to protect the growth of the berries, when they are still green, growing and susceptible (pea size: *BBCH*≈73). This optional treatment stage should yield no treatment applications on low epidemics years. As for Stage_3, the variables O and M are not updated through an observation before entering Stage_4. Nonetheless, their values are refreshed according to the rules presented at the top of Figure 8[7].

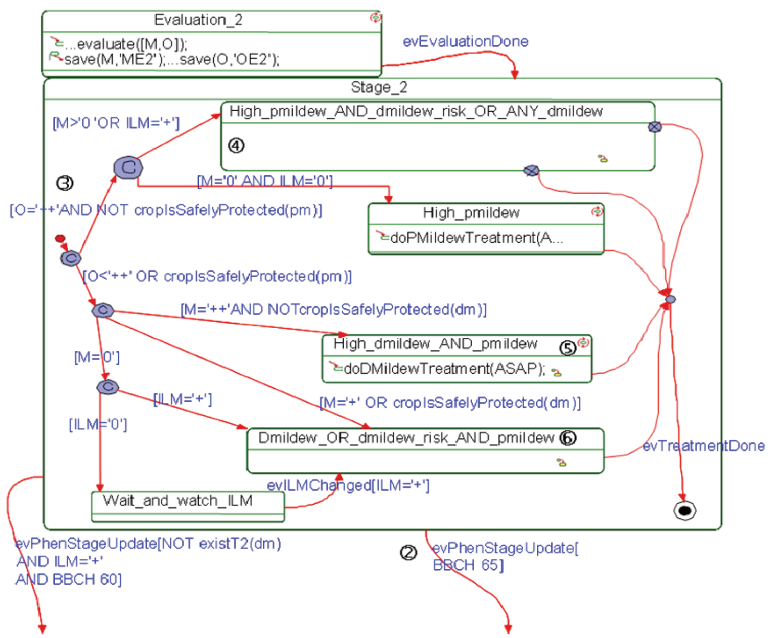
28 days after "T3", Stage_4 is followed by Evaluation_3. That is approximately 2 APs after "T3's" application. In the eventuality of a late "T4", exit may be postponed until the plot becomes accessible again.

Because Stage_4 was designed to yield at most one application in the field, it is most illustrative of the general logic of the decisions taken in GrapeMilDeWS (Except for the variables re-assignment). When the pre-flowering epidemics have been low (i.e. variables O='0' and M='0'), then the current value of ILM drives the decision. If ILM='0' then ILM shall be monitored ①. If downy mildew risks increases in the area, then the weather watch will be started ②. If only one of the field estimator is high, then only its target disease will be treated (tags ③ and 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.

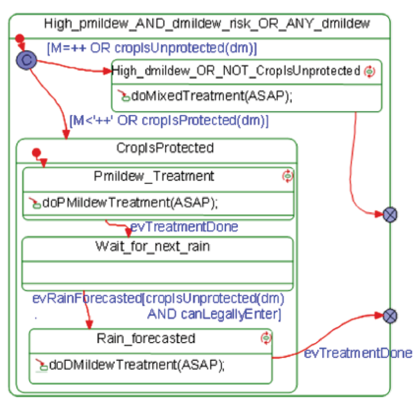
The other stages with optional treatment are adapted from this general framework to take into

Figure 9. GrapeMilDeWS stage_2 with substates

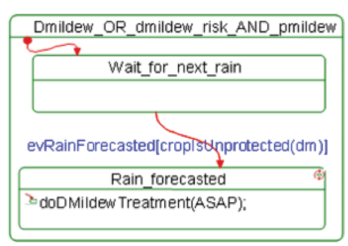
(a) *Stage_2 main substate*



(b)*Stage_2 High_pmildew_and_dmildew_risk_OR_ANY_dmildew*



(c)*Stage_2 Dmildew_OR_dmildew_risk_AND_pmildew*



account the desynchronisation of the previous treatments or to optimise the positioning of the following mandatory treatment. This will be illustrated with the case of the second stage.

