Design of a multimodel of a dairy cows herd attacked by mastitis

C. Force a,*, L. Péronchon b, D.R.C. Hill a

a LIMOS UMR CNRS 2239/ISIMA, Université Blaise Pascal, Campus des Cézeaux, 63173 Aubière Cedex, France
b INRA, Unité d’Épidémiologie Animale, Theix, 63122 Saint-Genès-Champanelle, France

Received 21 May 2002; received in revised form 21 October 2002

Abstract

We present the building of a discrete event multimodel simulation tool of livestock farming systems, regrouping several mathematical models, and reproducing cattle herd performances through an individual based model. The main goal of this simulator is to study and predict consequences of mastitis occurrences on dairy herds taking into account individual characteristics of animals and seasonal and livestock parameters. The model presented is the result of a close collaboration between epidemiologists and simulationists and we were able to use real field data from animal experiments. The project includes a domain analysis of dairy herd systems in France, using an object oriented approach and Unified Modelling Language tools. The software, written in Java, is to become a decision-aid tool for stock breeder advisors.

Keywords: Multimodel; Individual based model; Discrete event simulation; Dairy herd; Cows; Mastitis

1. Introduction

Mastitis is the most frequently occurring disease of dairy cows. Its effects on animal performances have great economic consequences and over the last fifty years studies have demonstrated the epidemiological complexity of such a disease [3]. The risk factors involved are multiple (animal characteristics, management practices, environment), interrelated (breed and milk yield, for instance) and located at different abstraction levels (cells, animal, herd) [22,26]. Consequences of mastitis
are manifold too: physiological (milk production and quality) and financial (vet care, drying off, culling, milk payment) [8,19,25]. Consequently to this complexity, the stock-breeder must make decisions according to his milk yield and quality objectives whilst also considering the constraints of his farm (milk quota in Europe) [18]. Such decisions have economical and sanitary consequences in the short or longer term.

A lot of papers propound various analyses of such systems. They offer a large diversity of data, but each of them focuses on only one given level of perception. These studies provide mathematical models related to particular aspects of these systems, mainly: milk production, mastitis occurrences, impact of mastitis and reproduction. [15,18]. They were used in order to design software applications (optimisation models or simulators), aiming at prediction of biological variables, and even to contribute to optimisation of a particular variable: farming profit maximisation, herd cellular numeration minimisation [1,13,14,18,27]. The study discussed in this paper deals with the two main performances of a dairy herd: mastitis impact and frequency on the one hand, and milk yield on the other hand, with two levels of perception: animal and herd. It presents a domain analysis, based on an object oriented approach, and the design of an individual based simulator. The simulator is implemented in Java. This work has a dual originality: first, the conceptual modelling respects biological entities facilitating future extensions of the simulator, and then, mathematical models previously developed are integrated [19–21,23]. The aggregation of different formalisms in the same model implied the use of multimodelling techniques [10]. The fundamental objective of this simulator is to allow comparison of zootechnical and sanitary performances of various breeding practices. Taking time into account is fundamental because stock-breeder decisions may occur at any time, reacting to events (mastitis for instance) and can depend on previous events (in culling process, particularly). Variables and results involved are located at animal or herd level, in regard to pertinent perception levels for mastitis risk factors as well as for stock-breeder actions.

In the first part of this paper we expose the design approach used to develop the discrete event and individual based simulation model. Then, we present the functionality of the simulator and examples of parameterisation and results. Finally, we discuss further extensions.

2. Data and methods

2.1. Data

Data used come principally from LASCAR database [5], which regroups zootechnical and sanitary information from three different experimental domains of National Institute of Agronomic Research (INRA) during the last 10–20 years. This database contains weekly dairy yields (average daily yields measured at milking), and information on diseases, individual cow characteristics (breed, parity) and management practices for 3851 lactations completed by 1179 cows. This large amount of
data made possible many statistical studies and the elaboration of models concerning milk production (lactation curves) as well as mastitis or lameness occurrences and consequences. In particular, mathematical models included in the simulator have been built thanks to the LASCAR database.

