iSimPlus[1]: A Multi-Method Simulation Tool for Modelling Complex Systems

Jim Duggan,

School of Engineering and Informatics,

National University of Ireland, Galway.

Email: jim.duggan@nuigalway.ie

Abstract

Simulation provides a means to gain insight into the past behaviour and future trajectories of complex social systems. The simulation process is one of discovery: individual mental models are communicated, formalised, and simulated under a range of scenarios. There are two main approaches to social simulation: system dynamics, centred on the feedback perspective, and agent based computational modelling, which uses the individual and their interactions as the basic building block. This short paper describes a new simulation tool that can accommodate both perspectives, and where all model building is achieved using an equation-based approach. The system design is summarised and an example based on the SIR model is described.

Keywords: System Dynamics – Agent-Based Modelling – Integral Equations – Markov Decision Processes

Introduction

In recent years, there has been a noticeable increase in the discussion of agent methods within the system dynamics literature, and also the annual conferences. Rahmandad and Sterman (2008) write that both approaches "should be viewed as regions in a space of modelling assumptions, and not as incompatible modelling paradigms." Osgood (2009) summarises a list of over 20 system dynamics based publications that have adopted individual-based formulations, and describes a number of classes of problems where individual-based models have proven attractive. Among those includes Schild's (2004) observation that to certain stakeholders, the rules governing individual behaviour can sometimes be easier to formulate than those associated with aggregate flows.

The role of stakeholders is an important factor, and this point is discussed by Sterman (2000), who comments that "as a rule of thumb, clients want to see more detail in a model that the modeller thinks in needed," and modellers "generally overestimate the detail necessary to capture the dynamics of interest." He references Roberts' (1978) view that "you must provide the level of detail that causes [the client] to be persuaded that you have properly taken into account his issues, his questions, his levels of concerns. Otherwise he will not believe the model you have built, he will not accept it, he will not use it." The common theme from these views is that more detail, which often results in increased disaggregation, heterogeneity and consideration of discrete events, can be important to the client, and that it has can have a positive effect of building their confidence in the model.

[1] The term iSimPlus stands for **i**ndividual **sim**ulator **plus**, where the plus represents additional planned functionality including loop dominance analysis and novel optimisation methods

The role of discreteness in system dynamic's models was also discussed by Forrester (1961), who wrote that "the study of individual events is one of our richest sources of information about the way the flow channels in our model should be constructed," and that "discreteness of events is entirely compatible with the concept of information feedback systems." In relation to the model building process, he concludes that when "a model has progressed to the point where such refinements [discreteness] are justified, and there is reason to believe that discreteness has a significant influence on system behaviour, discontinuous variables should then be explored to determine their effect on the model." However, he does caution that "we must be on our guard against unnecessarily cluttering our formulations with details of discrete events that only obscure the momentum and continuity exhibited by our [industrial] systems."

Agent-Based Modelling, with its focus on discrete representations of individuals on connected networks, provides a complimentary perspective for modelling complex systems, and this approach is rooted in the idea of *the generativist's experiment* (Epstein 2006), which is to:

> "Situate an initial population of autonomous heterogeneous agents in a relative spatial environment; allow them to interact according to simple local rules, and thereby generate – or "grow" – the macroscopic regularity from the bottom up."

Epstein also regards agent-based modelling as reductionist, as "it is precisely the generative sufficiency of the parts (the microspecification) that constitutes the whole's explanation." In the field of social simulation the agent approach provides the means to capture the way people behave and interact, and then to observe the macroscopic effect of these interactions. Agent models allows for models to be built where individual entities and their interactions are directly represented (Gilbert 2008).

An important element of agency is the spatial environment that situates individuals in a location, and allows them to interact with their neighbours at a local level. Many successful formulations of interaction structures have been developed (Rahmandad and Sterman 2008), from the grid-based neighbourhood structures (Schelling 1971), to graph-based models such as small world (Watts and Strogatz 1998), scale free (Barabasi and Albert 1999), lattice, random and fully connected. In addition to the spatial attribute of agent models, Epstein (2006) presents other significant properties of agents as:

- **Heterogeneity**, where, rather than individuals having similar properties and being compartmentalised into groups, they are represented with individual traits. For example, in a model of epidemics, individuals could have differences could spread across factors such as sociability, resistance to infection, and recovery delay.

- **Autonomy**, where there is no explicit "top-down" control over individual behaviour, although it is recognised that some form of conditioning by either social norms or societies will shape agent behaviour.

- **Bounded Rationality**, where agents only have access to their own local worldview, and typically make use of simple rules based on this local information.

Agent based models, in common with system dynamics applications, span a range of disciplines (see Heath et al. (2009) for a longitudinal survey of the field), including epidemiology, supply chain analysis, consumer dynamics, retirement planning, urban models, guerrilla warfare, drug use, generating ancient civilisations, electricity markets, and innovation networks. These agent models, while focused on the individual and interaction-level, are also concerned with the macroscopic implications that emerge out of individual actions. For example, in Epstein's (2006) smallpox model, results are presented from the individual perspective, as is the overall aggregate dynamic which replicates the classic bell shaped curve associated with the growth and decline of a virus in a susceptible population.