Stage 2

The substatechart of Stage_2 is shown in Figure 9. All tags in the following section will refer to Figure 9(a).

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

Table 1. Stage_2 as first documented before the formal modelling process

E2 (O/M)	M='++'		M='+'		M='0'	
	ILM='+'	ILM= '0'	ILM='+'	ILM='0'	ILM='+'	ILM='0'
O ='0'	Dm 18/20		Dm if rain forecast			/
O ='+'	Pm&Dm 18/20					Pm 18/20

evolution of each disease's foliar epidemics observed during the second evaluation. "E2" allows a second assessment of the level of epidemics, 15 days after "E1". All treatments are optional at Stage_2 and may be spared if pressure of both epidemics is low.

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.

Stage_2 as presented in Figure 9 presents a rich dynamic behaviour. Initially, a tabular representation was used to document the early design of the Stages' tactical decision logic. Table 1 is the original tabular "decision model" of the second stage.

Consider in Table 1 the cell labelled "*Pm&Dm 18/20"* which specifies a treatment against both diseases between the phenological stages 18 and 20 (8 to 10 leaves). It corresponds to the conditions where Powdery mildew is high and downy mildew is not nil (i.e. *O='+'* and *(M>'0' OR ILM='+')*). This original recommendation has been abandoned after the introduction of the Active period as a decision variable. The AP variable is essential to the management of crop protection in viticulture. The pathologists were using it when experimenting early versions of GrapeMilDeWS, but did not have a satisfying way to represent the management of this information. However, it was clear that potential desynchronization during Stage_1 between powdery mildew treatment and downy mildew treatment, could lead to repeated applications and waste of pesticide if the active period was not explicitly included in the system. The Statechart formalism allowed to analyse the different scenarios and thanks to the AP management functions (e.g. *cropIsSafelyProtected* at③on Figure 9 (a)) adequate and non trivial decision states were designed (see tags ④, ⑤ and ⑥).

For example, Figure 9 (b) refines the conditions of the table example: *O='++'* and *(M> '0' OR ILM='+')*. When entering High_pmildew_AND_dmildew_risk_OR_ANY_dmildew a mixed treatment is ordered if powdery and downy mildew have been treated early during Stage_1 (i.e. long ago) or if the disease levels at evaluation_2 were high. Otherwise, the downy mildew conditions may allow sparing a treatment, therefore the treatments are disjoined. The powdery mildew treatment is carried out as soon as possible and the chance is taken that a second application for downy mildew may be needed.

DISCUSSION

Supporting Expert Elicitation as well as Biological Uncertainties

We chose Statechart for its intuitiveness. Based on higraphs (Harel, 1988), it is efficient for the visual representation of union and conjunction. Among the variety of Statechart dialects, we chose Rhapsody's (Harel & Kugler, 2004) because it is UML compliant. (Glinz, 2002) proposes another Statechart semantic built for readers unfamiliar with the notation (e.g. he proposes truth tables for complex triggering conditions). Our own experience is that experts need some time to get familiar with Statechart, but that they become able to represent both logical rules and sequentiality. The nesting capabilities help them focus on different matter at different scales. This is consistent with the results that have been obtained by other authors in their study of the ergonomics of the Statechart language (Cruz-Lemus, Genero, Manso, & Piattini, 2005).

Stage_2 is a good example that using a process modelling language has helped the pest management experts think, analyse and make explicit a decision system design. That decision system can (i) be interpreted without any ambiguities, and (ii) is a system that may spare even more treatments than initially imagined. Here, the formalism has enhanced creativity and efficiency.

