O.D.D.: a Promising but Incomplete Formalism For Individual-Based Model Specification

Edouard Amouroux[1,2,3], Benoit Gaudou[1,2,3], Stéphanie Desvaux[4], Alexis Drogoul[1,2,3]

Abstract—**In this paper we introduce the ODD protocol, proposed in ecology for easing the communication and replication of Individual Based Models (IBM). First, we use it to describe a realistic model in epidemiology, which deals with the local determinants of the propagation of the H5N1 virus. From this description we first point out the advantages of ODD. Then we focus on its weaknesses, the major ones being its ambiguity and its inability to attain replication. Finally, we propose several improvements for this protocol in terms of re-organization, better specifications (adapted from existing software engineering methodologies) and the addition of a section dedicated to experimentations.**

Index Terms— **Agent-Based Model, Epidemiological Model, Model Specification, Multi-Agent System.**

I. Introduction

Individual based modelling (IBM) and simulation are becoming a popular approach in sociology and ecology [1][2][3]. Nevertheless, a basic issue still subsists: how to communicate clearly a model and ease its replication? This issue is even more crucial for modern modelling approaches such as *structural realism* [4]. These guidelines refuse *a priori* simplification of the system to be represented, avoiding many of the shortcuts commonly used in more classical models.

ODD (Overview, Design concepts, Details) is a domain independent protocol for communication of IBM [5] that is gaining momentum nowadays, in particular in ecology. It is intended to allow a complete re-implementation of a model and also to permit the replication of its results. This is of interest similarly to experimental science when one wants to replicate a scientific experiment.

Even though ODD has already been largely used, it is important to note that it has only been evaluated by modellers in ecology and sociology (see **[5]** annex, [3]). In addition, most of these papers rely on the description of theoretical or toy models. Therefore, if ODD is to become the mainstream protocol in IBM, there is the need to scrutinize it from outside these communities and use, if possible, more realistic models. Moreover, since modelling is now becoming pervasive in all the software-related disciplines [6], it could be interesting to see how it stands in front of the more classical *software engineering* approaches and what a formalism originated in a very different domain could bring to them.

The aim of this article is then to evaluate ODD from a computer science point of view, using a complex and structurally realistic model: GAMAVI. This model intends to allow thematician (domain specialist) to explore the influence of the environment on the spread and persistence of the avian influenza virus among poultry flocks in North Vietnam **[**7**]**.

This model is presented in section III using the O.D.D protocol. From this formal description an implementation has been done using the GAMA simulation platform*. GAMA is a domain-independent agent-based simulation platform [8] which focus on ease of access (thanks to an XML based modelling language) and geographical data access. This paper reports on our successes and failures in the former activity and is organized as follows: in section 2, we provide a general introduction to ODD. In section 3, we show how we used it to fully specify the GAMAVI model. Section 4 presents the different weaknesses of the current version of ODD, with respect to model specification and replication. Finally, Section 5 summarizes the paper and provides some perspectives on our future work.

II. Description of ODD

The main contribution of ODD is a *textual template* to describe a (individual-based) model in 3 sections: Overview, Design Concepts and Details.

Overview consists of three subsections: 1) *Purpose; 2) Entities, state variables and scales;* and 3) *Process overview and scheduling*. It is a detailed introduction to the model, which should give the reader enough information on the "how?" and "why?" of the model. A program skeleton with the main entities of the model should be derivable from this.

Design concepts consists of seven subsections: *Emergence*, *Adaptation*, *Objectives*, *Learning*, *Prediction*, *Sensing*, *Interaction*, *Stochasticity*, *Collectives* and *Observation*. This section allows the modeler "to clearly communicate important design aspects of the models" [2].

Details presents the model in depth in three subsections: *Initialization*, *Input*, *Sub-models*. This section may be extensive so it can be included in an online appendix [9] but has to be thoroughly completed, especially in the case one wants to allow for replication. *Initialization* presents the initial state of the system at the beginning of a simulation while *input* is about the data used. The *sub-models* section makes the protocol recursive as each sub-model can be considered as an independent O.D.D description of another self-contained model.

[1] IRD, UMI 209, UMMISCO, IRD France Nord, Bondy, F-93143, France
[2] UPMC Univ Paris 06, UMI 209, Paris, F-75005, France
[3] IFI, MSI, UMI 209, Hanoi, Vietnam
[4] UR AGIRs, CIRAD, Campus de Baillarguet, Montpellier, France
Emails: edouard.amouroux@ird.fr , benoit.gaudou@gmail.com, stephanie.desvaux@cirad.fr, alexis.drogoul@gmail.com

*:GAMA website: http://gama-platform.googlecode.com/

A deeper understanding of this protocol should be gained in the next section where the GAMAVI model is described using O.D.D.