While the agent approach shares a common goal with system dynamics – the explanation of a macroscopic behaviour over time – its process for generating that outcome pays specific attention to individual behaviours and interactions. Because of this focus, much of the agent-based literature tends to view system dynamics as being quite distinct. For example, Parunak (1998) write that "in many domains, agent-based modelling competes with equation-based approaches," and Gilbert (2008), concludes that system dynamics is "good for topics where there are large populations of behaviourally similar agents," and that modelling heterogeneity "becomes extremely cumbersome with more than a few different types." However, the discussion within the system dynamics community tends to focus more on the similarities between the approaches, which indicate the potential for an integrated and unified perspective.

This paper presents a new approach to combining system dynamics and agent-based simulation, one that is entirely grounded in an equation approach. This differs from the usual method for modelling agent systems (North et al. (2006), Borshchev et al. (2004)) through an object-oriented oriented programming paradigm. By focusing on equation building, our simulation tool provides system dynamics users with a framework where they can extend equation based models, and situate these on a range of interconnected networks. This supports disaggregation of models, and also allows for the representation of discrete variables, when can then be easily aggregated and compared with their continuous flow counterparts. The overall contribution is to provide an environment where model builders and stakeholders can transition easily from aggregate to disaggregate models, and therefore gain insight into the benefits and drawbacks of these approaches for the problem under study. The remaining sections of the paper include: a summary of the system architecture; a case study based on a diffusion process; and future work.

iSimPlus Architecture

The key ideas underlying the design of iSimPlus are shown in figure 1. At a modeling level, the client and model builder can abstract the problem into (1) aggregates, (2) individuals and (3) social networks. Each of these three building blocks are now described.

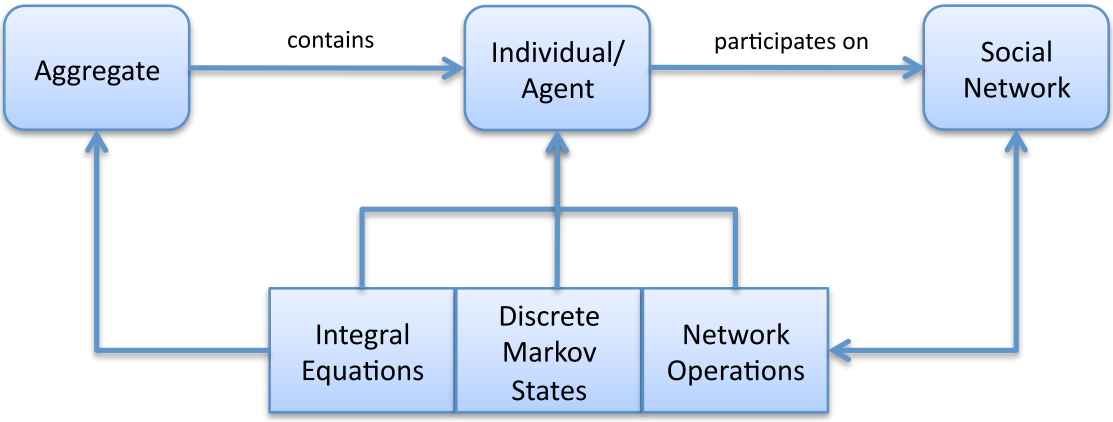

Figure 1: iSim+ conceptual design

1. **An aggregate** is a conventional SD model (integral equations) or set of models, and would correspond to a *sector* or a *view* in tools such as Stella and Vensim. Aggregates also have access to aggregator functions, which can be used to calculate the sum of all individual states across a given network.

2. **An individual** (or agent) usually models a person that interacts with neighbours on a network, but it can also represent other relevant abstractions such as an organization (e.g. a sector in the Beer Distribution Game). An individual be formally modeled using two different equations approaches. For continuous variables, for example, the energy levels of an individual, the standard stock and flow notation can be deployed. However, for discrete states that have binary values that are either "on" or "off", a different mechanism is required. Because a key requirement of our approach is to minimize computer programming, Markov chains were selected as a means for representing multiple states, where the transition between states is governed by probabilities, which in turn can depend on the continuous variables in the model.

   An example of a Markov chain is captured in figure 2. Using conventional notation, discrete states are represented as circles, and transition mappings are shown as connections between circles. Probabilities, for example, [(S_1, S_2),(S_1, S_1) (S_2, S_1) (S_2, S_2)] are associated with transitions, and these numbers [0,1] are used to drive a stochastic process, implemented using a roulette-wheel mechanism, which determines whether or not a state will change. At all times the sum of all states for a given group is 1, and only one state can be active at any one time. A timer transition is also available, which fires after a specified time has elapsed, and therefore state transition delays using an exponential distribution can also be modeled (e.g. when a patient recovers from an illness in an SIR-type model).

