Workflow modelling for experimenting a decision system on protection of grapevine against mildews

O. Naud, A. Fofack-Kitio, B. Léger

▶ To cite this version:

O. Naud, A. Fofack-Kitio, B. Léger. Workflow modelling for experimenting a decision system on protection of grapevine against mildews. AgEng 2010, International Conference on Agricultural Engineering, Sep 2010, Clermont-Ferrand, France. 9 p. hal-00804948

HAL Id: hal-00804948

https://hal.science/hal-00804948

Submitted on 26 Mar 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Workflow modelling for experimenting a decision system on protection of grapevine against mildews

O.Naud[1], A.Fofack-Kitio[1], B.Leger[2]

1/ Cemagref – UMR ITAP – 361 rue Jean François Breton 34196 Montpellier France

2/ Arvalis Institut du végétal - Pôle Ecophysiologie - Chemin de la petite minière 78280 Guyancourt, France.

Abstract:

Works initiated in 2001 by our partners phytopathologists at INRA Bordeaux led to the design of a decision system against both powdery and downy mildews of grapevine, which has been named “GrapeMilDeWS”, in French “POD Mildium”. GrapeMilDeWS is decomposed in 7 stages distributed all along the vegetative season. Each leads basically to two decisions made at the plot scale: whether or not to spray against each of the two diseases. Our contribution to system design has been on knowledge elicitation and formal modelling, with Statecharts. Statecharts describe processes with states and events, and are a formal construct built upon finite state automata. Details and complex time behaviour can be described thanks to hierarchy and concurrency mechanisms. The modelling was undertaken in order to produce a fully explicit and logically consistent decision protocol, so that many users would test it on many different plots, in different regions. The protocol now exists in a written form and has been tested on a network that has grown up to 35 plots in 2009.

Experimenting on a such a large scale, and getting back data and knowledge from this, causes numerous new challenges: (i) the decision system has to be robust to various conditions, while avoiding two loopholes: too much flexibility in the specification with risk of inappropriate implementation, or too many details (ii) the data to collect are numerous and include weather forecasts that are evolving by nature and need to collected in a timely manner; (iii) people involved are numerous: regional disease risk analysts, decision-makers for each plot, advisers who provide assistance to decision-makers and check conformance of decision to the original protocol.

We have thus undertaken to develop an experimentation support system that is based on workflow nets by van Der Aalst (YAWL framework). Workflow nets are derived from Petri nets, which like Statecharts are based on processes with events and states. The emphasis of Petri nets is on management of resources, represented by tokens, and control of concurrent process. With workflow nets, tokens represent cases to handle. The workflow system allows to manage processes and associated data in a consistent manner. The paper is about how GrapeMilDeWS, which we call a Decision Worklow, is being implemented in YAWL. The advantages of workflows nets for this application are discussed in regards to alternatives like Statecharts and Colour Petri Nets.

Keywords: Formal Modelling, Decision Support Systems, Crop Protection

Introduction

In 2001, phytopathologists at INRA Bordeaux undertook to design and experiment a series of guidelines, which they called “decision rules”, to handle grapevine diseases. This work was started within the French research network initiative on Integrated Crop Protection (or Integrated Pest management - IPM). One of these “decision rules” concerned management of downy mildew, another concerned powdery mildew. Both were designed as crop protection strategies with low fungicide input and significant information input from surveys in the plot. For downy mildew, other input data were weather forecasts and local bioclimatic risk. In 2005, they drafted new guidelines for managing both diseases in the same decision process. Combining the reasoning about downy and powdery mildews is consistent with practices of growers, who often mix specialties against both diseases in the same spraying operation. We started in the same time to collaborate to this research with computer and automation modelling techniques (Léger, 2008). The modelling work had three purposes: facilitate the design of a decision process, check consistency and completeness of the design, ease communication between designers and people in charge of experimenting the decision process.

The result of this collaboration is GrapeMilDeWS, where DeWS stands for Decision Workflow System and GrapeMilDeWS stands for Grape Mildews DeWS. GrapeMilDeWS is alternatively named “POD Mildium” in France. The elicitation of knowledge for GrapeMilDeWS with the modelling language Statecharts is reported in (Léger, 2009). The resulting model is explained in detail in (Léger et al, 2010a).