In order to assess the relevance of the elicitation process that we carried out, we have tested, in simulation, that the model would produce the same decisions as the experts, doing so at the same time if exposed to the same experimental conditions. We present the methodology for these evaluations in (Léger, 2008, pp. 162-187). We found that 85% of the decisions were similar. We have also been able to highlight a few behaviours that the pathologists had not mentioned when elicited, thanks to the comparison of the timed sequences of their experimental decision with those of the simulated GrapeMilDeWS (Léger, 2008, pp. 162-187). For example, they anticipated the treatment application order before the entry conditions in Stage_4 were met (i.e. before the end of T3's AP) to alleviate their workload and optimize the application of T4 just at the end of T3's AP. Such behaviour cannot be reproduced by the GrapeMilDeWS Statechart model which is purely reactive.

These timing errors were due to over-specification of the model. A solution would be to abandon the deterministic semantics of standard Statechart in favour of another formalism that provide some indeterminism about the time a transition is crossed, e.g. *timed Statechart* (Graf, Ober, & Ober, 2003; Kesten & Pnueli, 1992; Maler, Manna, & Pnueli, 1991). In response to an event, it is possible in these formalisms to replace immediate transition crossing by a time interval or a deadline. This would allow better revealing the pathologists' uncertainties about timing. Because we work with biological material under field conditions, we cannot be certain of any exact date for a required action. Even if knowledge was not lacking, statistical uncertainties would justify the use of such timed formalism. The trade-off for this more adequate modelling formalism is the loss of the possibility to run the model as a complete controller (decision maker) in simulations. The model then becomes a specification of what the controller should do.

Yet, simulation is not the only way of verifying the proper behaviour of a discrete event system. With such formal modelling, it is possible to perform "Model Checking"(Katoen, 2004). Model checking is a set of techniques using modal logics to validate temporal (Pnueli, 1977) or even real time (Alur & Dill, 1994; Penczek & Pólrola, 2006; Yovine, 1993) behavioural properties of a system (i.e. its model) with the strength of a mathematical proof. This applies to the formal models of a system's specifications. By formal, we mean here that the model should have an equivalent formulation in a finite state automaton formalism. Such models can be model-checked even if not deterministic, i.e. even if they cannot be simulated. The common behavioural properties of a system that can be model checked are *reachability, vivacity* and *safety*. To check a system's safety is for example to control that a forbidden state of the system *can never* be activated.

Modelling Agricultural Decision Making with Formal Mathematical Languages

In the previous sections, we have argued for the use of the engineering approach in agriculture as proposed in (Coléno & Duru, 2005; Day, Audsley, & Frost, 2008). Building computerized simulation models for decision making is quite common in systems agronomy (e.g. J.-M. Attonaty, Chatelin, & Garcia, 1999; Cros et al., 2004), with object oriented frameworks such as OTELO (J. M. Attonaty et al., 1994) or GPFarm (Ascough II, Hanson, Shaffer, McMaster, & Deer-Ascough, 1997) used for strategic planning or to develop novel agronomical decision rules (e.g. Chatelin et al., 2005; Rellier et al., 1998). These systems have been mostly developed from an artificial intelligence perspective. While much emphasis was devoted to the conceptual aspects (Martin-Clouaire & Rellier, 2009), their implementation has been rather specific instead of taking advantage of advances in control theory and formal modelling of discrete event processes. With an appropriate combination of formal models, it is possible to choose between simulation techniques and model-checking, or even combine them, in order to design decision making processes.

Our work is a case that supports Harel's point that 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 (Harel, 2004). These tools are now used in the fields of systems biology (Webb & White, 2005), manufacturing (Baresi, Orso, & Pezzé, 1997; Castillo & Smith, 2002) and medical research (ten Teije et al., 2006). For instance, systems such as an immune system (Cohen, 2007) or a complete *Caenorhabditis elegans* nematode worm (Kam et al., 2003) have been modelled using standard modelling languages such as the UML (Zheng, 2006) and the resulting models integrate the available knowledge from the literature.