III. ODD'S DESCRIPTION OF THE GAMAVI MODEL

A. ODD: Overview

1) ODD: Purpose

Epidemiologists want to study mechanisms of local spread and persistence of H5N1 in the context of semi-industrialized and traditional poultry sectors in North Vietnam. The purpose of GAMAVI is to investigate and evaluate the importance of various factors, including poultry production, environments, topography, etc., on such phenomena. Specifically, the model is about investigating the relationships between environments (as virus reservoirs) and the poultry production system.

2) ODD: Entities, state variables and scales

The model considers mainly three kinds of entity: poultry flocks, farms and virus reservoirs in the environment: rice fields and ponds.

a) ODD: Entities

(1) ODD: Agents

Poultry flocks represent groups of poultry. They are the main type of entities in our system. These groups hold homogeneous individuals (with respect to their species, age, behavior and location). The infectious status of each individual is held, within the flock, in an "infection matrix".

TABLE I
FLOCKS' STATE VARIABLES

VARIABLE NAME	BRIEF DESCRIPTION
Matrix of individuals	Represents individuals' infection status (SIR) and elapsed time in this status. Species information is added in the backyard case.
Location	Which environment it is in
Species	It can be backyard, chicken, (Muscovy) duck
Number of individuals	Determined at creation and implies the matrix dimension
Production type	Outlet (meat/egg), determines the length of production and seasonality
Housing	Whether the flock is kept indoors all the time or not
Within-flock infection probability	A daily probability for an infected individual to infect each susceptible individuals
BID50*	Quantity of virus to ingest to get 0.5 probability of getting infected
Excretion rate	Quantity of virus excreted per time-step (duck only)
Renewal transmission probability	If the flock is infected it might leave virus within the building after its renewal and thus infect the new flock (0 means there is a building quarantine procedure)

Farms aggregate several flocks and are responsible for the renewal of them (see appendix [9] for its state variables).

(2) ODD: Spatial units

Spatial units are drawn from GIS data (spot images + "vectored" Google maps).

The *village* is a (multi) polygon that contains other polygons representing *farms* and ponds. Rice fields are obtained by a tessellation, into parcels, of the rice field area surrounding the village. Rice-field and ponds are considered as specific *virus reservoir*s.

Virus reservoirs, receive virus from flocks, manage its depletion and allow flocks to be infected by it (see appendix [9]).

TABLE III
VIRUS RESERVOIRS STATE VARIABLES

VARIABLE NAME	BRIEF DESCRIPTION
Location	Coordinates
Volume	Water volume, to compute the concentration of virus
Virus concentration	
Depletion Rate basis	The basic depletion rate
Temperature	Used to make the actual depletion rate vary over time

b) ODD: Scales

The system modeled is the village, which is a few kilometers wide. The smallest elements are farm buildings, (a dozen of meters wide). The simulated time can go up to two years. The discrete time step between two states of the model is set to one hour.

3) ODD: Process overview and scheduling

This section is devoted to the behaviors of the agents. At each step, agents execute their own behavior simultaneously (using the pseudo-parallel schedule of GAMA).

a) Flock

Flocks' main processes are related to infection and are parameterised with species-dependent rates (see sub-models for a detailed description). Excretion and Virus collection (if located in a reservoir) happen hourly.
The daily-based processes involve the computation of direct transmission, the computation of environment mediated infection (in virus reservoirs), the update of the individuals' infectious status, and the natural death computation.
In addition to these processes, movement ones occur but are specific to free-range and backyard which are described in the appendix [9] (*ODD: Details* section).

b) Farm

Farms renew flocks according to a schedule that depends on the production type. This renewal can be instantaneous or include a building quarantine (a few days). The renewal schedule is not necessarily regular and is drawn from input data.

c) Virus reservoir

These agents can receive virus from a flock; they compute the new virus concentration from the flock's excretion. They change this concentration according to temperature dependent equations. The change of season impacts their behaviours; for instance, rice fields only act as reservoirs in the warm season (when they are watered).

B. ODD: Design concepts

In GAMAVI, we do not have anything related to adaptation, objectives, learning or prediction thus we will not fill the corresponding sections.

1) ODD: Emergence

Experts have not obtained all the necessary data yet, but several realistic dynamics have been observed with plausible data. Depending on the scenario chosen, we observe:

- The virus circulating among poultry flocks during the warm season and stocked in virus reservoirs during the cold season

- Faster epidemics with a higher proportion of chicken
- A significant correlation between the number of ponds/rice fields, free-range ducks and the probability for epidemics to last more than a year.

2) ODD: Sensing

Flocks manage their individuals (in respect to their infectious state).

Farms have access to their flocks and the date. Virus reservoirs know the current temperature, and rice-fields the current date.