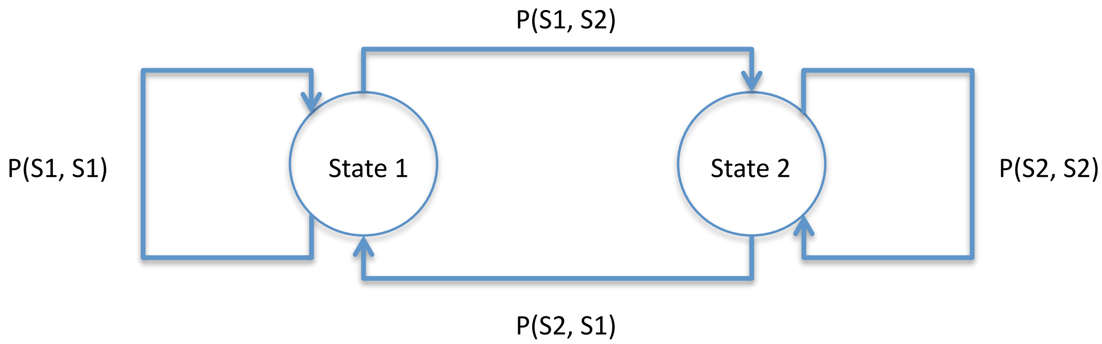

Figure 2: A Markov Chain Model for iSimPlus

Individuals can share information with their neighbours using a number of pre-defined functions, which include:

NetGetNeighbourhoodProportion(NetID, Variable)	Returns the percentage of neighbours in the given state.
NetGetNumberOfNeighbours(NetID)	Returns the number of neighbours for a given agent

Additional network functions can be added through an external API, in order to support increasing levels of requirements that domain models might require.

3. **Social networks** model the interaction structures for collections of individuals, and are parameterized based on the network type. Currently, there are three types of networks available: small world, scale free and random. Individuals can participate on any number of networks, and therefore their individual states and stocks can depend on the state of their surrounding network. This mechanism facilitates experimentation according to the previously mentioned *generativist's experiment* (Epstein 2006), and allows the modeller to locate a population of individuals in a spatial environment, facilitate their interaction through simple rules, and from this "silicon Petri dish", allow the macroscopic structure to emerge.

Network structures can also be visualised to convey spatial relationships, individual states, and the degree of connectivity. For example, in figure 3, two distinct topologies are captured. On the left, characterised by its clustering and similar average number of connections, is a small world network. To the right is a scale free structure with preferential attachments, where the “rich get richer” and a positive feedback process drives new nodes to seek out nodes that are highly connected.

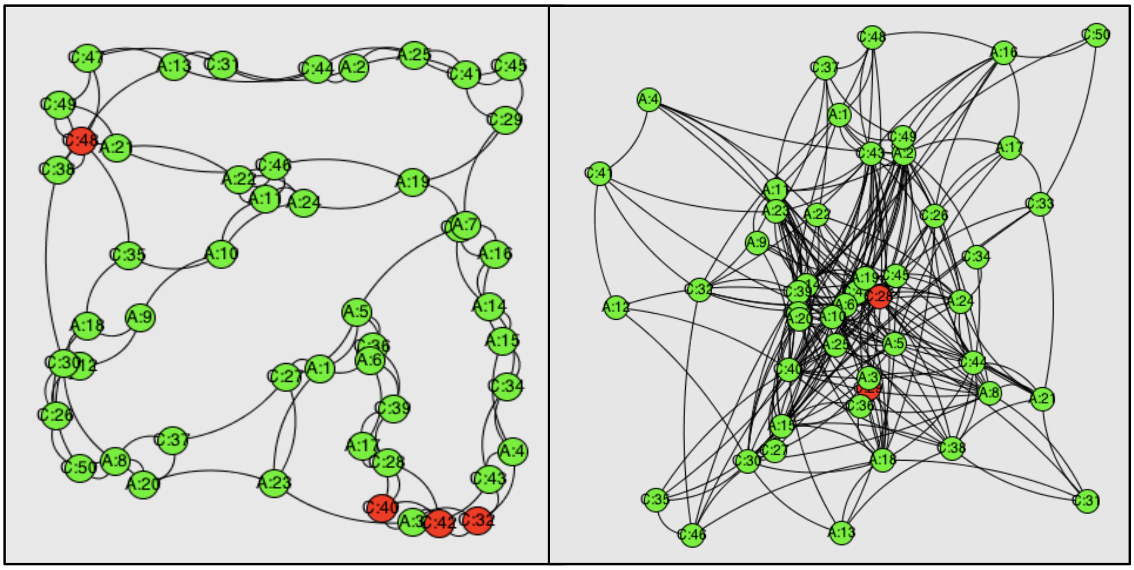

Figure 3: iSimPlus sample network topologies: small world (left) and scale free (right)

For many problems, the choice of network structure and parameters is driven by the problem at hand, and is an empirical choice. We now present an example of an equation-based individual model, and sample output from a number simulation runs.

