

A generic framework for simulating agricultural production systems

R. Martin-Clouaire, J.-P. Rellier
INRA, UR875 Biométrie et Intelligence Artificielle
F-31326 Castanet-Tolosan, France
rmc@toulouse.inra.fr, rellier@toulouse.inra.fr

Abstract: The relevance of simulation approaches to the study and design of agricultural production systems is widely claimed. The methodology and computer software appropriate to such a task have however still not reached the state of a mature technology and are mainly developed in research laboratories. Suitable computer models need to represent the structure and dynamics of the underlying biophysical system together with the coordinated human activities involved in the management of the farm production process. Most existing approaches focus primarily on biophysical processes.

This paper outlines the generic framework DIESE especially designed for building and running agricultural production system models. DIESE relies on a rich conceptual basis under the form of an ontology of agricultural production systems. It supports the modelling of the decision process in terms of activities, resources required to realize them, and well-structured constraints bearing on the relevance and feasibility of activities, the interdependencies between them and the restrictions on the uses of resources.

Computationally the ontology comes under the form of a C++ library. In developing a farm production system model, the ontology acts as a metamodel; implementing a model amounts to particularizing the ontology concepts as required by the domain and then instantiating the corresponding classes to capture the specific aspects of the system to be simulated. A discrete event simulation mechanism realizes the step by step interpretation of the strategy and the progressive execution of the decided activities, which in turn alters the biophysical state that otherwise responds to external factors, e.g. weather, influencing biophysical processes.

DIESE is currently used in large modelling projects dealing with various kinds of production such as cash crop, vineyard, pasture-based livestock and pig systems, which attest to the wide scope of applicability of the framework.

Keywords: simulation, ontology, management, plan, activity, resource

1. Introduction

From a system perspective, an agricultural production system involves at least a biophysical system (composed of land, crops, livestock, etc.), a decision system (the farm manager) and an operating system that implements the decisions using various resources (input, labor, machinery, etc.). The relevance of simulation approaches to understand, evaluate and design such systems is widely claimed, although the methodology and computer software currently used to support such investigation have still not reached the state of a mature technology and are mainly confined to research settings. Suitable computer models need to represent the structure and dynamics of the underlying biophysical system together with the coordinated human activities involved in the management of the farm production processes. Production process improvement involves studying timely interactions among biophysical processes and decision making processes at the farm level while most existing approaches tend to address one at the expense of the other, usually focus primarily on biophysical processes.

Mathematical models of farm management are logically attractive but are of limited practical relevance. In particular, those based on static equilibrium conditions can hardly address challenges created by uncertainty and dynamics. Indeed farm systems are complex and dynamic and farming is conducted under conditions of uncertainty. Performance depends very much on how uncertainty is dealt with. Approaches based on averages do not work either. Indeed aggregate differential equations tend to smooth out fluctuations. It is essential that the model be able to reveal how fluctuations might be amplified and how the system may become unstable to large perturbations. Moreover, individual farms are unique and farmers have significantly different practices and preferences that are objects of study in their own right.

Simulation-based approaches that explicitly incorporate production management processes provide a more promising framework thanks to their ability to grasp realistic situations and issues. The management system model should explicitly represent the decision-making process and the implementation of the technical actions resulting from this process. Agricultural production management deals with how farmers combine land, water, domesticated living things, machinery, commercial inputs, labour, and management skills to produce crop and livestock commodities. Farm management (Dillon, 1979) is the process by which resources and situations are manipulated over time by the manager in trying, with less than full information, to achieve his or her goals that might be competing (increase profits, respond to social objectives, or maintain a way of life). The farmer's management behaviour is observable through the choice and timing of activities involved in the various production aspects. This behaviour results from the situation-dependent implementation of his management strategy. Simply stated a management strategy is a kind of flexible plan coming with its context-responsive adaptations and the necessary implementation details to constrain in situ the stepwise determination and execution of the actions.

This paper outlines the generic framework DIESE (Martin-Clouaire and Rellier, 2009) especially designed for building and running agricultural production system models. The capability to represent a farmer's production management behaviour is emphasised. DIESE supports the modelling of the decision process in terms of activities, resources required to realize them, and well-structured constraints bearing on the relevance and feasibility of activities, the interdependencies between them and the restrictions on the uses of resources. Section 2 introduces the ontological basis of the DIESE framework. The main aspects specific to the modelling of management activities are presented in Section 3 and illustrated in Section 4. Dealing with resources in DIESE is addressed in Section 5.