3) ODD: Interaction

Flocks excrete virus in a virus reservoir and "collect" virus from it. An infected flock can infect a fully susceptible flock whether it is within the same farm or not (global probabilities).

4) ODD: Stochasticity

Backyard flocks' movement is made towards a randomly selected coordinates at a pre-determined distance. Free-range flocks randomly select a free rice field within a radius.

The renewal of flocks concern a random number of individuals (truncated Gaussian), and a semi-random (with a probability dependent on production type) flag for the use of building quarantine.

Whenever a flock gets infected, an infected individual is chosen at random. When a flock collects more than a predetermined level of virus, each individual has a probability of 0,5 to get infected.

5) ODD: Collectives

Farms are collectives of flocks, which are collectives of individuals.

6) ODD: Observation

The model offers have a graphical representation of the system where the environments, the moving flocks and the spread of the virus are represented. The evolution of the number of individuals per infectious status, flocks or restriction per production type, average concentration of virus in the reservoirs, number of infected reservoirs, instantaneous global transmission rate are all available. We plan in a future version to add a representation of the graph of infection.

C. ODD: Details

Details are described in the online appendix [9].

IV. Advantages and Weaknesses of ODD

Building on this example, the literature and our experience of the protocol, we can now present the major advantages and limitations of ODD and make some proposals.

A. Advantages

ODD includes the description of fundamental yet specific characteristics of complex system models such as scales, environment, accounting of both individual and collectives behaviours, etc. This is a major advantage over non-dedicated languages, like, for instance, UML. In GAMAVI, specifically, the collectives and the environment are key elements of the models.

The representation of both environments and entities include qualitative aspects that would be difficult to mention in traditional software engineering languages (for example: quality of data, lack of data, etc.). Consequently, ODD is a good trade-off between domain issues (data, measure, initial application problematic, etc.) and the modelling activity (with entities, hypotheses, simplification, experimentations).

Finally, ODD, although originated in ecology, seems to be appropriate for models of virtually any domain, including, in our case, Epidemiology [3][5].

B. Weaknesses

However, and even though ODD represents a great step for standardized communication of models it is still far from achieving its objective of replication.

1) ODD as a communication protocol

ODD is a textual template and, as such, is inherently ambiguous. Even though it has been acknowledged in [3] and numerous examples have been proposed to clarify its concepts (see annex of [5]). Examples have never replaced specifications, and the lack of a clear meta-model (rather than a list of items) is clearly a drawback.

GAMAVI is a complex model that makes heavy use of field data. It means that, whenever these data are used, domain and model-specific data are described together. The same is true for the "*Purpose*" section, which mixes the context of the study ("why") and the overview of the model ("how").

Collectives are difficult to account for: they should represent individuals grouped into some kind of collection (e.g. a social group [5]). In our model, we consider that flocks and farms are entities although they can be seen as collectives. Actually, farms are aggregation of flocks and have their own behaviors and attributes. On the contrary, a farm could exist without flocks (imagine a farm of seasoned only flocks) but a collective in ODD cannot exist without its constituent.

Additionally, relegating the parameterization of the model to the *sub-models* section is awkward. In most cases, like in GAMAVI, knowing the parameters helps in understanding the purpose of the model and the question it is designed to answer.

The redundancy of information and the absence of a clear policy of cross-references are other important problems of ODD. As a matter of fact, a lot of information can be placed in one section or another (like behaviors in "*Entities*" or "*process overview and scheduling*", or reservoirs, which belong at the same time to "*Spatial units*" and "*Entities*").

With this description in ODD, our goal was also to obtain a self-contained description that could be reused, for instance, as is, in larger epidemiological models. However, in the "sub-models" section, ODD does not mention (and so does not require) anything related the linking of sub-models in terms of organization, dataflow and synchronization.

2) ODD as replication protocol

In the perspective of model re-implementation and results replication, these ambiguities and repetition are even more problematic. Moreover, in our case, and after having filled all the required sections, several characteristics of our model and simulation have not been presented.

The first and foremost one is probably to state the policy used for the generation of random numbers. As our model is highly stochastic this is mandatory but it is not required anywhere in ODD.

Second, scheduling, although quickly mentioned, is not highlighted enough and can have also serious implications. GAMAVI relies on the GAMA pseudo-parallel scheduler, where, for example, no agents are added or deleted from the system during a step. But what would be the results if we decided to port the model to another platform? Meurisse [10] showed on a simple model that the scheduling strategies of the platform could totally change the result of a model even when it is clearly specified.

Third, floating point arithmetic [11] is not considered at all. In our model, the probabilities of direct are as low as 10^{-2} but for other disease rates can be much lower and thus the way the floating point rounding is executed can impact the system's dynamics. Another example lies in the use of GIS data, where some errors of rounding can dramatically impact the environment of the agents.