Formal Modelling as a Large Experiment Management Tool

The grapevine mildews pathosystem is still poorly understood. The consequence of this is that no optimal solution can be computed and bio-technical simulations cannot be used to assess performance of newly designed strategies in viticulture. We therefore have created GrapeMilDeWS in an incremental process, alternating design and modelling phases with field experiments. Experimentations are done at the plot scale (>0.5ha). They are carried out without untreated control areas, for these would bias the results of the experiments, through unmanaged epidemics. The crop protection performance (quantity and quality of the harvest) of a plot managed with GrapeMilDeWS is compared to the output obtained under the conventional protection carried out by the grower in a similar plot, using the same spraying equipment. The data from experiments provide insight about the validity of the architecture of the pest management strategy as well as on the thresholds of the tactical variables.

In the earlier stages of the design process, prior and during the modelling process, GrapeMilDeWS was experimented by its designers: the pathologists. Now that it has been made explicit, it is experimented on a large scale throughout France (2008 to 2010). One of the difficulties of such large experimental set up is to control the quality of the execution and to be able to compare heterogeneous results achieved in different regions with very different climatic conditions. The analysis of the harvest and sanitary results linked to the sequence of decisions using a process conformance methodology (Léger & Naud, 2009; Rozinat & van der Aalst, 2008), gives insights to improve the model while controlling experimental artefacts. That is why we believe that in addition to its knowledge transfer qualities, formal modelling provides tools to manage large scales experiments.

In this respect too, it seems to us that model checking methodologies and algorithms for pest management are a priority for further research. Model checking, by automating the quality control of the decision system's internal behaviour, would allow to further focus the necessary field experiments on performance validation. Some facts about the 2008 campaign support this research perspective. We have been confronted with a sequence of event which did not leave people enough time to react, this caused some of the plots to fail their production target at harvest (Delière et al., 2008). With such costly experimental set up, it would be best not to waste data by discovering process bugs during the experiments. These should be devoted to validating the agronomic performances of the system.

CONCLUSION: A PRESCRIPTIVE DESIGN AS THE BACKBONE OF A FUTURE DSS

On the research track of designing a DSS that would help the grower manage the targeted pests with less fungicide treatments, we started by building a decision system, in a prescriptive approach. During the tests on individual real-scale plots, GrapeMilDeWS provided 40 to 60% reductions in the number of treatments, and seems therefore to be an appropriate backbone for a future DSS. We applied a method that is similar to the five-step method presented in (Debaeke et al., 2008). Our initial motivation for this incremental method is that there is an urgent need for innovation in viticulture. One should remember that control of grape powdery and downy mildew traditionally leads to 7 and 6 annual treatments on average in France. It is also worth recalling that French regulations forbid the use of resistant cultivars for wine with a registered designation of origin. Then, when designing GrapeMilDeWS, we focused on rationalizing the tactical organisation of pest management. Having to work on these tactical aspects of pest management, we adopted the formal tools that we present in this article. We also realised that little work had actually been done so far to capture the operational aspects of pest management by modelling.

In agriculture, simulation has become very popular to study the farmer's decision processes and to design innovative cropping systems. These approaches usually link a decision or crop management simulator with a biophysical simulation model. As we lack a realistic biophysical model of the multi-pathosystem of vine-powdery & downy mildews, we had to validate the biological efficiency of GrapeMilDeWS through field experiments. Because such experiments are expensive and should last several years in order to cover most of the bioclimatic cases, our idea is that candidate processes should be made "bug free" from an informational and logistical point of view before going out to the field. A year of data at some plots can be wasted if this is not done. We demonstrated in our case that Statecharts could be an appropriate language for the modelling and we derived well-formed guidelines from the model that were used for experimenting in many plots of different regions. We have reasonable arguments to claim that timed formalisms such as timed Statecharts would be even more appropriate. We also believe that applying model checking to decision in crop protection and other fields of agronomy, is a promising research perspective. Results obtained in checking medical guidelines (ten Teije et al., 2006) are also comforting this perspective.

Furthermore, we think that implementing IPM and drastically reducing pesticide use requires a high degree of technicality, and sometimes involves accepting more risks. Adopting a system which reduces the number of treatments by at least 40% could be seen by professionals as a leap of faith, a perilous change. Indeed, there is still little knowledge on how the vineyard pest control system behaves under these low fungicides input conditions. By building a prescriptive model like GrapeMilDeWS, pathologists of the team have contributed to signpost a safe path in this unknown territory.