2. Overview of the production system ontology and DIESE

Basically an ontology (Chandrasekaran et al., 1999) is an explicit and declarative description of the domain we are interested in, that is, the concepts in this domain, the properties of these concepts and the constraints on these properties. In addition to providing a shared vocabulary and sense disambiguation, an ontology enables to reuse pre-formalized concepts and templates that can be particularized, instantiated and then mapped into an executable model interpreted with a discrete event simulation engine.

The ontology of agricultural production systems (Martin-Clouaire and Rellier, 2009) serves as a conceptual meta-model supporting the modeling framework. It contains a number of pre-formalized concepts, templates and mechanisms describing the studied system components at a high level of abstraction. The DIESE modeling framework is an object-oriented modeling tool that implements the ontology through a C++ library of pre-programmed classes and services. It comprises a discrete event simulation engine that enables to emulate the continuous and discontinuous features of the simulated system. The modeling enterprise relies on the ontological concepts, templates and mechanisms by particularizing, instantiating and

then mapping these into an executable dynamic model of a specific system. The ontology considerably eases model development and implementation for non-computer-specialists, as the framework helps in clarifying how to organize the expert knowledge about the system of interest in the knowledge base.

The three fundamental concepts of the ontology are: entity, process and event. These represent the structural, functional and dynamic aspects of a system respectively (Rellier, 2005). An entity describes a kind of material or abstract item in the area of interest. The state of a system at a given moment in time is the value of the slots (properties) of the entities it comprises. A process is a specification of part of the behavior of a system, i.e. of the entities composing it. Typically, the process code specifying this behavior includes the use of methods attached to entities affected by the process. A process causes a change in state when a particular event occurs. Thus, events convey the temporality of process triggers. For instance, in a livestock system model, the biophysical part may involve an entity such as *animal-batch* that has slots such as *location* and *composition*. The type of value that the composition slot might take is another entity, *set-of-animal* that has a numerical slot describing its size and slot *representative-animal* describing the typical animal of this set through slots such as *weight*, *intake-capacity* or *intake-amount*. The processes that might affect the state of an individual instantiating *representative-animal* include for instance *growth* and *intake*. An event starting a *growth* process might be created at birth time of an animal.

The DIESE framework includes specific constructs to represent various aspects relevant to the management functions. Fundamental to our conceptual model is the commitment to understand things from a farmer's point of view. To be effective, management behavior must be specified by using constructs and language that are intelligible and conceptually close to those actually used in an agricultural setting. The basic unit of analysis in our approach is work activity, which is a common high level concept in production management. An activity is a purposeful engagement driven by certain needs to achieve a certain purpose. Activities are contextual in the sense that actual circumstances condition their relevance and greatly affect the way the intended objective is achieved. Activities usually involve the use of resources (equipment, labor). Whenever a combination of activities must be undertaken with a view to achieving a pre-conceived result, a plan is needed to express how those composite activities should be coordinated. A work plan is the result of reflection on prior experiences and in anticipation of particular goals and likely occurrences of important events. Because of this, plans are not rigid in the sense of a definite and precise specification of the execution steps. Plans are flexible and adaptable to circumstances. A slightly more formal and encompassing conceptual description is given in the next subsection.

3. Modeling flexible management

In its simplest form, an activity, which we will call a primitive activity, denotes something to be done to a particular biophysical object or location, e.g. a herd batch or building, by an executor, e.g. a worker, a robot or a set of these. Besides these three components, a primitive activity is characterized by local opening and closing conditions, defined by time windows and/or predicates (Boolean functions) referring to the biophysical states or indicators. An indicator is a contextual piece of knowledge or information invoked, assembled, or structured to substantiate a decision-making step, e.g. appraisal of remaining forage amount on a field to decide withdrawal of the herd from it. The opening and closing conditions are used to determine at any time which activities are eligible (according to the manager's intention) for execution; they play a key role in defining the timing flexibility.