Worked Example

The choice of model is the diffusion of infectious diseases, and is the SIR originally model published by Kermack and McKendrick (1927). Our motivation for selecting this model is informed by the choice of Rahmandad and Sterman (2008), who chose a variant of this problem (SEIR model) to compare diffusion dynamics between agent based and differential equation models. In their paper, they support the choice of model based on the following criteria: (1) the SEIR model contains important characteristics of complex systems, such as positive and negative feedbacks, time delays, nonlinearities, stochastic events and agent heterogeneity; (2) network topological structures connecting individuals have a significant role to play in the diffusion process; (3) the differential and agent based paradigms both have a history and a wealth of literature of existing models; and (4) diffusion is a fundamental process in different physical, social, biological and economic situations. The markov processes for the SIR agent are shown in figure 4, along with the transition probabilities. A number of points are worth mentioning:

- All the probabilities are declared using equations and with access to in-built functions;
- For P(S,I) the value *proportion infected* is calculated from a network routine, while the value for *infectivity* is a stochastic value sampled from a uniform distribution, and is different for each agent.
- For D(I,R), this is modeled using a delay function drawn from a random exponential distribution. The delay function is not probabilistic, so all agents, once infected, will eventually recover.
- Agents who transition to a recovered state will remain in that state, as there is only on transition from R (which is to itself).

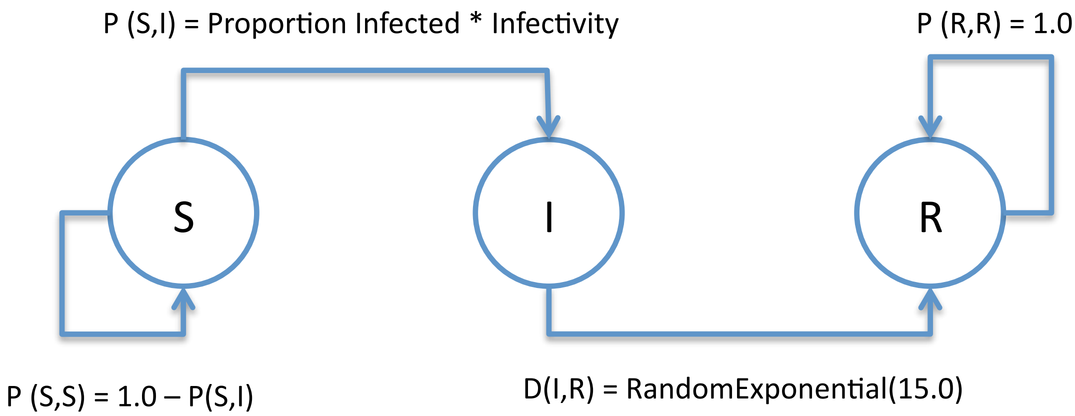

Figure 4: Markov processes for SIR model

The full specification of the model is contained in Appendix A. Currently, iSimPlus supports a range of common SD functions, including delays, random number generators, mathematical functions, and common network operators. An aggregate model is also specified, and its equations (1,2,3, 4 and 5) are shown below (these use simple aggregators to monitor the population-level variables).

(1)	aux STot = SUM(S);	Sum over all agents for state variable S
(2)	aux ITot = SUM(I);	Sum over all agents for state variable I
(3)	aux RTot = SUM(R);	Sum over all agents for state variable R
(4)	aux ChildITot= SumByAgent(Child, I);	Sum over Child agents for state variable I
(5)	aux AdultITot = SumByAgent(Adult, I);	Sum over Adult agents for state variable I

The simulation can be run in flight-simulation mode, which delays the algorithm so that the changing network structures can be observed by the user. An open-source network visualiser (JUNG) is used to show the individuals, and these can be configured for a number of structures[2]:

- KKLayout - The Kamada-Kawai algorithm for node layout;
- FRLayout - The Fruchterman-Rheingold algorithm;
- SpringLayout - A simple force-directed spring-embedder;
- ISOMLayout - Meyer's "Self-Organizing Map" layout.
- CircleLayout - A simple layout places vertices randomly on a circle.

For our illustrative example, the CircleLayout is used, and output is shown in figure 5, where green = susceptible, red = infected and blue = recovered.

