A Comparison of Individual-Based and Dynamic Modelling using Bullant

W. J. Chivers[a], R. D. Herbert[b] and S. Keen[c]

[a] *School of Business, The University of Newcastle, Ourimbah, NSW 2258, Australia (wchivers@mail.newcastle.edu.au)*

[b] *School of Business, The University of Newcastle, Ourimbah, NSW, 2258, Australia (ric.herbert@newcastle.edu.au)*

[c] *School of Economics and Finance, University of Western Sydney, Penrith South, NSW 1797, Australia (s.keen@uws.edu.au)*

Abstract: The Bullant programming language is used to compare an individual-based approach to modelling two populations with a more traditional differential equation-based approach. The Lotka-Volterra equations describing the interaction of a predator and prey species and an individual-based simulation of the two species are implemented and compared in this new language. In comparing the two models attention is given to the different approaches, the model parameters, their complexity and their deterministic or stochastic nature. We find that the individual-based approach can be used to specifically model individual environmental parameters and follow individuals through a simulation, but at the cost of a much higher level of complexity. The new programming language Bullant is found to be highly suitable for building the models described, and very promising for the construction of more complex models.

Keywords: Agent-based modelling; Equation-based modelling; Lotka-Volterra equations; Bullant programming language.

1. INTRODUCTION

In the past decade individual-based modelling has become an important tool in the modelling of systems characterized by large numbers of discrete elements [Grimm, 1999; Van Dyke Parunak, Savit and Riolo, 1998]. Individual-based modelling is an alternative to the more traditional dynamic or equation-based approach in which ordinary or partial differential equations are used to predict system outcomes over time.

An individual-based model (IBM) consists of a set of heterogeneous discrete objects which change their state over time. Execution of the model is effected by simulating local interactions between individuals and the time-varying but heterogeneous environment. The majority of IBMs are hand-coded [Lorek and Sonnenschein, 1999] and are synonymous with their computer implementation; the model is often not implemented or even described in any other way. The foundation of individual-based modelling is the assumption that if the individual elements are accurately modelled, then the system-level outcomes will be accurate.

In contrast equation-based modelling takes a top-down approach, seeking to express relationships between global outcomes. Global outcomes are modelled directly in a set of equations, and execution consists of evaluating the equations over time, often in discrete time intervals. Populations, rather than individuals, are modelled.

In order to compare these two approaches we start with a well-known equation-based model, the Lotka-Volterra predator-prey model of the interaction of two species [see, for example Adler, 1998]. This model is implemented using the Bullant language. We then build an individual-based model of a predator species and a prey species also using the Bullant language. We do not attempt here to model real species, meaning that some arbitrary decisions about species behaviour are necessary. Nor do we attempt to build two models which produce identical results, as is done by Wilson [1998],

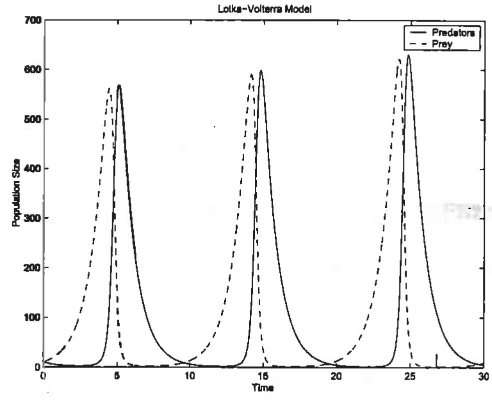

Figure 1. Lokta-Voltera Model.

who modifies his equation-based approach to reflect his IBM results. Instead we compare the two approaches in general terms while evaluating Bullant as a programming tool for building models. Bullant is also briefly compared with other IBM modelling tools.

We find that the individual-based approach may result in a more intuitive model than does the dynamic approach because the former allows significant environmental variables to be specifically modelled. It also allows individuals to be followed through a simulation. The implementation of the individual-based approach is, however, far more complex than the implementation of the Lotka-Volterra equations. The Bullant language is found to be well-suited to building these models from scratch. Bullant offers many features of interest to model builders and promises to be very useful for the construction of complex individual-based models involving very large numbers of individuals.