The "something-to-be-done" component of a primitive activity is an intentional

transformation called an operation, e.g. the harvest operation. The step-by-step changes to the biophysical system as the operation is carried out constitute a functional attribute of the operation. These changes take place over a period of time by means of a process that increases the degree of achievement at each step of the operation until it is completed. An operation is said to be instantaneous if its degree of achievement goes from 0 to 1 in a single step, otherwise the operation is durative which implies that its execution might be interrupted. An operation may require resources such as a mower in case of cutting. In addition, the execution of an operation is constrained by feasibility conditions that relate to the biophysical state. Objects on which an operation is carried out can be individual objects, e.g. a field or a set of fields, or objects having numerical descriptors, e.g. an area. Speed is defined as a quantity e.g. number of items or area, which can be processed in a unit of time. The duration of the operation is the ratio of the total quantity to the speed. In order to have the effect realized the operation must satisfy certain enabling conditions that refer to the current state of the biophysical system, e.g. the field to be processed should not be too muddy, muddy being an indicator. The ability to reap the benefits of organizational and timing flexibility depends on execution competence determined by the involved resources (both operation resources and the executor). Careful representation of the resources and their availability might therefore be essential to get a proper understanding of the situation under study.

Activities can be further constrained by using programming constructs enabling specification of temporal ordering, iteration, aggregation and optional execution. To this end, we use a set of non-primitive or aggregated activities having evocative names such as *before*, *iterate*, and *optional*. Others are used to specify choice of one activity among several (*or*), grouping of activities in an unordered collection (*and*) and concurrence of some of them (e.g. *co-start*, *equal*, *include*, *overlap*). Formally a non-primitive activity is a particularized activity. As such it might also be given opening and closing conditions as well as other properties such as a delay between two activities involved in a *before* aggregated activity. In particular, it has a relational property that points to the set of the other activities directly involved in it (or constrained by it). In addition, it is equipped with a set of procedural attributes that convey the semantics of the change in status specific to each non-primitive activity. The opening and closing of a non-primitive activity depend on their own local opening and closing conditions (if any) and on those of the underlying activities. All the activities are connected; the only one that does not have a higher level activity is the plan. In addition to the timing flexibility attached to the opening and closing conditions of its activities a plan is made flexible by the use of composed activities that enable optional execution or choice between candidate activities. Whether an optional activity is executed and which alternative activity is chosen are context-dependent decisions.

Notwithstanding the flexibility of activities it may be necessary to adapt the plan when particular circumstances occur. Indeed a nominal plan conveys the rough course of intended steps to go through under normal circumstances. The specification of when and what changes should be made to a nominal plan is called a conditional adjustment. The trigger for a conditional adjustment is either a calendar condition that becomes true when a specific date is reached, or a state-related condition that becomes true when the current circumstances match this condition. The adjustment can be any change to the nominal plan such as the deletion or insertion of activities. It can also affect the resources used in some activities. Actually a conditional adjustment can also specify a change to be made to conditional adjustments themselves. By this means, the management can be reactive and thus cope with unexpected (though still feasible) fluctuations of the external environment (e.g. drought) and various contingencies.

4. Examples of activities in grassland-based livestock systems

The example (Martin et al., 2009b) considered here concerns the hay-making activity on a farm: cutting the herbage of a grassland plot and, once it is dry enough, storing this new-mown hay. Two primitive activities are involved: cutting and storing. For the cutting activity, the object operated by the cut is a plot, in particular the component herbage, and the executor is the farmer equipped with his tractor and mower. The speed of the cut (the *something-to-be-done* component) is a harvestable area per unit of time. Its effect is the creation of a harvested herbage, the initialization of a drying process on this harvested herbage, and the re-initialization of the herbage component of the plot with its descriptors updated (leaf area index, dry matter, growth cycle age, digestibility, etc.). For the storing activity, the object operated by the storage is the harvested herbage, and the executor is the farmer equipped with his tractor, round-baler and trailer. The speed of the storage is a storable quantity of hay per unit time. Its effect is the crediting of the amount of hay stored in the barn by the harvested quantity minus some losses to the yield associated with the whole hay-making process. Storing of harvested herbage can occur only once cutting is complete. Thus hay-making is a sequence of two primitive activities which can be written:

hay-making = *before* (cutting:

operation: cut with mover

operated object: plot

performer: farmer

storing:

operation: store with tractor, round-baler and trailer

operated object: harvested herbage created in cut

performer: farmer)