The purpose of this paper is to explain why and how to move from a DeWS modelled in the Statecharts language to a management system for this DeWS, based on the workflow language YAWL (for Yet Another Workflow Language; van der Aalst et al, 2010).

Decision Workflow

GrapeMilDeWS is about how to decide when to spray and when not to spray for powdery and downy mildews of grapevine. These diseases are polycyclic: untreated, the pathogen agents perform a number of contamination-latency-activity-sporulation and further contamination cycles during the season. Infecting leaves and young berries, they can potentially destroy the yield of a year. Roughly said, the current infections cannot be really cured, and the way to break or slow down the cycles is to spray preventively to further contaminations and infections. Yet, there is social demand in favour of diminishing the quantity of sprayings (Léger et al, 2010b; Bonicelli et al, 2010). Optimising the spraying process itself in order to get all fungicide on target, and according to the quantity of vegetal to protect, is indeed relevant. Besides, the potential savings in quantities sprayed, by adapting dosage and timing of sprayings, is very much dependent on clever tactics. Let us illustrate this by sample information about each disease.

Tactical ingredients for managing the mildews

Once primary contaminations have started, the propagation of downy mildew within a plot is due to rainy events. Each fungicide is labelled with an Active Period (AP) in days. This duration is calculated after normalised trials where treatment is repeated periodically. In fact, most products do not have a constant efficiency during this Active Period. One obvious reason is that vine is growing. Systemic products dilute. So, for a heavily contaminating rain, there is better protection if the previous treatment is not at the very end of its Active Period. Choosing the appropriate timing for spraying according to rain forecasts is thus a useful tactical ingredient for managing downy mildew.

As for powdery mildew, the number of contaminations on each cycle is dependent on the number of previously infected and sporulating sites. The dynamics of typical epidemics follow “logistic curves”. It is therefore quite obvious that these epidemics are more easily managed by early actions. Furthermore, it has been shown by experiments that early epidemics are often the most severe. Yet, one has to optimize quantities and define when and how much to spray. Spraying too early may be a waste of chemicals and spraying too late may reduce or destroy the yield. Indeed, early symptoms of powdery mildew are located on the rear-side of leaves and very inconspicuous (with the exception of the “flag-shoots” for certain cultivar like Carignan). The strategy of GrapeMilDeWS is to spend observation time in fields by checking a number of leaves, twice before flowering as the flowering period is the most critical for the yield. Tactics corresponds to the careful positioning in time of observations, sampling methods and thresholds, in order to discriminate best the cases of low epidemics from the severe ones (Léger et al., 2007).

Describe tactics by sequences of tasks and information events

The timeline for describing the tactics for crop protection has two clocks. The first follows the annual calendar in days, because the Active Periods are specified in days, and the resources, human as well as equipment, are managed on this basis, and so are the weather forecasts. The second clock provides the phenological status of the vine, that is the mediane growth stage of vineyard. This clock tics from bud-break to ripening and maturity, via the critical and short period of flowering. The designers of GrapeMilDeWS have chosen to breakdown the season in 7 management stages, where each stage leads to at most one treatment per disease. Within each stage, there are tactics to appropriately place in time sprayings and information retrieval according to indicators. This is why we chose Statecharts (Harel, 1987) for eliciting and modelling GrapeMilDeWS (Léger, 2009). The hierarchical construction provides abstraction features. It is possible during one interview to focus on one part of the Statecharts or another, to enhance details progressively during the series of interviews. The diagrams depict clearly both sequential and logical aspects of the process, as shown in the example of stage 0 on Figure 1. The original exhaustive set of Statechart diagrams of GrapeMilDeWS as of 2007 can be found in (Léger, 2008).