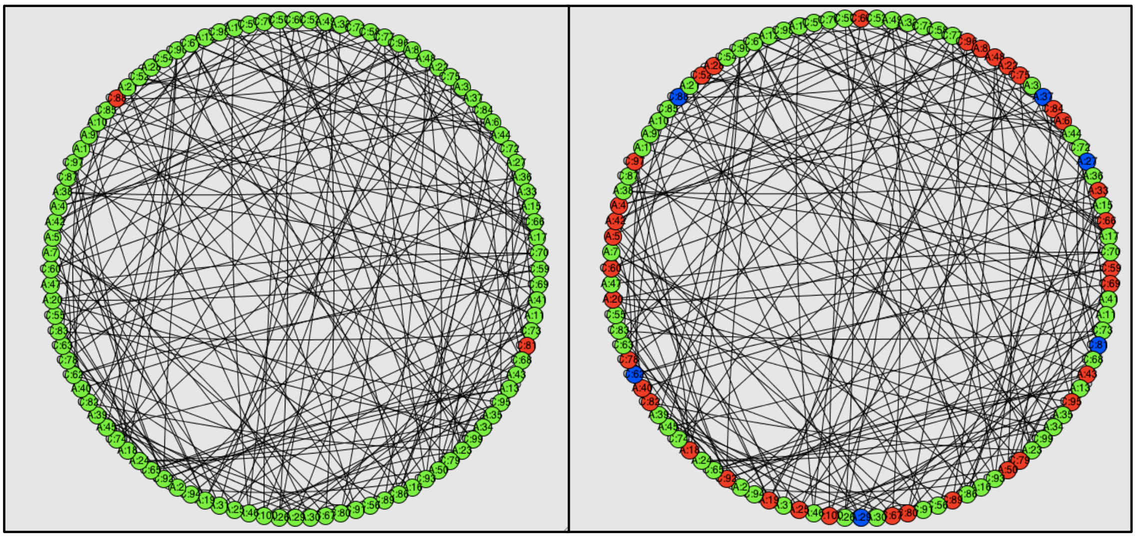

Figure 5: SIR Model before (left) and after (right) a single simulation run

Comprehensive data analysis on all aspects of the model is also possible, as detailed data on each simulation run is stored in CSV files, therefore a full behaviour-trace is available for an individual agent (see table 1), where the values of the state transition probabilities can be viewed. For example, the probability of becoming infected increases as the number of infected neighbours increase, and this is clearly shown in figure 6. The transition probability equals the infectivity when all neighbours are infected (which is consistent with the model formulation). Also, the delay variable D1 is shown as a constant value (although it is different for each agent), and this is achieved through the specification of an special auxiliary variable (keyword *param*), which generates a random value at initialisation stage, and maintains this value for the entire simulation run. This is shown in detail in the agent specifications contained in Appendix 1.

[2] http://jung.sourceforge.net/applet/showlayouts.html

TIME	ID	S	I	R	PROPINFECT	INFECTIVIT	P1	P2	D1
0	ADULT:109	1	0	0	0.25	0.657865	0.164466	0.835534	8.429047
1	ADULT:109	1	0	0	0.25	0.657865	0.164466	0.835534	8.429047
2	ADULT:109	1	0	0	0.5	0.657865	0.328932	0.671068	8.429047
3	ADULT:109	0	1	0	0.5	0.657865	0.328932	0.671068	8.429047
4	ADULT:109	0	1	0	0.75	0.657865	0.493399	0.506601	8.429047
5	ADULT:109	0	1	0	1	0.657865	0.657865	0.342135	8.429047
6	ADULT:109	0	1	0	1	0.657865	0.657865	0.342135	8.429047
7	ADULT:109	0	1	0	0.75	0.657865	0.493399	0.506601	8.429047
8	ADULT:109	0	1	0	0.75	0.657865	0.493399	0.506601	8.429047
9	ADULT:109	0	1	0	0.75	0.657865	0.493399	0.506601	8.429047
10	ADULT:109	0	1	0	0.75	0.657865	0.493399	0.506601	8.429047
11	ADULT:109	0	1	0	0.75	0.657865	0.493399	0.506601	8.429047
12	ADULT:109	0	0	1	0.75	0.657865	0.493399	0.506601	8.429047
13	ADULT:109	0	0	1	0.75	0.657865	0.493399	0.506601	8.429047
14	ADULT:109	0	0	1	0.75	0.657865	0.493399	0.506601	8.429047
15	ADULT:109	0	0	1	0.75	0.657865	0.493399	0.506601	8.429047
16	ADULT:109	0	0	1	0.5	0.657865	0.328932	0.671068	8.429047
17	ADULT:109	0	0	1	0.5	0.657865	0.328932	0.671068	8.429047