The opening of any hay-making activity, and consequently of the cutting activity, has to occur within a particular time range delimited by a minimum and a maximum beginning date. In addition, the opening predicate refers to a threshold on harvestable yield and a given phenological stage for the corresponding herbage, i.e. between stem elongation and flowering, to ensure a compromise between harvested quantity and quality. Once the opening predicate of the hay-making activity has been verified the feasibility conditions attached to the cut operation are examined. These feasibility conditions concern the bearing capacity of the grassland plot, sufficient free space in the barn to store additional hay, and a satisfactory expected air saturation deficit and rainfall in the coming days to ensure proper drying conditions in the field. No closing conditions are specified in this case to ensure completion of the hay-making activity. To summarize the hay-making aggregated activity is represented as follows:

hay-making = *before* (cutting, storing)

earliest beginning date

latest beginning date

opening predicates concerning :

- minimum yield to harvest (an amount expressed in kg / ha)

- earliest phenological stage (stem elongation expressed in degree days)

- latest phenological stage (flowering expressed in degree days)

Farmers seldom make hay on a single field at a time. Typically, they do it on a set of fields that are close together, i.e. that belong to the same islet. This practice may be risky if too many fields are cut and long period rainy weather occurs. A typical risk-limiting attitude is to make small groups of plots and harvest these groups in sequence. Bad weather during drying then harms only the plots in the last group treated. The example of practice considered in this paper, hay-making on the plots of a group can only start if the last hay-making activity

executed in the previous group is complete. Moreover a delay is imposed between the hay-making on the different groups of plot (for instance to keep time for daily routine work that cannot be done on the busy days of hay-making). The grouping of activities enables management constraints to be attached to this set, such as the delay between the processing of the groups {Field1, Field2, Field3} and {Field4, Field5}. Using an *and* to make the grouping gives flexibility in the order of execution of the concerned activities, using for instance yield-based preferences. The sequence of hay-making on the two groups of fields can then be written:

before (*and* (hay-making Field 1, hay-making Field 2, hay-making Field 3),
 and (hay-making Field 4, hay-making Field 5)
 in-between delay = 4 days)

Due, for example, to particular weather conditions in a given year, such a plan might be unachievable. Conditional adjustments of the plan are then necessary to recover a consistent management situation. For instance, in a showery weather period, the farmer might decide to reverse the order of the groups of hay-making activities in the sequence (*before*) to take advantage of the lower drying requirements of herbage on fields 4 and 5. Another adjustment could be the changing of the delay between the processing of the two groups. The above composed activity would then be changed into:

before (*and* (hay-making Field 4, hay-making Field 5),
 and (hay-making Field 1, hay-making Field 2, hay-making Field 3)
 in-between delay = 3 days)

5. Resources

As pointed out by Dillon (1979), resource management is the essence of farm management. Basically, in the DIESE framework, a resource is an entity that supports or enables the execution of activities. Typically, the activity executors, the machinery involved and the various inputs (seeds, fertilizer, water, fuel) are resources. Resources are generally in finite supply and have significant influence on when and how activities may be executed. The availability of a resource is restricted by availability constraints that specify the conditions allowing their use or consumption. The constraints are temporal constraints (time windows of availability), capacity-related constraints (the amount available) or state-related constraints. Any resource is possibly constrained with respect to the maximum number of operations supported simultaneously and the maximum number of resources of other types that can be used simultaneously.

There are many types of resources that must be dealt with (Smith and Becker, 1997). A resource can be either consumable (usable only once) or reusable after it has been released. It can be a discrete-state resource (whose availability is expressed by a qualitative state such as *ready* or *not ready*) or a capacity resource (whose availability is characterized by a vector of numerical values expressing a multi-dimensional capacity). We distinguish between single resources and aggregate resources, which are collections of resources.

In a primitive activity, the role of resource is played by the operated object, the operation resources and the executor as well. An operated object is a discrete-state resource that is a part of the biophysical system (an entity or a set of entities of the biophysical system). It is characterized by its ability to be transformed by several operations simultaneously. It may allow several resources to be simultaneously involved in transformations, and several executors to carry out certain transformations simultaneously.

An operation resource is either a discrete state resource (e.g. tools) or a capacity resource (e.g. diesel fuel). It is characterized by its ability to be used simultaneously for several objects

acted upon in the biophysical system, to be involved simultaneously in several operations, and to be used simultaneously by several executors.

An executor is a discrete-state resource characterized by its ability to work simultaneously on several objects in the biophysical system, to be involved simultaneously in several operations, to cope with several operation resources used simultaneously in the operations it is engaged in. Another feature of an executor is its work power that has an effect on the speed of the operation and on the requirement of operation resources if the latter are declared proportional to power. An executor is either an individual resource (e.g. a worker) or a labour team (a set of individual workers whose work power is by default the sum of the powers of the individual workers it comprises).