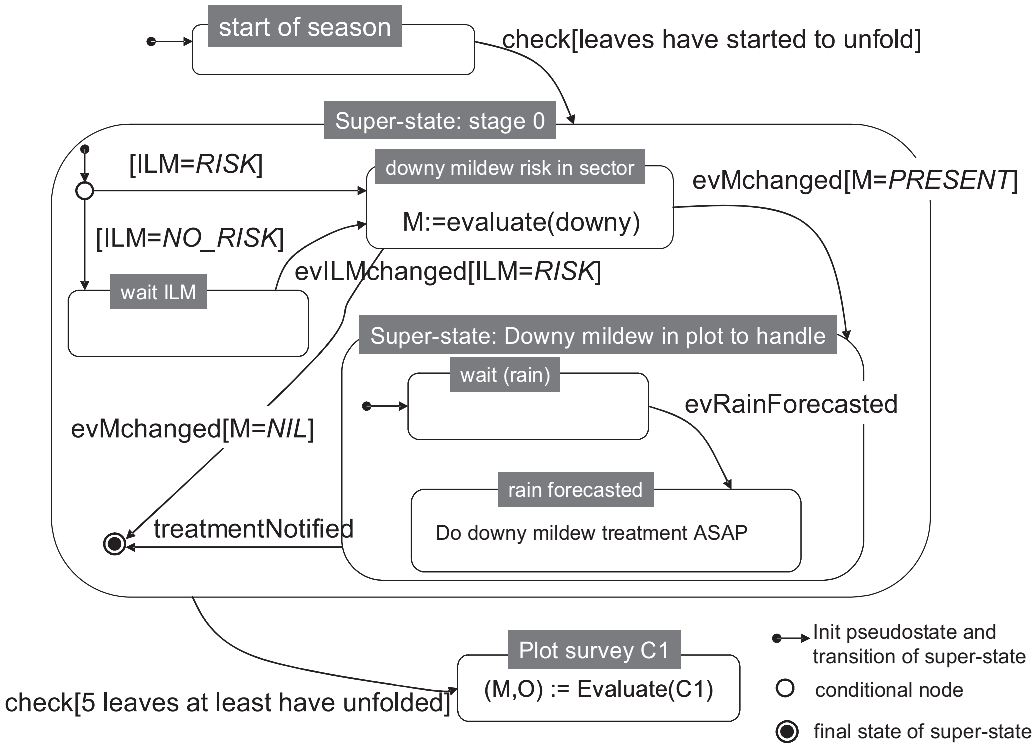

Figure 1 Sample from GrapeMilDeWS: stage 0

About notations for transitions (arcs from one state to another) in Figure 1, please note that := represents variable assignment and = is the equality operator. The path that is taken depends on two variables ILM and M, which have qualitative values (integer or boolean codes). It also depends on the occurrence of two events: an update in ILM, and forecast of a rain. ILM is about risk of downy mildew estimated in the area, M is about the presence of downy mildew spots within the plot under control. It is worth noticing that the GrapeMilDeWS design is cycle-less, which makes the enumeration of possible behaviour patterns straightforward. The system however includes implicit poling loops which are external to the core statechart: i.e. routines which monitor ILM changes and rain forecasts. In stage 0, this affects only the two "wait" states. It should also be noticed that stage 0, which is a super-state can be terminated by an event at upper level that provokes change to next super-state. For example, if downy mildew has been found in the plot (which can only happen if [ILM=RISK] is true once), then the decision system would stand in the wait (rain) state. If the checking of phenology (another implicit monitoring) shows that C1 should be performed because there are more than 5 leaves unfolded, then the wait (rain) would be cancelled as a consequence, and no spraying would be done in stage 0.

GrapeMilDeWS had been modelled with the experts checking the statechart diagrams. The modelled was then verified against the actual past decision behaviour of experts, both qualitatively (Léger et al, 2009) and quantitatively (Léger et al, 2010b; Léger, 2008). Once it was confirmed that the model was the best expression of GrapeMilDeWS that was available, it was deemed the official specification of the system and from then on the norm to refer to when communicating all the decision logics and processes between the expert designers and the number of people who would apply GrapeMilDeWS in different plots of different regions. As explained in (Léger et al, 2010b), the performance and robustness of this decision system needs to be evaluated through field experiments.
Yet, besides the core decision system, there was also need to document the testing protocol: plot choice, sampling methods in the field, data to collect, and protocols to assess quality and efficacy of the protection. It was then decided to derive tree structures of quasi-natural language statements from the Statecharts, one tree for each stage. This transcription could then be included into the testing protocol. The "no-loop" feature and the decomposition in different stages over the season of the core model greatly eased this work. A sample of this transcription process is given in table 1.