2. MODELLING POPULATIONS

The coupled set of differential equations developed by Lotka and Volterra in 1925 [Adler, 1998] are used here to implement an equation-based approach to modelling the population changes over time resulting from the interaction of a predator species and a prey species. The equations model the rate of growth of the prey as dependant on interactions with predators (the prey have unlimited resources such as food), and the rate of growth of the predators dependant on the availability of prey. The model is parsimonious and deterministic.

The equations are:

$$\frac{db(t)}{dt} = [\lambda - \epsilon p(t)]b(t) \qquad (1)$$

$$\frac{dp(t)}{dt} = [-\delta + \eta b(t)]p(t) \qquad (2)$$

Where b represents the population of prey, p the population of predators, λ the prey growth rate in the absence of predators, ϵ the chances of a prey object being eaten, δ the predator growth rate in the absence of prey and η the chances of a predator eating. The initial population levels are known. We use the Euler method to implement an ODE solver in the Bullant language.

The assumptions on which the Lotka-Volterra model is based include the following:

- The rate of growth in each population is a) constant relative to its own population size, and b) proportional to the other population size.
- Prey never escape an encounter with predators: escape and hunting strategies are not modelled.
- Homogeneous distribution of both species, which encounter each other at a rate proportional to the product of the two population sizes.
- Predator behaviour is unchanged by a meal.
- Prey have unlimited resources and grow exponentially in absence of predators. Their only cause of death is being eaten.
- The predators have unlimited resources apart from food.
- Only two species are interacting.

These assumptions are intrinsic in the four model parameters, which represent many environmental factors and characteristics of the two species. The difficulty in mapping the four model parameters to the numerous environmental factors that they represent is a problem if these individual environmental factors are important to the research being undertaken. While the assumptions can be manipulated by increasing the number of parameters and/or using partial differential equations, the resulting model is often complex and not easy to change [Van Dyke Parunak et al., 1998].

Figure 1 is a graph of the data produced by the Bullant implementation of the Lotka-Volterra model. The model is calibrated with the following parameter values: $\eta = 0.01$, $\epsilon = 0.01$, $\delta = 1.0$ and $\lambda = 1.0$. The simulation was run for 30 time units with a time step of 0.01. The initial number of both predators and prey is 10. The increasing size of the population maxima is due to the truncation errors of the Euler solver.

3. MODELLING INDIVIDUALS

An individual-based approach to modelling involves the modelling of individuals rather than

Table 1. Map of the model parameters

Lotka-Volterra parameter	*Individual-based parameters*
λ	μ, ν, κ, α, ω
ϵ	χ, p/b
δ	μ, θ, ν, κ, α, ω
η	σ, p/b

populations. In the case of the predator-prey species, the individuals are the members of the two species. They have characteristics such as age and hunger level, and behaviour resulting from these characteristics. Here they are modelled by creating objects in the computer memory with corresponding attributes and behaviour. The individual objects in the computer memory, each representing an individual in the species, are able to store characteristics as attributes of the objects and behaviour as program methods of the objects.

The implemented simulator adds the predator and prey objects to linked lists in the computer memory and processes all members of the two lists during each cycle of the simulation. Processing of each predator involves determining whether the predator will eat in this cycle or perhaps die of starvation, and whether it should reproduce in this cycle. Processing of each prey object involves determining whether the prey object will be eaten in this cycle and whether it should reproduce in this cycle. The determination of whether a predator eats or whether a prey object is eaten is affected by the ratio of predators to prey. When either predators or prey die, they are removed from the appropriate list. When reproduction occurs objects are added to the appropriate lists and are processed in subsequent cycles.

The assumptions on which this model is based are largely identical to those on which the Lotka-Volterra equations are based. Differences include the following:

- The rate of growth. The litter size of the predators is proportional to the ratio of the population size of the prey to that of the predators; the litter size of the prey is proportional to the ratio of the population size of the predators to that of the prey.
- Homogenous distribution is assumed as in the Lotka-Volterra equations, but the chances of a predator encountering a prey object here is proportional to the ratio of the population size of the prey to that of the predators and the chances of a prey object being eaten is proportional to the ratio of the population size of the predators to that of the prey.