The design of such decision processes, or well-formed guidelines, can greatly benefit from a formalism that helps the pathologist to visualise the potential behaviours of his design. We have shown (Léger & Naud, 2009) that elicitation using Statechart is efficient in this respect. We acknowledge that an accurate pathosystem simulator would make the design simpler. Yet, some authors also have used a graphical modelling formalism to elicit the dataflow of decision making in precision agriculture at different scales (Fountas, Wulfsohn, Blackmore, Jacobsen, & Pedersen, 2006). They used a graph to present the elicited information to the interviewed "experts" for validation. They report that the experts appreciated the graphical format and that formalising the dataflow gave them greater insight into their decision process. This data centric approach could actually complement our work in providing a better and more formal representation of the indicators used in the DeWS.

In conclusion, we started this work with the idea that a radical change in practices was needed. However we discovered that although the general knowledge is available, it is not put into effect by the growers. Therefore, GrapeMilDeWS was developed as a pedagogical tool to transfer the knowledge in a more operational format. The underlying knowledge encapsulated in the process is simple enough for any grower to learn. In fact, the information used in the DeWS is readily available to many growers; GrapeMilDeWS is mostly about managing the timing and use of this "common knowledge". It could very well be enacted without a computer aid, provided that the number of plots managed remains small. However, the best (economic and environmental) results from implementing GrapeMilDeWS will be achieved when the decisions will be made at the plot scale, while the logistics of spraying would remain nevertheless organised at the farm scale. However, management of all these concurrent decision processes will become tedious without a computer program. Furthermore, as we plan to move to timed formalisms and manage anticipations, we then need to manage the revision of decision when something changes (e.g. rain forecast), and a computer program would allow to trace this. Therefore, the Decision support system we aim at is some kind of Workflow Management System (WfMS) that would take as input any DeWS model along with a farm description. Its output would help the grower plan and manage his crop protection throughout the whole farm.

REFERENCES

Alur, R., & Dill, D. L. (1994). A Theory of Timed Automata. *Theoretical Computer Science*, *126*(2), 183–235.

Ascough, J. C. II, Hanson, J. D., Shaffer, M. J., McMaster, G. S., & Deer-Ascough, L. A. (1997). *Great Plains Framework for Agricultural Resource Management (GPFARM): A decision support system for whole farm/ranch strategic planning* (*Vol. 2*). Paper - American Society of Agricultural Engineers.

ASK, B. M. I. (2000). Enquête sur les exploitations viticoles et leurs pratiques phyto-sanitaires. *Onivin-Info., 78*.

Attonaty, J.-M., Chatelin, M.-H., & Garcia, F. (1999). Interactive simulation modeling in farm decision-making. *Computers and Electronics in Agriculture*, *22*(2-3), 157–170.

Attonaty, J. M., Chatelin, M. H., Poussin, J. C., & Soler, L. G. (1994). OTELO: un simulateur a base de connaissance pour raisonner equipement et organisation du travail. In Le conseil en agroequipement et en organisation du travail. Outils et premières experiences (Vol. 66, pp. 37-49): APCA-Assemblee Permanente des Chambres d'Agriculture (FRA).

Aubry, C., Loyce, C., Meynard, J. M., Chatelin, M. H., Rellier, J. P., & Verjux, N. (1997). *Outils et méthodes pour la conception et l'évaluation d'itinéraires techniques*. Paper presented at the Colloque du cinquantenaire de l'INRA.

Baresi, L., Orso, A., & Pezzé, M. (1997). *Introducing formal methods in industrial practice*. Paper presented at the 20th international conference on software engineering.

Boller, E. F., Avilla, J., Gendrier, J. P., Jörg, E., & Malavolta, C. (1998). Integrated plant protection in the context of a sustainable agriculture. *IOBC/WPRS Bulletin*, *21*(1).