Stage 0

Start : bud-break, first leaf unfolded

Quit : 5 leaves unfolded. Goto plot survey C1 provided security delay if there was a spraying

Procedure :

- Watch-out ILM (daily if possible)
- If ILM goes positive (ILM = RISK) and as soon as this happens
  - Proceed to plot survey C0 (downy mildew)
    (Sampling Protocol of plot Survey omitted)
  - If M result of C0 is positive (M = PRESENT)
    - Watch-out weather forecast and wait a rain is forecasted
    - If rain is forecasted
      - Spray against downy mildew before date of this rain

Table 1 Sample from GrapeMilDeWS written protocol as of 2008: stage 0

Note that the "ASAP" (As Soon As Possible) for spraying of Statechart has been replaced by a "before rain" statement which is more functional. Generally speaking, the original reactive system assumption (Léger et al, 2010b) of common Statecharts was replaced by explicit deadlines (rain events, duration related to activity period of products). The modelling of time deadlines, or $[date_{min}\ date_{max}]$ time periods in which an action should be performed according to the protocol, can be achieved by timed flavours of discrete event systems such as timed automata or timed statecharts.

Stakes of GrapeMilDeWS experiment

The network for experimenting GrapeMilDeWS has been continuously growing since 2007. It included more than 20 plots in 2008, and 35 plots in 2009 (Naud et al, 2009). Since 2008, the experiment is based on a protocol document introduced above, and which includes a tree-structured written version of GrapeMilDeWS (denominated hereunder "tree").

Yet, a few cases have shown that stage 0 as of 2007 should be ameliorated, at least for early and quick epidemics of downy mildew (e.g. some cases in 2008 in Bordeaux region). A modification about powdery mildew management was also introduced in stages 0, 1 and 2. As a consequence, because GrapeMilDeWS describes precisely the appropriate timing of actions, the specification and its protocol transcription grew in size. For instance, stage 0 could be translated in 2008 in just 7 lines, (as seen in table 1) and grew to be 28 lines long in 2010. The new details are hereunder exemplified in table 2. The changes in stage 0 did propagate need for change in stage 1, to account for example of the remaining protective activity during stage 1 of an eventual spraying during stage 0. In 2008 the protocol's stage 1 was 28 lines and has grown in 2010 to be 97 lines long. It has become more tedious for people in charge of testing to follow the decision procedure from the protocol. The tree for certain stages like stage 1 contains details that are necessary in order to guaranty the completeness and consistency of system's logics but which may have little use except in special circumstances.

The design and evolution of GrapeMilDeWS, and probably of any Decision Workflows (DeWS) for other pathosystems, raises contradictory concerns. The initial design should have a simple structure in order to be discussed between experts, to trigger innovative idea and foster creative risk management. The reasoning principles and the requirements should be understood by people who test the DeWS. The DeWS should cover situations with intermediate epidemics, as well as cases with low or strong dynamics. Yet, to be operational, it should provide some time flexibility, with well-defined limits. Finally, it should be robust to special circumstances, and be sufficiently adapted to many of the various bioclimatic and terroir conditions found in France.

We believe that the main cases, the most frequent configuration, should be outlined in a natural language protocol document. The complete specification, with the finer details and time constraints, needs to be implemented into a workflow management system. This is reason number 1 for implementing GrapeMilDeWS in the YAWL framework. Besides providing the guidance and calculation to choose the right decision path, the program should allow user to make anticipations, make these persistent in the system as provisional decisions, and revise the latter according to changes in resources availability or weather forecasts. The user should also be able to store comments in the system, for his own use or feedback to DeWS designers.

The second reason for implementing the DeWS with YAWL is the need for collecting data about the tests for later analysis. Without a support program for GrapeMilDeWS, the data have been collected in electronic forms, and aggregated later. This makes work of people testing the DeWS partly redundant, because they have to gather information to build the decision, and then store this same information, with all details such as evolution of rain forecasts, in electronic forms. Systems dedicated to workflow management seamlessly integrate procedures, web services and databases.

Last, collecting numerous test data about GrapeMilDeWS in many places has much interest beyond GrapeMilDeWS itself. Data about non-treated plots, either for epidemic studies or for crop health vigilance, are available in many regions. Yet, there is currently a lack of data about plots with few or very few treatments. Data from GrapeMilDeWS experiment could serve other experts to design their own candidate DeWS. They should contribute to practical epidemiological knowledge, with characteristic facts for each period of the season.