The effects of these differences are not investigated here. This issue, and the development of an IBM that more closely matches the assumptions of the Lotka-Volterra model, will be the subject of future research.

The following parameters affecting the predators are entered into the simulation before each execution:

- The initial number of individuals (b or p)
- The maximum age of an individual (μ)
- The maximum number of cycles between meals (θ)
- The probability of eating during a cycle (σ)
- The number of cycles between litters (ν)
- The minimum age before reproduction (κ)
- The maximum litter size (α)
- A per capita growth rate parameter used to affect the litter size (ω)

The prey parameters are the same with the exception of θ, which does not exist, and σ, which becomes the environmental parameter affecting the chances of being eaten (χ). For the simulations reported here, μ was disabled in order to keep the assumptions of the model closer to those of the Lotka-Volterra equations.

The parameters σ, χ and ω are environmental variables that represent factors such as the chances of a prey object escaping an encounter with a predator. They are general parameters similar to those in the Lotka-Volterra equations, providing constants with which other parameters are adjusted in the IBM. They would possibly be analyzed and broken down into identifiable parameters if real species were being modelled and their identification were important for the model being developed.

The fifteen parameters (eight for the predators, seven for the prey) are mapped to the four Lotka-Volterra parameters in Table 1.

Each simulation reports on events that occur to each individual during the execution of each cycle, and graphically presents the number of predators and prey per cycle. Reported events include being eaten, eating, starving and having offspring. The unique ID numbers of the parents and offspring of each individual may also be reported. At the completion of each cycle the numbers in each population, the number that died of various causes, the number born and the number of meals are reported.

The graphs in Figure 2 were produced by the Bullant model executing for 70 cycles. Given

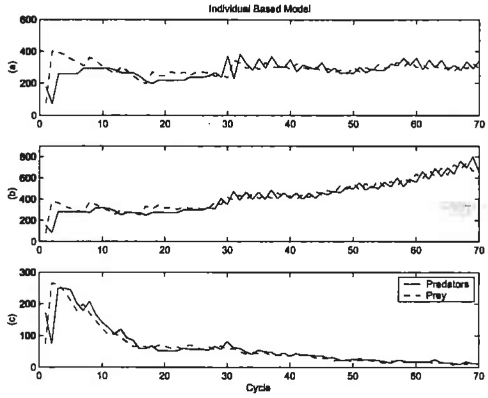

Figure 2. Individual Based Model.

the stochastic nature of this model these results should be presented with upper and lower bounds estimates if conclusions are to be drawn from the results, but this is not done here. These figures are presented as being representative of the results produced by this model only.

Figure 2a is a graph of the data produced by a control simulation with which the other two graphs are compared. Over many cycles it illustrates the predator population responding to a growing prey population by increasing, which in turn negatively affects the prey population and then the predator population.

Lowering κ for the prey from 10 cycles to 7 has the effect of increasing the prey population more quickly, and the predator population is seen to respond in Figure 2b.

Figure 2c illustrates a change to χ. The probability of a prey object being eaten is $p\chi/b$, where χ is an environmental variable reflecting factors such as the prey individual occasionally escaping an encounter with a predator, or of the different behaviour of a hungry predator and a satiated predator. In Figure 2c χ has been increased from 0.09 to 0.12. This results in a trend to extinction of both species.

4. THE TWO APPROACHES COMPARED

In building two different models using Bullant the aim was not to build two models that produce identical results as is done by Wilson [1998]. The results produced by the two models may nevertheless be compared, and they exhibit some similarities and some differences. For example, to produce Figure 2b κ for the prey population is lowered. This has the effect of increasing the prey population, and the predator population follows. The parameter κ is partially mapped in Table 1 to the parameter λ in the Lotka-Volterra model. Increasing λ in the Lotka-Volterra model also increases the prey population and hence the predator population in each cycle. A significant difference, however, is that lowering κ in the IBM produces an overall increase in population numbers over many cycles, but increasing λ in the Lotka-Volterra model does not produce an overall increase in numbers over many cycles.