Booth, T. L. (1967). *Sequential machines and automata theory*. New York: Wiley.

Castillo, I., & Smith, J. S. (2002). Formal modeling methodologies for control of manufacturing cells: Survey and comparison . *Journal of Manufacturing Systems*, *21*, 40–57.

Chatelin, M. H., Aubry, C., Poussin, J. C., Meynard, J. M., Masse, J., & Verjux, N. (2005). DeciBle, a software package for wheat crop management simulation. *Agricultural Systems*, *83*(1), 77–99.

Cohen, I. R. (2007). Real and artificial immune systems: computing the state of the body. *Nature Reviews. Immunology*, *7*(7), 569–574.

Coléno, F., & Duru, M. (2005). L'apport de la gestion de production aux sciences agronomiques. Le cas des ressources fourragères. *Nature Sciences Sociétés*, *13*, 247–257.

Cros, M. J., Duru, M., Garcia, F., & Martin-Clouaire, R. (2004). Simulating management strategies: the rotational grazing example. *Agricultural Systems*, *80*(1), 23–42.

Cruz-Lemus, J., Genero, M., Manso, M., & Piattini, M. (2005). Evaluating the Effect of Composite States on the Understandability of UML Statechart Diagrams . In *Model Driven Engineering Languages and Systems* (*Vol. 3713*, pp. 113–125). Berlin, Germany: Springer verlag.

Damm, W., Josko, B., Votintseva, A., & Pnueli, A. (2003). *A formal semantics for a UML kernel language* (No. part 1 of the deliverable D1.1.2). grenoble: Project IST-2001-33522 OMEGA.

Day, W., Audsley, E., & Frost, A. R. (2008). An engineering approach to modelling, decision support and control for sustainable systems. *Philosophical Transactions of the Royal Society B-Biological Sciences*, *363*(1491), 527–541.

Debaeke, P., Munier-Jolain, N., Bertrand, M., Guichard, L., Nolot, J., Faloya, V., et al. (2008). Iterative design and evaluation of rule-based cropping systems: methodology and case studies. A review. *Agronomy for Sustainable Development, 50*(preprint).

Delière, L., Cartolaro, P., Naud, O., Léger, B., Goutouly, J. P., Davidou, L., et al. (2008). Conception et évaluation de Mildium, un processus opérationnel de décision pour une gestion fongicide coordonnée à apport réduit. *PHYTOMA - La Défense des Végétaux, 621*, 20-24.

Dent, D. (1995). *Integrated Pest Management*. Chapman & Hall.

Feldman, R. M., & Curry, G. L. (1982). *Operations research for agricultural pest management* (pp. 601–618). Operations Research Society of America.

Fountas, S., Wulfsohn, D., Blackmore, B. S., Jacobsen, H. L., & Pedersen, S. M. (2006). A model of decision-making and information flows for information-intensive agriculture. *Agricultural Systems*, *87*(2), 192–210.

Girard, N., & Hubert, B. (1999). Modelling expert knowledge with knowledge-based systems to design decision aids: the example of a knowledge-based model on grazing management. *Agricultural Systems*, *59*(2), 123–144.

Glinz, M. (2002). *Statecharts For Requirements Specification – As Simple As Possible, As Rich As Needed*. Paper presented at the ICSE International Workshop on Scenarios and State Machines: Models, Algorithms and Tools. Retrieved from http://www.cs.tut.fi/~tsysta/ICSE/Program.html

Graf, S., Ober, I., & Ober, I. (2003, Oct.). *Timed annotations in UML*. Paper presented at the Workshop on Specification and Validation of UML models for Real Time and Embedded Systems (SVERTS 2003), a satellite event of UML 2003, San Francisco.

Harel, D. (1987). Statecharts: A visual formalism for complex systems. *Science of Computer Programming*, *8*(3), 231–274.

Harel, D. (1988). On visual formalisms. *Communications of the ACM*, *31*(5), 514–530.