From decision procedure to workflow

Concurrent processes

The table 2 shows the 2010 version of the transcription of stage 0. Most details are related to the case of one important cultivar of southern east of France. More important to our point about concurrent processes: it can be noticed that pre-scheduling of important dates such as one that ends stage 0 (C1) has been mentioned. This fixes a date for which concurrent monitoring processes such as rain forecast and updating ILM can be ended. With the tree structure, concurrency of such processes can only be suggested by appropriate terms. The Statechart (Figure 1) insisted on the preferred sequence: first check ILM, then rain forecasts, after having done a C0 survey on the plot.

Start : bud-break, first leaf unfolded

Quit : 5-7 leaves unfolded. Goto plot survey C1 provided security delay if there was a spraying

Procedure :

- Schedule date of plot survey C1 (date code : D(C1))
- Watch-out ILM (daily if possible)
- Watch-out rain forecasts

- Case of Carignan cultivar if powdery mildew flag-shoots observed year before
    - Estimate possible evolutions of ILM till stage 5 leaves unfolded
    - If ILM should stay NO_RISK till stage 5 leaves unfolded
        - Schedule a treatment against powdery mildew (date code : D(T0)) preferably at stage 3 leaves unfolded and latest stage 5 leaves
        - (Re-)schedule D(C1)) accordingly (delay for going in plot after treatment)
        - Perform the treatment according to schedule (see above)
    - Otherwise: if ILM susceptible to reach RISK (ILM init to NO_RISK on stage0 start)
        - Schedule a treatment against powdery mildew (date code : D(T0)), stage 5 leaves latest
        - (Re-)schedule D(C1)) accordingly (delay for going in plot after treatment)
        - On occurrence of ILM reaching RISK
            - If rain forecasted before D(C1)
                - Perform a mixed (downy/powdery) treatment before this rain
            - Otherwise (no occurrence of rain forecast before D(C1)
                - Perform the treatment powdery against powdery mildew as scheduled
        - If no such occurrence of ILM reaching RISK
            - Perform the treatment powdery against powdery mildew as scheduled

- Other cultivar cases or Carignan without flag-shoots observed year before
    - On occurrence of ILM reaching RISK
        - If rain forecasted before D(C1)
            - Perform a treatment against downy mildew before this rain
        - Otherwise (no occurrence of rain forecast before D(C1)
            - End of stage without spraying
    - If no such occurrence of ILM reaching RISK
        - Keep checking ILM and rain forecasts till D(C1)
        - When D(C1) occurs
            - End of stage without spraying

Table 2 Sample from GrapeMilDeWS written protocol as of 2010: stage 0

This C0 survey is no longer part of the protocol. A few cases in 2008 suggested that this optional survey could be counter-productive. If we ignore the case of Carignan, Stage 0 is meant to handle early downy mildew epidemics without forcing to schedule the C1 survey (survey of both diseases) too early. In stage 1, the result of C1 for M can take 3 values: NIL, LOW, HIGH. Both conditions [ILM=RISK] or [M =LOW] trigger the same pattern in stage 1: protect grapevine of the next rain that could spread contamination. Depending on man-power, the scheduling of plot survey C0 could delay such an appropriate pattern, in case of quick and strong epidemics, so it was discarded and replaced by careful estimation of ILM.

The implementation of a software support for experimenting GrapeMilDeWS requires that the concurrent processes are checked with the user and synchronised appropriately. The YAWL language has synchronisation mechanisms with formal semantics and graphic symbols. The

Event Process Chain (EPC) schematics were designed to specify business process specifications, with semantics that were initially not completely formal (van der Aalst et al, 2010). In order to prepare the modelling in YAWL, we sketched the processes with EPC like schematics, using abbreviated quasi-natural language statements. The result for stage 0 is given in Figure 2.