As an illustration, consider a cutting activity having the resource specifications shown in Figure 1. The operated object specification refers to a set of spatial entities that are dynamically generated by expanding the entity set specification defining this set. Considering it as a resource is useful in case it is decided to disallow two simultaneous operations on any of these entities. The specification of resources coming with the operation component states that two machines are required: a mower and a tractor. The executor is a person to be selected either from the farmer's sons or his employees.

<i>What is specified:</i>	<i>Specification:</i>	<i>Instances of entities or resources (*):</i>
Operated objects	"non-grazing fields greater than 0.5ha"	FIELD: {f1, f2, f3, ...}
Operation resources	"one mower and one tractor"	MOWER: {m1, m2} TRACTOR: {t2}
Executors	"one person from farmer's sons or his employees"	SON: {s1, s2, s3} EMPLOYEE: {e}

(*): small capitals refer to classes, normal characters refer to existing instances of the class.

Figure 1. Resource requirements in a cutting activity

If we have instances available in each of these classes, we have to consider two alternative allocations. In this example, at the time of allocation, the allocation procedure would return two alternatives {(f1, m1, t2, s2), (f1, m1, t2, e)} if f1 is the only field satisfying the request, m1 and t2 are the mower and tractor that are available, and s2 and e are respectively the second son and the employee that have no duty at that time. It might return a set of only one collection of assignments if either no son or no employee is available. It might of course return no solution at all, meaning that it is impossible to execute the activity immediately.

The use of resources is restricted by various constraints that make resource allocation a tricky combinatorial task. In addition to availability constraints, the ontology makes it possible to specify co-usage restrictions that concern the simultaneous use of a resource in different operations and combined with other resources. These co-usage restrictions are defined as specific entities having a slot whose value is a set (conjunction) of inconsistency conditions defined as cardinality limitations. The restriction called *activity-inconsistency-conditions* stipulates the limitations on the use of some resources for any primitive activity of a given type whereas the one called *resource-sharing-violation-conditions* stipulates for a resource of a given type the restriction on its simultaneous uses with other resources involved in set of primitive activities executed concurrently. Finally, a third type of usage restriction called *activities-resources-inconsistent-commitments* is available in the ontology. It has two slots whose values are a set of *activity-inconsistency-conditions* and a set of *resource-sharing-violation-conditions*; these two conditions must be satisfied in order to meet the requirement expressed by the *activities-resources-inconsistent-commitments*.

Investigating resource allocation means looking at resource bottlenecks and inefficient utilization that have a significant impact on the system performance. Simulation must include the process of dynamic allocation that takes place repetitively as part of the action-oriented decision making. This process implements a dedicated constraint satisfaction solver. At any time, the management strategy can tell what activities are deemed appropriate and the resource allocation must determine among them the combinations that are feasible with respect to the availability and co-usage constraints. As any constraint satisfaction problem, resource allocation may become a highly combinatorial problem depending on the richness of the pool of resources and the flexibility in their usage. When several allocations of resources are possible and, more generally, when several sets of allocated activities are eligible some preferences are used to ultimately select the best one among them and engage execution.

6. Conclusion

Keating and McCown (2001) already suggested that challenges for farming system modelers are “not to build more accurate or more comprehensive models, but to discover new ways of achieving relevance to real world decision making and management practice.” In this sense, the DIESE project is the result of consistent efforts to improve the representation of farm management strategies and get closer to the questions raised in practice. Using the integrative conceptual framework that we have described one can develop elaborate simulation models of agricultural production systems. It provides a common structure to help organize and frame monitoring and management activities that can be applied effectively and consistently across the production system. Running such a simulation model under various scenarios of external conditions (weather in particular) helps to give a realistic view of the system’s behavior and performance, its sensitivity to external factors and the quality of the tested management strategy as regards robustness and flexibility. We can use this approach to give a clearer meaning to the selection and prioritization of management activities by placing the management process in context.

At this stage, the decision making behavior coming with the DIESE framework lacks an explicit representation of goals. This would become necessary to take into account anticipatory decision making capabilities. Incorporating such capabilities could be done by implementing a Belief-Desire-Intention (BDI) type of decision making architecture (Wooldridge, 2002) in which beliefs express the decision maker current state of knowledge about the production system, intentions are the activities structured in a plan and desires are specifications about target states of the production system.