4.1 The Model Parameters

A major difference between the individual-based approach and the equation-based approach is the method by which the environmental parameters are modelled. The number of environmental parameters may be large, but the latter approach tends to reduce these to a low number of generalized parameters. If the adjustment of individual parameters is not important to the model builder, then the small number of parameters of the equation-based approach may be an advantage. Reducing the many natural parameters which affect a system to a few does, however, increase the difficulty in interpreting or adjusting these parameters. In contrast, individual-based modelling has the potential to explicitly model every relevant parameter more easily than equation-based modelling, facilitating the interpretation and adjustment of individual parameters.

This in turn may facilitate direct experimentation with and verification of the model. For example, litter size or cycles between litters may be directly adjusted in the IBM built here. The importance of making adjustments to individual parameters in some research is emphasized by Van Dyke Parunak et al. [1998]. The individual-based approach is intrinsically better suited to modelling individual parameters and making these adjustments than is the equation-based approach.

4.2 Tracking Individuals

The modelling of individuals with individual parameter values and behaviour allows those individuals to be followed through a simulation, which may be of importance in understanding the dynamics of the simulation or of the system being modelled. The values of individual parameters across a population, or for an individual across time can be examined, as can the relationships between parameters identifiable in an IBM but not so easily identifiable in an equation-based model. Using the individual-based model developed here, for example, it is possible to determine how many individuals in the final population were present in the initial population, and how the maximum number of cycles between meals might affect this number in the predator population. This level of individual detail is not possible with a population-based approach.

4.3 Complexity

The modelling of individuals results in the benefits of more intuitive model parameters and the possibility of following individuals through the simulation, but this comes at the cost of complexity. The program code necessary to implement an individual-based model is far more complex than that needed to implement a differential equation model. Grimm [1999] and Railsback [1999] both report difficulties in the reproduction of realistic system behaviour in individual-based models because of this complexity. Complex program code is problematic because of the associated increase in the chances of errors, difficulty in code verification and difficulty in replication of the code by other researchers.

An associated problem is the difficulty in accurately modelling individual behaviour. Small inaccuracies in the modelling of individual behaviour in an IBM are likely to lead to large inaccuracies in the system behaviour [Van Dyke Parunak et al., 1998]. The identification of all relevant parameters and their accurate modelling are critical to the accuracy of an individual-based model [Railsback, 1999].

4.4 IBM is Stochastic

The equation-based model described here is deterministic. In reality we would expect uncertainty, and the individual-based model developed here does reflect the stochastic nature of real species populations.

5. THE BULLANT LANGUAGE

The Bullant language used to implement both models in this study is a proprietary object-oriented language which forms part of a new commercial Internet client-server application development environment being built in Sydney by the Bullant company (www.bullant.com). The Bullant development environment includes the Bullant language, a proprietary Internet protocol and database technology.

The commercial use of Bullant is in the development of client-server applications with very high numbers of transactions to be run on the Internet. In applications written in Bullant all processing occurs on a server, with the clients only displaying a user interface. The thin protocol and thin client mean that powerful machines and a fast networking infrastructure are not necessary for a modest client machine to display application software that might otherwise require powerful hardware. On the server all data and code are stored in a single memory space, eliminating data transformations and shunting and resulting in higher performance. These features of Bullant suggest that it will be well-suited to the development of complex simulations involving very high numbers of individual objects. These simulations could then be displayed remotely by large numbers of clients using modest hardware (a powerful server would be necessary for this, however).

The pre-coded libraries of Bullant are similar in many ways to those of Java, even to the extent of giving similar objects similar names. For example, the Bullant GUI layout manager classes include a BorderPanel class and a FlowPanel class. The Canvas class was used here to create graphical output from the models. Anyone experienced with the Java or C++ GUI tools will find the Bullant GUI libraries very familiar. It is worth noting that the Bullant documentation and language does assume some experience with modern object-oriented application development.