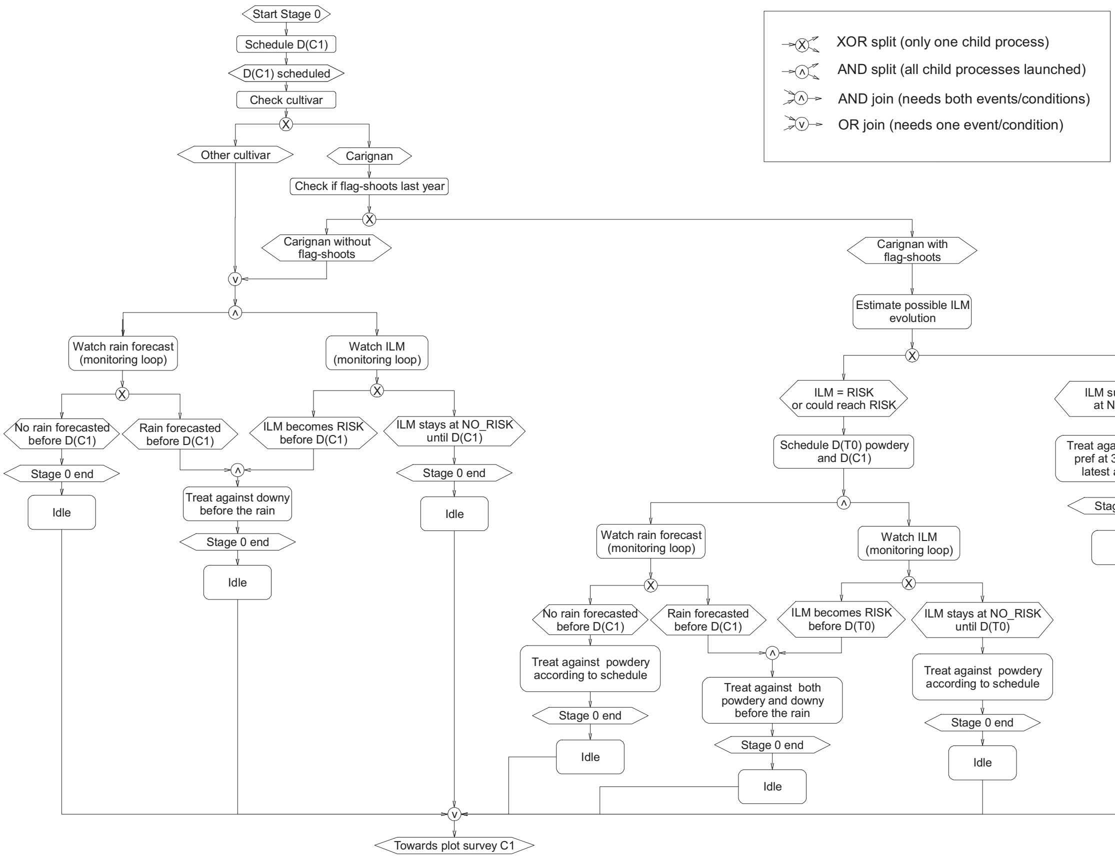

Figure 2 Sample informal EPC of GrapeMilDeWS: stage 0

In Figure 2, it can be seen that monitoring for rain and change in ILM are concurrent processes. One can also check the pattern “schedule a decision deadline (action planned, like survey or spraying, or reference phenological stage)” THEN “monitor till this deadline”. The monitoring will end before this deadline if an event occurs and conditions for action are verified. This pattern can be found in a number of stages of GrapeMilDeWS.

Roles in a multi-actor process

The workflow is a flow of work within an organisation. The decision workflow is related to actual physical and human resources through information need about results of plot surveys, acknowledgment about achievement of scheduled actions like spraying or surveys, updated ILM information with a forecast horizon that also depends on resource management, updated rain forecasts. It does make sense that several persons will receive work items to do and information to report, according to their role in the organisation. A decision workflow is similar to any workflow in that sense. Furthermore, when it comes to experimenting the decision protocol in a network of plots in different regions, some roles concern decision about several plots or giving advice to the whole network. More structure is then needed.

To manage one plot, the following roles have been defined for implementing the worklow: Manager of downy mildew bioclimatic risk (ILM manager); Manager of crop surveys who reports about execution and results of crop surveys; Crop Manager who reports about execution of scheduled spraying; Decision manager who takes and revise decisions.

For managing the whole network, other roles are: Expert adviser about the decision protocol (mainly, its designers or people with extensive experience and critical analysis on the decision protocol); local adviser; data analyst. Besides, the managers of ILM and crop surveys often have several plots under supervision.

Defining precise roles helps to set up easily the workflow for a given organisation, by affecting the roles to the different people participating. Each user of the system has access to a list of work items (one work item is an instantiation of a task of the workflow) that are to be done. YAWL allows defining if a work item should be assigned each time "manually" by an administrator to a given person or automatically proposed to a group. In the latter case, once a user starts one item, it will not appear in the to-do list of the others.