2.2. Methodological approach

Object oriented analysis offers a conceptual framework well suited to biological system modelling [4]. It proposes to the various domain experts a unified view of a system and helps solving communication problems. Unified Modelling Language (UML) [2] and the Unified Software Development Process [17] were chosen because of their universality.

The simulation model relies on an individual based model [7,16]. The characteristics of each animal are preponderant in mathematical models and in stock-breeder decision making as well. This approach allows us to include in the model a large individual variability on breed, lactation number and production potential, to make environmental conditions evolve (seasons) and to achieve results at different levels.

3. Conceptual modelling

3.1. Multimodelling

A specific model usually simulates a special aspect of a complex system with a particular modelling and simulation technique (Partial Differential Equations, Cellular Automata, Neural Networks, Individual Based Model with or without Genetic Algorithms, etc.). A multimodel of a complex or huge system can be achieved with a composition of different specialised models of arbitrary kinds and with several abstraction levels (Fig. 1).

In addition, since model components can be simulated separately, the principle of multimodelling is to support the hierarchical refinement of heterogeneous models through functional coupling. The computational refinement or derivability of a

![Fig. 1. UML class diagram presenting a metamodel of the multimodelling principle.](image-url)
model is a tough problem that should not be hushed up. The analogy between multimodeling and object oriented programming has been demonstrated by [6]. Furthermore, multimodels can be specified by Zeigler’s DEVS formalism (Discrete Event system Specifications), which presents a mathematical ground helping to handle the well known aggregation problem [28] encountered by biologists and simulationists. DEVS has been successfully applied to various ecological problems with multiple aggregation levels over the last decade. There is a close link between multi-formalism and multimodelling: they are both founded on theoretical system concepts but they can be distinguished since they rely on different abstraction levels. The history of object oriented analysis and design methods has led to UML in 1997. Now recognised as a standard in the scope of the Object Management Group (OMG), although, controversies and alternatives (OPEN Modelling Language: OML) exist, we retained UML for our multimodel designs since the resulting diagrams will be more widely read using this graphical notation.

3.2. Domain model

In object oriented approaches one of the first important steps is the building of a Domain Class Model representing the expert view of the studied system, this also applies to [12]. In order to facilitate the understanding of the numerous associations between classes on the diagram, we decided to direct the main focus of our design to an individual based model which has been broken down into three packages (Fig. 2):

1. The Breeding package describes the herd objects under normal breeding conditions: the stock-breeder manages his herd acting directly on animals or globally on the herd. Animals are fed and produce milk, stored in a tank.
2. The Renewing package contains entities involved in input and output of animals: a cow can leave the herd by natural death or on a stock-breeder decision (culling); the stock-breeder manages the calf flow coming from his own cows and can also purchase heifers outside.

Fig. 2. UML package diagram of domain breakdown.
3. The Mastitis package showing interactions between the cattle herd system and mastitis pathology is discussed below.

In addition to these packages, the global multimodel is also composed of three additional mathematical models, aiming to handle events occurring during the individual animal life. The first model was designed to model the lactation curve, the second to predict the occurrence of mastitis and the third to deal with the impact of mastitis on milk production. All three models were previously developed by our research team at INRA and will be discussed in the next section. Fig. 3 presents a simple UML metamodel of our multimodelling design.

Since the link between the metamodel and the regular model design is not obvious, we propose more insight by showing in Fig. 4 the UML model of Mastitis package. Pathogen germs, situated in the environment (cattle shed, grazing) among a given distribution, enter the udder. Clinical mastitis are detected by the presence of clots in the milk or by inflammation of the udder. Subclinical mastitis are manifested by an abnormal increase of leukocytes in the milk (cellular numeration). Mastitis can act on quality and quantity of milk yield. The stock-breeder may decide to treat the infected cow by antibiotics, or not. Milk is not kept during the treatment period. Individual characteristics of each animal influence the occurrence risk and consequences of mastitis.

The central classes “cow”, “stock-breeder” and “herd” appear in the three package diagrams, each of them focusing on specific associations. The whole domain class model was elaborated in collaboration with a group of biologists and was completed by dynamic models as state/transition diagrams and sequence diagrams.