In order to use Bullant for our purposes on a single machine, we needed the console (an interface with the kernel) and the remote client applications. Both are freely available from Bullant and may be run simultaneously on a single machine. Bullant markets rapid application development tools such as Bullant Builder which are not freely available, but are not absolutely necessary for model building. The alternative to these RAD tools is to code the application by hand, which was done here. The rich set of freely available pre-coded libraries and documentation facilitated this task. Our applications included graphical user interfaces and standard data structures. The libraries also include classes for persistent data, secure transactions and the creation of individual tasks which support multiprocessing as well as many other common features of modern object-oriented languages (but not used for the models developed here).

In order to build an IBM the researcher must either build the model from scratch or use a modelling tool such as WESP or a specific simulator built for one type of problem [Lorek and Sonnenschein, 1999]. Building a model from scratch affords ultimate flexibility in the architecture of a model, at the cost of increased difficulty for non-programmers. Tools such as Ecosim and Swarm are built for this type of development and include class libraries specifically built for the development of individual-based models [Lorek and Sonnenschein, 1999]. Bullant does not have the advantage of such specific purpose libraries, but other features of Bullant suggest to us that it may prove very useful as a modelling tool. As already mentioned, the Bullant memory model is designed to execute very large numbers of transactions quickly and Bullant can be used to develop network client-server applications. The following features of Bullant are also very promising for the construction of complex IBMs involving multiprocessing, concurrency or object history:

- Bullant keeps a history of previous states of all objects. An entire Bullant program, or a part of one, can be frozen and wound backwards or forwards transaction-by-transaction.
- Bullant tasks are highly suitable for use in representing objects to be modelled, especially in multiprocessor systems. Tasks are uniquely identified and can be created or destroyed in real time.
- The implementation of secure transaction blocks in the code is highly automated.

Although this language is not being developed with IBMs as a primary goal, we found that the pre-written libraries made the development of the models described here, including their data structures and GUIs, very straightforward. We expect that Bullant will prove to be a very useful tool for the construction of complex IBMs involving very large numbers of interacting individuals in future research.

6. CONCLUSIONS

Two approaches to the building of a model which simulates the interaction of two species are compared in this paper using a new programming language. The aim was not to develop two models that produce identical results, but to compare the models in general terms and to evaluate the Bullant language. In comparing the two approaches to modelling, we find that if the manipulation of individual environmental parameters or the tracking of individuals through a simulation are important to the research, then the IBM approach to modelling has advantages over the equation-based approach. These advantages, however, come at the expense of increased complexity. This complexity has the potential to introduce problems of reliability, validity and documentation of the model.

The Bullant language is found to be a competent tool for model building, although this is not a primary goal of its development. Bullant does not include class libraries specifically developed for the construction of IBMs, but we found that the existing Bullant class libraries were effective in the development of these models. The models built here could have been built in other languages and were not large or complex enough to make use of the features of Bullant that we feel are very promising for the construction of large and complex individual-based models. We expect that Bullant will be a very useful tool in the development of such models.

7. ACKNOWLEDGEMENTS

The insightful comments of Rod Bell (Macquarie University) and an anonymous referee are gratefully acknowledged.

8. REFERENCES

Adler, F. R., *Modeling the Dynamics of Life: Calculus and Probability for Life Scientists*, Brooks/Cole, 1998.

Grimm, V., Ten years of individual-based modelling in ecology: What have we learned, and what could we learn in the future?, *Ecological Modelling*, **115(2-3)**, 129–148, 1999.

Lorek, H. and Sonnenschein, M., Modelling and simulation software to support individual-based ecological modelling, *Ecological Modelling*, **115**, 199–216, 1999.

Railsback, S., Complex Adaptive Systems Meets the Real World: Making Agent-Based Simulation Work for Ecological Management and Research. Seminar presented to the Santa Fe Institute and the Center for Nonlinear Studies at Los Alamos National Laboratory, 1999.

Van Dyke Parunak, H., Savit, R. and Riolo, R., Agent-based modeling vs. equation-based modeling: A case study and users' guide. In Proceedings of Multi-Agent Systems and Agent-Based Simulation (MABS'98), 10-25, Springer, LNAI 1534, 1998.

Wilson, W., Resolving discrepancies between deterministic population models and individual-based simulations, *American Naturalist*, **151(2)**, 116–134, 1998.