Revision of decisions

Within the management system of the decision workflow, there is no assumption that the decisions of treatment are immediately and automatically applied. This is replaced by the following mechanism to ensure best application of the decision protocol. When the decision manager calculates with the system that a spraying should be performed, the spraying decision receives the status "scheduled". A special process monitors if the scheduled sprayings have been done and gives them this status when it is the case. While the decision is still in the "scheduled" status, it can be aborted or re-scheduled by the decision manager. This can be important for example when adapting to sudden change or precision in weather forecasts, or when a spraying initially targeted to "powdery mildew alone" should be revised to target both mildews. The workflow management system will let the crop manager know if the deadline of a scheduled treatment has passed and no "done" acknowledgement has been given for this treatment.

Weather forecasts and ILM information also can be revised.

Workflow Architecture

The three tier nature of the YAWL system is detailed in (van der Aalst et al, 2010). The core of the system is the YAWL engine. The designer of the workflow will upload the process model onto a server which runs this engine. The specifications are created using a dedicated graphical YAWL editor. As seen above, the workflow includes the "control flow logics" which is created with the graphical editor. The workflow also includes an "organisational model" which will handled by the YAWL engine as well. Finally, the data that support this "control flow", which are inputs, state variables and outputs of the workflow, are specified in an XML schema.

For workflow execution, the YAWL system provides basic user interfaces calculated from the workflow. Although it is not included in the design environment, the architecture makes it possible to use dedicated web-interfaces for the users, which will be developed separately.

Conclusion

The implementation in a workflow management system of the GrapeMilDeWS was an obvious objective so as facilitate extensive testing of this decision protocol, and harvesting the test data in the same time. We have detailed what is at stake. We also have shown the importance of concurrent processes and timing details, which are more easily managed with the assistance of a workflow management system than by reading a potentially complex hierarchy of statements written in a document.

Yet, the workflow for testing the decision protocol is a more detailed model than the model of the abstract decision protocol itself, which we will call “core model”. The pragmatic choice that we made at this stage of this research is to complement the original core model by scheduling and revision processes, which are not formally separated of this core. Yet, we have seen that the core model is susceptible to be enhanced thanks to feedback provided by testing. The testing workflow will have to be maintained accordingly.

Bibliography

Bonicelli B., Naud O., Rousset S., Sinfort C., de Rudnicki V., Lescot J.M., Ruelle B., Scheyer L., Cotteux E. (2010). The challenge for precision spraying. AgEng 2010 – Int. Conf. on Agricultural Engineering, Clermont-Ferrand (France), Sept 6-8, 2010.

Harel D. (1987). Statecharts: A visual formalism for complex systems. *Sci. Comput. Program., 8*(3), 231-274.

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

Léger B., Cartolaro P., Delière L., Delbac L., Clerjeau M., Naud O. (2007) An expert based crop protection decision strategy against grapevine's powdery and downy mildews epidemics: Part 1) Formalization, pp. 145--153, In G. C. Lozzia, A. Lucchi, S. Ragusa Di Chiara, H. Tsolakis, (ed.) Working Group “Integrated Protection in Viticulture”, Proc. of the W.G. meeting in Marsala (Sicily, Italy). IOBC/wprs Bulletin, Vol. 36. IOBC.

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

Léger B., Naud O., Bellon-Maurel V., Clerjeau M., Delière L., Cartolaro P., et al. (2010a). GrapeMilDeWS: a formally designed integrated pest management decision process against grapevine powdery and downy mildews. In J. Papathanasiou (Ed.), Decision Support Systems in Agriculture, Food and the Environment: Trends, Applications and Advances Hershey (PA): IGI Global.

Léger B., Naud O., Delière L., Cartolaro P., Gouache D. (2010b). Contribution of formal modelling and field experiment to leverage knowledge and expertise for designing crop protection guidelines. AgEng 2010 – Int. Conf. on Agricultural Engineering, Clermont-Ferrand (France), Sept 6-8, 2010.

Naud O., Cartolaro P., Delière L., Léger B. (2009). Modelling and experimenting crop protection decision workflows, some lessons from GrapeMilDeWS research. IOBC/WPRS Integrated Protection&Production in viticulture, Staufen im Breisgau (Germany), 1-4 nov 2009. To appear in IOBC/wprs Bulletin.

van der Aalst W., Adams M., ter Hofstede A., Russell N. (2010). Modern Business Process Automation - YAWL and its support environment. Springer. ISBN 978-3-642-03120-5