Harel, D. (2004). A Grand Challenge for Computing: Towards Full Reactive Modeling of a Multicellular Animal . In *Verification, Model Checking, and Abstract Interpretation* (pp. 323–324). Berlin, Heidelberg: Springer.

Harel, D., & Gery, E. (1996). Executable object modeling with statecharts. In *Proceedings of the 18th international conference on Software engineering* (pp. 246-257). Berlin, Germany: IEEE Computer Society.

Harel, D., & Kugler, H. (2004). The RHAPSODY semantics of statecharts (or, on the executable core of the UML) - (Preliminary version) . In *Integration of Software Specification Techniques for Applications in Engineering* (*Vol. 3147*, pp. 325–354). Springer-Verlag.

Kam, N. A., Harel, D., Kugler, H., Marelly, R., Pnueli, A., Hubbard, E. J. A., et al. (2003). Formal Modeling of C. elegans Development: A Scenario-Based Approach. In *Proceedings of the First International Workshop on Computational Methods in Systems Biology* (Vol. 2602, pp. 4-20): Springer-Verlag.

Katoen, J. (2004). *Principles of model checking*. Enschede, The Netherlands: University of Twente.

Kesten, Y., & Pnueli, A. (1992). Timed and hybrid statecharts and their textual representation. *Formal Techniques in Real-Time and Fault-Tolerant Systems*, 591–619.

Kogan, M. (1998). Integrated pest management: Historical perspectives and contemporary developments. *Annual Review of Entomology*, *43*, 243–270.

Léger, B. (2008). *Recueil et Formalisation de procédés experts pour conduire une protection intégrée du vignoble.* (Unpublished PhD Thesis), Supagro Montpellier (France).

Léger, B., & Naud, O. (2009). Experimenting statecharts for multiple experts knowledge elicitation in agriculture. *Expert Systems with Applications*, *36*, 11296–11303.

Lorenz, D. H., Eichhorn, K. W., Bleiholder, H., Klose, R., Meier, U., & Weber, E. (1995). Phenological growth stages of the grapevine (Vitis vinifera L. spp. vinifera) - codes and descriptions according to the extended BBCH scale. *Australian Journal of Grape and Wine Research*, *1*(2), 100–103.

Maggiolo-Schettini, A., Peron, A., & Tini, S. (2003). A comparison of Statecharts step semantics. *Theoretical Computer Science*, *290*(1), 465–498.

Mahaman, B. D., Passam, H. C., Sideridis, A. B., & Yialouris, C. P. (2003). DIARES-IPM: a diagnostic advisory rule-based expert system for integrated pest management in Solanaceous crop systems. *Agricultural Systems*, *76*(3), 1119–1135.

Maler, O., Manna, Z., & Pnueli, A. (1991). *From timed to hybrid systems.* Paper presented at the REX Workshop, Mook, NL.

Martin-Clouaire, R., & Rellier, J. P. (2003). *Modélisation et simulation de la conduite d'un système de production agricole*. Paper presented at the 4 ème Conférence de Modélisation et Simulation-Organisation et Conduite d'Activité dans l'Industrie et les Services (MOSIM'03).

Martin-Clouaire, R., & Rellier, J. P. (2009). Modelling and simulating work practices in agriculture. *International Journal of Metadata . Semantics and Ontologies*, *4*(1), 42–53.

Nash, E., Dreger, F., Schwarz, J., Bill, R., & Werner, A. (2009). Development of a model of data-flows for precision agriculture based on a collaborative research project. *Computers and Electronics in Agriculture*, *66*(1), 25–37.

OMG. (2007). *Unified Modelling Languages 2.0*. Retrieved from www.uml.org

Penczek, W., & Pólrola, A. (2006). *Advances in Verification of Time Petri Nets and Timed Automata, a temporal logic approach* (*Vol. 20*). New York: Springer.

Pnueli, A. (1977). *The Temporal Logic of Programs.* Paper presented at the FOCS.