Table 1: Sample numerical output from the simulation

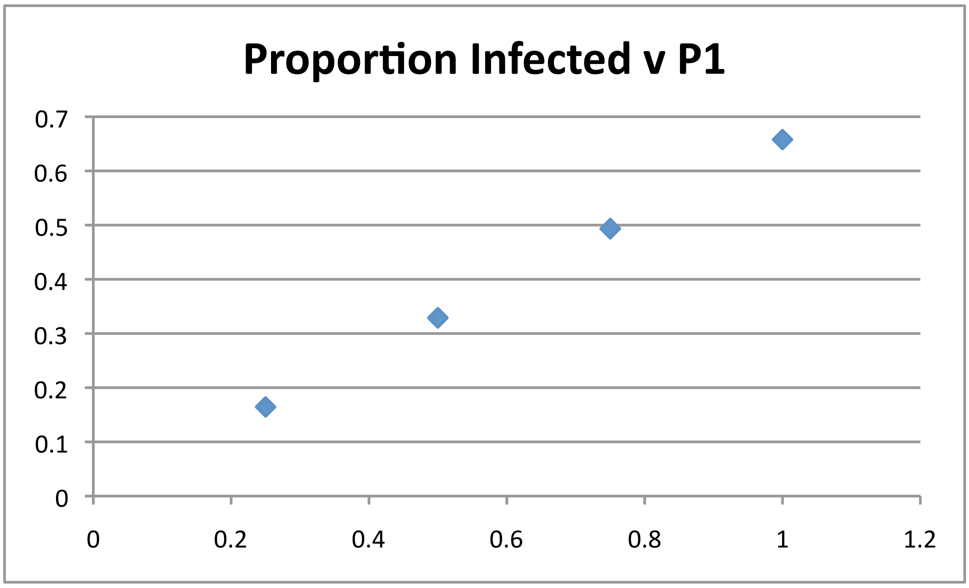

Figure 6: relationship between proportion infection (x) and transition probability (y)

In addition to the network representations and individual data, aggregate results are also available, which indicate either aggregate variables (i.e. stocks) or summations over entire populations (i.e. the macroscopic view of the individual states and behaviours). Sample behaviour over time graphs are shown in figure 7, and these capture the aggregations of discrete states in the model, and show, for a particular simulation, how the numbers of susceptible, infected and recovered change over time.

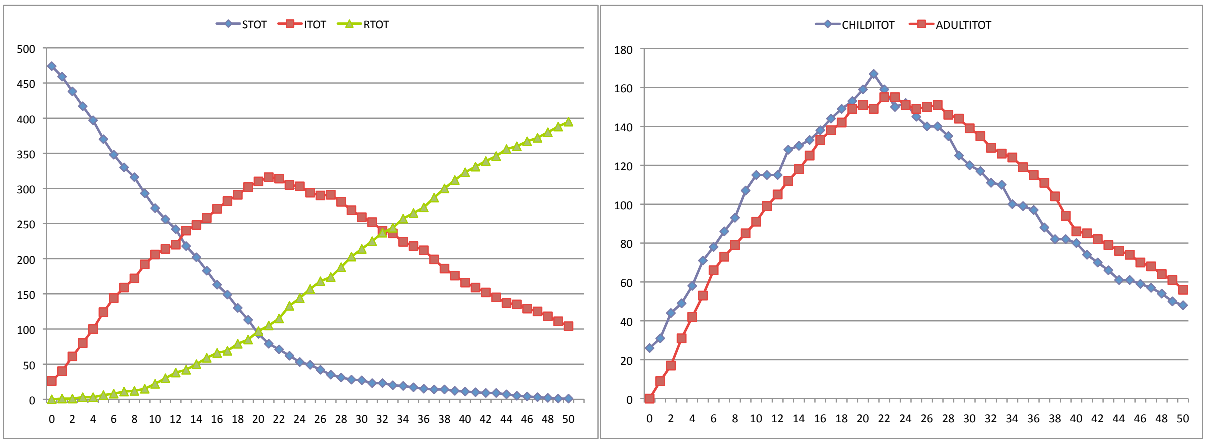

Figure 7: Aggregate output from a single simulation run

To summarise, this section has demonstrated, through a diffusion-based example, how discrete models can be constructed, and their results communicated. For such stochastic models, a significant amount of sensitivity runs are required, and our system is being extended in order to support this. However, what we have shown here is that an equation-based approach, based on the core constructs of *aggregate, individual* and *network* can be used to quickly construct disaggregate models and so provide a richer perspective for experimentation and validation for clients. The following section concludes with a discussion on the benefits of this approach.

Discussion

An important question that merits discussion is the utility of iSimPlus, given that there are a number of agent-based modeling tools already available and in use. In order to address this, it is important to consider the model development process, and the earlier identified need to provide clients with "the level of detail that causes [the client] to be persuaded that you have properly taken into account his issues, his questions, his levels of concerns" (Roberts 1978).

Figure x summarises the iterative nature of modeling, where "the results of any step can yield insights that lead to revisions in any earlier step" (Sterman 2000). As stakeholders and clients cycle through this process, models may be refined or disaggregated, depending on the requirements. For example, if a company were building a consumer behaviour model, the client might well like to explore the impact of a policy on a particular customer segment, or might well request a scenario that takes a contact network structure into account. This approach can support such a process, where a model builder can "drill down" to discrete models (i.e. where individuals have states), and compare and contrasts these results with aggregate formulations. An advantage of this is that aggregate models can be benchmarked against their disaggregate counterparts, and this can help build confidence in the higher level aggregate and feedback structures.