The complexity of the farmer’s management task is not due to the number of components or possible states of the system but rather to the dynamic behavior of the different components which arise from their interactions over time and their dependence on uncontrollable driving factors such as weather. The dynamic complexity relates to human difficulty in dealing consistently with feedback effects, and multiple and delayed consequences of interventions. Much of the information about biophysical system functioning and the cognitive process involved in production management resides in the mental models of farmers where it remains tacit. By using the DIESE framework, one can expect to capture part of this subjective and context-specific knowledge and, in this way, make it an object of scientific investigation. Improving our ability to make this knowledge explicit and usable for formal modeling and learning can have important effects on both research and practice. Researchers are in a better position to build more complete, accurate and insightful models and practitioners can increase their awareness and mastery of organizational and management issues.

DIESE is currently used in large modelling projects dealing with various kinds of production such as cash crop (Cialdella *et al.*, 2009), vineyard (Ripoche *et al.*, 2009), pasture-based

livestock (Martin *et al.*, 2009a; Chardon *et al.*, 2007) and pig systems (Rigolot *et al.*, 2009), which attest to the wide scope of applicability of the framework.

References

- Chandrasekaran B., Josephson J. and Benjamins V. (1999) What are ontologies and why do we need them. *IEEE Intelligent Systems*, 14 (1): 20-26
- Chardon X., Rigolot C., Baratte C., Le Gall A., Espagnol S., Martin-Clouaire R., Rellier J.-P., Raison C., Poupa J.-C. and Faverdin P.. (2007) MELODIE: a whole-farm model to study the dynamics of nutrients in integrated dairy and pig farms, in: L. Oxley and D. Kularsiri (Eds.), MODSIM 2007 International Congress on Modelling and Simulation, Modelling and Simulation Society of Australia and New-Zealand. pp. 1638-1645.
- Cialdella N., Rellier J.-P., Martin-Clouaire R., Jeuffroy M.-H. and Meynard J.-M. (2009) SILASOL: A model-based assessment of pea (*pisum sativum* L.) cultivars accounting for crop management practices and farmers' resources. In Proceedings of Farming Systems Design 2009, Monterey, CA.
- Dillon, J.L. (1979). An evaluation of the state of affairs in Farm Management. *South African Journal of Agricultural Economics*, 1, 7-13.
- Keating, B.A. and McCown, R.L. (2001) Advances in farming systems analysis and intervention. *Agricultural Systems*, 70, 555-579.
- Martin G., Duru M., Martin-Clouaire R., Rellier J.-P. and Theau J.-P. (2009a) Taking advantage of grassland and animal diversity in managing livestock systems : a simulation study. In Proceedings of Farming Systems Design 2009, Monterey, CA.
- Martin G. (2009b) Analyse et conception de systèmes fourragers flexibles par modélisation systémique et simulation dynamique. Ph. D. thesis, Université de Toulouse.
- Martin-Clouaire R. and Rellier J.-P. (2009) Modelling and simulating work practices in agriculture. *Int. J. of Metadata, Semantics and Ontologies*, 4(1-2), 42-53.
- Rellier, J.-P. (2005) DIESE : un outil de modélisation et de simulation de systèmes d'intérêt agronomique. Internal report UBIA-INRA, Toulouse-Auzeville.
http://carlit.toulouse.inra.fr/diese/docs/ri_diese.pdf
- Rigolot C., Chardon X., Rellier J.-P., Martin-Clouaire R., Dourmad J.-Y., Le Gall A., Espagnol S., Baratte C. and Faverdin P. (2009) A generic framework for the modelling of livestock production systems : MELODIE. In Proceedings of Farming Systems Design 2009, Monterey, CA.
- Ripoche A., Rellier J.-P., R. Martin-Clouaire, Biarnès A., Paré N. and Gary C. (2009) Modeling dynamically the management of intercropped vineyards to control the grapevine water status. In Proceedings of Farming Systems Design 2009, Monterey, CA.
- Smith S.F. and Becker M.A. (1997) An ontology for constructing scheduling systems, In Proceedings of the AAAI Spring Symposium on Ontological Engineering, pp 120-129, Palo Alto, CA.
- Wooldridge M. (2002) *Introduction to multiagent systems*. John Wiley & Sons.