It is fair to say, then, that ODD only take implementation issues into account in a marginal way. We think that ODD, although it is now one of the most interesting proposal for the communication of models, needs to seriously pay attention to the issues pointed above if it is intended to allow for their replication.

C. Proposals

1) Disambiguation

It would be necessary to explicitly include a "*Context"* section alongside "*Purpose"* for non-field-specialists to understand all the ins and outs of a model.

The use of a more formalized language from software engineering, perhaps a derivative from UML (like Agent UML [12]), should be considered instead of relying on free text. Especially, the processes and interactions would clearly beneficiate from such standardization, as well as the description of complex data structures.

Finally, the sub-models are just, now, an unorganized list of models. A formalism such as DEVS [13] could greatly help in organizing the relationships among themselves and their host model.

2) Updates to the template

Adding an *Experimentation* section appears mandatory, at least for "case-based" models. This section would regroup: *Emergence*, *Initialization*, *Observation, and Parameterization*. In addition, we think is would be a good thing to add a *Numerical results* and *Pattern results* sections that would be coupled to *Emergence* in a *Result* subsection of *Experimentation*. A *Limits* section could also be envisioned as most if not all models have a defined domain of validity (a part of the parameters space where the model represents adequately the real system).

This *Experimentations* section has mainly two possibilities of structure. On the one hand, in order to be the clearest possible, it could be scenario-based with the actual value of parameters, inputs, etc and results altogether. Obviously, the drawback is the extent of such a presentation. On the other hand, it could be a general *Experimentations* section where all the values are defined with their domain. Major results and/or interpretations (linkage between inputs and outputs) would be added to it.

V. Conclusion

We introduced briefly the ODD protocol and its objectives: model communication and replication. Then, we used this protocol to present our model about H5N1 issues in North Vietnam. From this example and a thorough bibliographical review, we drew the main advantages and weaknesses of ODD. Especially ambiguities in communication and the lack of real specifications, which practically prevent the re-implementation of complex models and the replication of their results.

Our perspectives are now twofold: (1) proposing and having the community validate the addition of an "*Execution environment*" section for the sake of model replication. Such a work has been started by Railsback et al in [14]. (2) Making a thorough review of the possible contributions of existing software engineering approaches (in terms of data description, processes, methodologies, etc.) and how they could contribute to strengthen ODD – which is, finally, about writing software.

References

[1] V. Grimm, S.F. Railsback, "Individual-Based Modeling and Ecology", Princeton University Press, 2005

[2] S.F. Railsback, V. Grimm, "A Course in Individual-based and Agent-based Modeling". http://www.railsback-grimm-abm-book.com/ 2010

[3] J.G. Polhill, D. Parker, V. Grimm, "Using the ODD Protocol for Describing Three Agent-Based Social Simulation Models of Land-Use Change". JASSS (2008) vol. 11 (2 3) pp. 1-30

[4] Piou, U. Berger, V. Grimm, "Proposing an information criterion for individual-based models developed in a pattern-oriented modelling framework". Ecological Modelling (2009)

[5] V. Grimm et al "A standard protocol for describing individual-based and agent-based models". Ecological Modelling 198 (1-2), 115-126. 2006

[6] S.J. Mellor, S. Kendall, A. Uhl, D. Weise, "MDA Distilled". Addison Wesley Longman Publishing Co., Inc. 2004.

[7] E. Amouroux, S. Desvaux, A. Drogoul: Towards Virtual Epidemiology: An Agent-Based Approach to the Modeling of H5N1 Propagation and Persistence in North-Vietnam. PRIMA08: 26-33, 2008

[8] E. Amouroux, C. Thanh-Quang, A. Boucher, A. Drogoul, "GAMA: An Environment for Implementing and Running Spatially Explicit Multi-agent Simulations". PRIMA07: pp. 359-371, 2007

[9] Online appendix: http://code.google.com/p/gama-platform/downloads/detail?name=rivf_annex.pdf&can=4&q=

[10] T. Meurisse, "Simulation multi-agent: du modèle à l'opérationnalisation", PhD thesis, 2004

[11] G. Polhill, L. Izquierdo, N. Gotts, "The ghost in the model (and other effects of floating point arithmetic)", JASSS vol. 8 (1), 2004

[12] B. Bauer, J. Muller, J. Odell. "Agent UML: A Formalism for Specifying Multiagent Interaction", AOSE, pp. 91-103, Springer-Verlag, 2001

[13] B. Zeigler, T. Gon Kim, H. Praehofer, "Theory of Modeling and Simulation", Academic Press, 2000

[14] S. Railsback, S. Lytinen, V. Grimm, "StupidModel and Extensions: A Template and Teaching Tool for Agent-based Modeling Platforms" Technical report, condor.depaul.edu (2005)