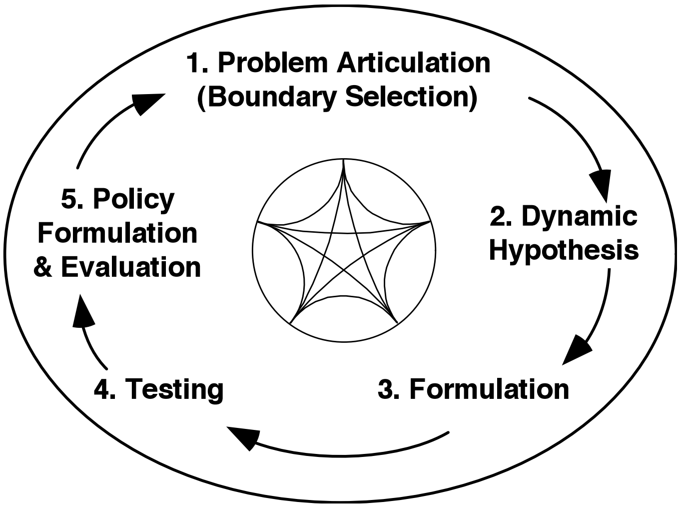

Figure x: The Modelling Process (Sterman (2000) – Chapter 3, p.87)

A further benefit of this approach is that it is entirely equation based and, as a result, transparent. In communicating with stakeholders, each state equation, stock formulation, and feedback structure can be examined and explained, whereas alternative agent-based approaches that rely on the code paradigm generate models that are "black-box", and cannot be easily shared and validated with clients. The mathematical modeling emphasis of the tool also allows any proficient SD modeler to construct disaggregate models, bring their expertise of feedback awareness and stock and flow formulation to disaggregate and network-inspired models.

This system can also be used to support education programmes for complex systems modelling (K-12, undergraduate, postgraduate), particularly where students that have taken a course in system dynamics, and so have a good grounding in mathematical modeling, can begin to explore network effects and the impact of network structures and interactions on overall system behaviour. Furthermore, because the framework is intuitive with its *Aggregate-Individual-Network* classification, teaching individual based modeling using this approach should be easier than having to specify models using computer code.

Future features are under development which will enhance the capability in iSimPlus. These include: implementation of the Behavioural Method (Ford 1999), sensitivity analysis to support stochastic simulations, and particle swarm optimization to allow users explore a wide range of alternative parameter values in the policy space. A set of training materials is under development, and a number of illustrative models and examples are being prepared as part of its planned release.

Acknowledgement

The support of Science Foundation Ireland for is gratefully acknowledged.

References

Barabasi, A.L. and R. Albert. 1999. Emergence of scaling in random networks. *Science.* 286(5439) 509-512.

Borshchev A. and A. Filippov. 2004. "From System Dynamics and Discrete Event to Practical Agent Based Modeling: Reasons, Techniques, Tools." *Proceedings of the 22nd International Conference of the System Dynamics Society.* Oxford, England. 2004.

Epstein, J.W. 2006. *Generative Social Science: Studies in Agent-Based Computational Modeling*. Princeton University Press.

Ford, D.N. 1999. A behavioral approach to feedback loop dominance analysis. System Dynamics Review, 15 (1) , Pages: 3-36.

Forrester, J.W. 1961. *Industrial Dynamics.* Pegasus Communications, Waltham MA.

Gilbert, N. 2008. *Agent-Based Models*. Sage Publications: Quantitative Applications in the Social Sciences.

Heath, B., R. Hill and F. Ciarallo. 2009. "A Survey of Agent-Based Modeling Practices (January 1998 to July 2008)" *Journal of Artificial Societies and Social Simulation*. Volume 12

Kermack, W. and A. McKendrick. 1927. Contributions to the mathematical theory of epidemics. *Proceedings of the Royal Society,* 115A 700-721. Reprinted, Bulletin of *Mathematical Biology* (1991) 53 (1-2), 33-55.

North, M.J., Collier, N.T., and Vos, J.R. 2006. "Experiences Creating Three Implementations of the Repast Agent Modeling Toolkit." ACM Transactions on Modeling and Computer Simulation. Vol.16, No.1, pp 1-25.

Osgood, N. 2009. "Lightening the Performance Burden of Individual-Based Models through Dimensional Analysis and Scale Modeling". System Dynamics Review, 25(2). 24pp.

Parunak, H. Van Dyke, R. Savit and R.L. Riolo. 1998. Agent-Based Modeling vs. Equation-Based Modeling: A Case Study and Users' Guide. Proceedings of Multi-Agent Systems and Agnet-Based Simulation (MABS'98), 10-25, Springer, LNAI 1534.

Rahmandad, H. and J.D. Sterman. 2008. Heterogeneity and Network Structure in the Dynamics of Diffusion: Comparing Agent-Based and Differential Equation Models. *Management Science*, Vol. 54, No. 5, pp 998-1014.

Roberts, E. 1978. *Managerial Applications of System Dynamics.* Waltham MA, Pegasus Communications.