### 3.3. Mathematical models

In order to generate and treat events occurring along individual animal life, we have previously said that three mathematical models have been developed:
Model of milk yield evolution during a lactation period under standard INRA management conditions [23]. This model computes the lactation curves of each animal according to its breed, stage of lactation (week) and yield potential, taking into account an effect of season and an effect of pregnancy. This model was retained because of the quality of the adjustment. It performs the prediction of the theoretical (i.e. without pathology) yield of milk of each animal. This computation is performed at calving, for each week of the forecast lactation period. It serves as a basis for computing the actual yield milk which depends on events occurring along the lactation period.

Predictive model of mastitis occurrence [20]: this model is used first to assign a number of cases of mastitis (up to three) for each individual during its lactation period, taking into account variables related to the cow (such as breed and parity) and to management practices (like housing hygiene, milking practices and drying off procedures) through the herd risk level. Moreover, equations give the interval from calving to first case, and the intervals between subsequent cases. This model is used in order to generate dates of insertion of mastitis occurrence events in the scheduler.

Model of impact of mastitis on milk production [19]: this model is used during simulation in order to provide mastitis consequences and their duration according to the week of mastitis onset and the production level at this date. Four response patterns are possible: (a) no marked modification of the lactation curve, (b) decreased production for less than five weeks, (c) decreased production for more than five weeks and (d) drying off or culling. In case of decreased production, the model computes the milk loss along the time, and we obtain the effective milk yield of the cow.
In case of drying of or culling, the corresponding events are inserted in the scheduler, modifying the forecast lifecycle of the animal.

Complementary models were developed in order to forecast weeks of fecundating insemination, calving, insertion of a new cow and culling under normal conditions. These models have been built thanks to the LASCAR database [5] and [24].

4. Design of the simulator

4.1. Functionality

The simulator design is mainly focused on interaction between clinical mastitis and milk yield. Objectives, in result terms concern frequency and distribution of this pathology along time, its impacts on milk yield, and expected milk yield (without pathology). All results need to be available at herd and individual levels. A simulation run covers several consecutive lactations. The simulation step is one week. Culling of cows and input of heifers are performed.

Stock-breeder is the decisional entity of the system, he controls his cattle herd applying management rules: control of minimum quantity of milk yield before drying off, culling rule, duration of antibiotic treatment, dates of place change (grazing, shed).

Several mathematical models being not deterministic, the number of replications requested by the user is automatically performed as well as the result file management into folders.

Two kinds of results are generated: graphical results can be displayed in order to follow the simulation runs, text file results are produced with a view being statistically exploited later. File results are configurable: a user can choose the observation variables he is interested in, split them into several files and control the file structures.

4.2. Software architecture

As shown at Fig. 5, the simulator is structured into five realisation packages:

- **BioIo package** regroups classes coming from the domain analysis and includes the Event package which contains event classes generated during simulation:
  - Calving event,
  - Culling event,
  - New_Cow event (introduction in the herd of a heifer, ready to calve),
  - Drying_off event,
  - Mastitis event,
  - End_of_Treatment event, etc.

- **Utils package** contains mathematical models and functions. This package is designed according to the Strategy Design Pattern [11]. This choice was made because of two constraints: first, the need to take into account improved mathematical models in the future, and compare them with older ones. Second,
in order to validate interactions between models, we need to replace one or several mathematical models by real observations.

- **Interface package** supplies a graphical user interface for entering system and simulation parameters: number, breed, stage and week of lactation of cows, simulation starting date and duration, number of replications. Characteristics of cows can be given individually or globally by means of percentages.
- **Techno package** includes technical classes of the simulation kernel and the scheduler.
- **Output package** is in charge of formatting outputs. It contains observer classes which can be parameterised in order to collect results requested by user.

### 5. Parameterization of simulation

Initialization parameters are of five categories:

- General parameters: working directory, number of replications, beginning date and number of simulation weeks.
- Stock-breeder management rules: insemination, drying off, culling.
- Cattle herd characteristics: mainly occurrence risk of mastitis.
- Individual cow characteristics: breed, stage and week of lactation, initial milk yield.
- Graphical results choice and content of output files.