Rellier, J. P., Martin-Clouaire, R., Navarrete, M., Jeannequin, B., Gary, C., & Montbroussous, B. (1998). Modelling and simulating decision making for greenhouse tomato production: the conserto project . In Marcelis, L. F. M. (Ed.), *Acta Hort* (*Vol. 456*, pp. 485–492). ISHS.

Rozinat, A., & van der Aalst, W. M. P. (2008). Conformance checking of processes based on monitoring real behavior. *Information Systems*, *33*(1), 64–95.

Sebillote, M. (1987). Du champ cultive aux pratiques des agriculteurs. Reflexions sur l'agronomie actuelle. *C. R. Académie d'Agriculture Française*, *73*(8), 69–81.

Sebillote, M., & Soler, L.-G. (1988). Le concept de modèle général et la compréhension du comportement de l'agriculteur. *C. R. Académie d'Agriculture Française*, *74*, 59–70.

Sebillotte, M. (1978). Itinéraires techniques et évolution de la pensée agronomique. *Comptes rendus de l'Académie d'agriculture de France, 11*, 906-914.

Segarra, J., Jeger, M., & van den Bosch, F. (2001). Epidemic Dynamics and Patterns of Plant Diseases. *Phytopathology*, *91*(10), 1001–1010.

Shoemaker, C. A. (1984). The optimal timing of multiple applications of residual pesticides: deterministic and stochastic analyses . In Conway, G. R. (Ed.), *Pest and pathogen control: strategic, tactical and policy models* (pp. 290–309). Chichester, UK: John Wiley & Sons.

SRPV-Aquitaine. (2007, July). Avertissements Agricoles. Retrieved Oct, 2007, from http://www.srpv-aquitaine.com/_publique/default_publique.asp

Steinberger, G., Rothmund, M., & Auernhammer, H. (2009). Mobile farm equipment as a data source in an agricultural service architecture. *Computers and Electronics in Agriculture*, *65*(2), 238–246.

ten Teije, A., Marcos, M., Balser, M., van Croonenborg, J., Duelli, C., & van Harmelen, F. (2006). Improving medical protocols by formal methods. *Artificial Intelligence in Medicine*, *36*(3), 193–209.

van der Aalst, W., & van Hee, K. (2002). *Workflow Management: Models, Methods, and Systems*. Cambridge, MA: MIT Press.

von der Beeck, M. (1994). *A comparison of statecharts variants.* Paper presented at the Formal Techniques in Real Time and Fault Tolerant Systems.

Webb, K., & White, T. (2005). UML as a cell and biochemistry modeling language . *Bio Systems*, *80*, 283–302.

Wolfert, J., Matocha, D., Verloop, C. M., & Beulens, A. J. M. (2009). *Business Process Modeling of the Pesticide Life Cycle: a service-oriented approach*. Paper presented at the Joint International Agricultural Conference. Retrieved from http://de.scientificcommons.org/50724910

Yovine, S. (1993). *Méthodes et outils pour la vérification symbolique de systèmes temporisés.* (Unpublished PhD Thesis), Institut National Polytechnique de Grenoble, Grenoble (France).

Zadoks, J. (1989). EPIPRE, a computer-based decision support system for pest and disease control in wheat: Its development and implementation in Europe. *Plant Disease Epidemiology*, *2*, 3–29.

Zheng, W. J. J. (2006). Engineering approaches toward biological information integration at the systems level. *Current Bioinformatics*, *1*(1), 85–93.

ENDNOTES

1. French National Institute for Agricultural Research
2. See implementation details in (Léger, 2008, pp. 162-187)
3. 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.
4. The active period is the period during which the product efficiency is "guaranteed", and after which we consider the plot has become susceptible again
5. This is Rhapsody object Statechart semantics; it requires the object which behaviour is described by the Statechart to have the focus. While the Statechart owning object has focus, 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.
6. A complete presentation of the model can be found in (Léger, 2008, pp. 127-155)
7. 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 "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