Schelling, T.C. 1971. Dynamic Models of Segregation. *Journal of Mathematical Sociology*. Vol. 1. Pp 143-186.

Schild H. 2004. Enforcement in a free-flow system: a case study. In *Proceedings of the 22nd International Conference of the System Dynamics Society*, Oxford, UK. System Dynamics Society.

Sterman, J.D. 2000. *Business Dynamics: Systems Thinking and Modeling for a Complex World*. Irwin McGraw Hill.

Watts, D. and S. Strogatz. 1998. Collective dynamics of small-world networks. *Nature* 393(6684) 440-442.

Appendix 1 – The iSimPlus Model

```
sector setup
{
        DT    = 1.0;
        START = 0;
        FINISH = 100;
        POPULATION = 500;

}

sector global.references
{
   gref Type.Agent     = [Child](S)(I)(R);
   gref Type.Agent     = [Adult](S)(I)(R);
   gref Type.Aggregate = [SIRTest](S)(I)(R);

}

sector agent
{
        networks:
                participates.on (1);

        properties:
                agent.type    = Child;
                agent.trace   = true;


        markov.states:
                mgroup g1 = [(S,P,0.90)(I,D,d1,0.10)(R,P,0.0)](S,I,p1)(S,S,p2)(I,I,p3)(I,R,p4);

        equations:

         aux propInfected = NetGetNeighbourhoodProportion(1, I);
         param infectivity =  RandomRangeDouble(0.50, 0.75);
         param exp = RandomExponential(15.0);
          aux p2 = 1 - p1;
         aux p1 = propInfected * infectivity;
        aux p3 = 0.90;
        aux p4 = 1.0 - p3;
       aux d1 = exp;
}

sector agent
{
        networks:
                participates.on(1);

        properties:
                agent.trace   = true;
                agent.type    = Adult;

        markov.states:
                mgroup g1 = [(S,P,1.0)(I,D,d1,0.0)(R,P,0.0)](S,I,p1)(S,S,p2)(I,I,p3)(I,R,p4);

        equations:
```

```
            aux propInfected = NetGetNeighbourhoodProportion(1, I);
            param infectivity =  RandomRangeDouble(0.50, 0.75);
            param exp = RandomExponential(15.0);
             aux p2 = 1 - p1;
            aux p1 = propInfected * infectivity;
            aux p3 = 0.90;
           aux p4 = 1.0 - p3;
          aux d1 = exp;
}

sector aggregate
{
        properties:
                aggregate.name        = TopLevel;
                aggregate.level       = 0;

        equations:

                aux STot = SUM(S);
                aux ITot  = SUM(I);
                aux RTot  = SUM(R);
                aux ChildITot = SumByAgent(Child, I);
                aux AdultITot = SumByAgent(Adult, I);
}


sector network
{
        network.id = 1;
        network.name = Network1;
        network.trace = false;
        network.label = "Social Network";
        network.type = SMALLWORLD;
        network.degree = 5;
        smallworld.rewireprob = 0.05;
}

sector network
{
        network.id = 2;
        network.name = Network2;
        network.trace = false;
        network.label = "Blog Network";
        network.type = SCALEFREE;
        network.degree = 10;
scalefree.alpha = -1.7;
        scalefree.node.att.param = 3;
}
```

```
sector aggregate
{
            properties:
                        aggregate.name    = SIRTest;
                        aggregate.level   = 0;

            equations:
                stock s = s + (-IR) *DT();
                init    s = 9999;
               stock i = i + (IR-RR)*DT();
               init i = 1;
              stock r = r + (RR)*DT();
              init r = 0;

              flow IR = 6 * s * (i/10000) * 0.25;
              flow RR = i/2;
                 priority:  [IR];


}


sector graph
{
  graph.name = TestGraph;
  graph.label  = "Overall Values";
  graph.type   = TIMESERIES;
  graph.target.type = AggregateGraph;
  graph.target.id    = TopLevel;
  graph.y.vars   = [ChildITot , AdultITot ];
  graph.x.var    = TIME;
}

sector graph
{
  graph.name = TestGraph12;
  graph.label  = "Exploring Infections";
  graph.type   = TIMESERIES;
  graph.target.type = AggregateGraph;
  graph.target.id    = SIRTest;
  graph.y.vars   = [S,I,R];
  graph.x.var    = TIME;
}


sector statedisplaymanager
{
  state.manager.id = ColourMgr1;
  ColourMap = [Child](S,Colour.Green)(I,Colour.Red)(R,Colour.Blue);
  ColourMap = [Adult](S,Colour.Green)(I,Colour.Red)(R,Colour.Blue);
}




sector main
```

```
{
        action          = SIMULATE;
         flightsim.mode = (false, 50);


        disaggregate.mode = true;


        instantiate:
                create(Adult, 0.5);
                create(Child, 0.5);
        networks:
      create(1);
      //create(2);
}
```