The interface presents a screen by parameter category and creates an input file for each of them. These files are kept on the working directory and can be reused with or without modification. A user entering new parameters can use a pseudo random generation instead of giving a value. As shown at Fig. 6, fields possess a symbol open (手持) or closed (走). A click on the “random” button fills the fields whose padlock is open. The user may choose a distribution law. Default values are available. Characteristics of cows can be entered individually for every animal or by means of distribution law for the different parameters. A statistic graphical window is displayed for verification.
6. Results and verification

During execution, histograms give the number of mastitis occurrences by week of simulation. We can also follow during the execution the calving date repartition, or the evolution of the herd composition (lactation number of cows). Such curves and histograms aim only at progress monitoring. Graphical display is not available when performing replications. Main results of simulation, at individual and herd level, are recorded in output files.

Result files are produced under a portable format in order to be processed by statistics and data analysis software products. The user can request the number of replications he wants. For each replication, all files specified by the user are stored in a numbered folder under the working directory. Such an organisation of files allows and facilitates confidence interval computing, which is essential in particular on global and economical results.

File results can be obtained at herd level: total milk yield and milk loss, for instance, and at individual level: individual milk production and track of events related to each animal (insemination, mastitis, drying off, calving or culling). Fig. 7 shows, as an example of individual results, two consecutive lactation curves of a specific cow: the first lactation contains two mastitis occurrences, and the second lactation none. The predicted curve at calving does not take into account the gestation and mastitis impact, while the simulated one respects all events occurring during the lactation. The first lactation shows a milk loss due to mastitis, while in the second one, the little difference is explained by the gestation.

Individual results, and, in particular, lactation curves of cows, are conform with results of precedent studies and have been compared with models of lactation curves published in [23] and [19]. The model of mastitis occurrences and complementary models (fecundating insemination, calving, insertion of new cow, culling) have also been verified. The process of verification was facilitated by the development
of a software architecture giving the possibility of replacing a stochastic model by data issued from ground studies. Such a Strategy Design Pattern application permitted specific simulation runs aiming to verify each category of model independently.

Validation can be driven thanks to the large amount of data on livestock farming recorded by INRA in LASCAR particularly. Even though the entire process of validation is not complete, results on a stock-farming evolution can now be obtained and statistically compared with real data.

7. Conclusion

We have presented the design and implementation of a multimodel simulator of dairy herd. The design is based on a domain analysis, using an object oriented approach. The development relies on an individual based model of the stock-breeder and animals, and on mathematical models for supplying the fine working of the system. A discrete event model manages the system dynamics and reproduces the behaviour of a cattle herd in over a period of time. The combination of the previous models uses different formalisms at different abstraction levels. The multimodel is described by an UML metamodel.

The corresponding software is fully operational with satisfactory execution times. The software architecture is built in order to be portable and extensible, in addition the user interface authorises the specification of the output variables and also enables the management of replication experiments in various directories. The technical packages are reusable in other domains of breeding. As input data sets, the software can use real observations, expert assumptions or a mix of both kind of data. The results obtained have been validated by epidemiologists and physiologists. The simulation produces individual results and aggregated results that were considered realistic by experts. Individual results will be used to understand peculiar behaviour.
observed at the herd level. Other developments may concern both simulator enrichment and domain deepening, including:

- Integration of models concerning quality and cellular numeration of milk.
- Computing risk of mastitis occurrence in terms of stock-farming characteristics.
- Adjunction of other pathologies.
- Stock-breeder representation as a thinking and acting agent [9], taking into account environmental constraints.

Such a simulator, after being supplemented, will become a decision-aid tool, allowing testing and recommending sanitary and management practices to stock-breeders, according to their objectives, and respecting livestock farming constraints.

Acknowledgements

The authors thank F. Cunha, V. Fauconnot, E. Fudral, F. Garcia, M. Garcia, E. Teyssier, for their work in the model implementation.

References