Data integration in agent based modelling

Terry Bossomaier[1], Denise Jarratt[2], M. M. Anver[2], Tim Scott[1] and James Thompson[1]

[1]*Research Group for Complex Systems, Charles Sturt University, Bathurst, NSW 2795, Australia*

Email: tbossomaier@csu.edu.au

[2]*School of Marketing and Management, Charles Sturt University, Bathurst, NSW 2795, Australia*

Email: djarratt@csu.edu.au

Abstract

We describe a framework, SMUSA, for developing scenarios in a networked business environment using agent based modelling. The framework is usable by anybody with domain expertise, but requiring no programming knowledge. By developing the specification details in XML, the framework integrates smoothly with data mining tools for parameterising customer preferences and identifying opportunities and threats. A key requirement of successful agent based modelling is the incorporation of real data. The interface to data warehouses and other resources, we achieve through the universal interchange format, XML. We describe a simple example of financial services in rural areas using RePast as the ABM software engine.

1. Introduction

Many complex systems, in nature and society, arise from the interaction of many autonomous entities, or agents. It is in general not possible to derive the system behaviour from an understanding of the behaviour of individual agents, no matter how detailed. Thus computer simulation of the emergent system behaviour is the only course of action. There are many successful simulations of complex systems in the physical and biological sciences, from climate modelling to fluid dynamics, from genetic regulatory networks to ecosystems (Kauffman, 1993; Gregorio and Serra, 1999; Bossomaier and Green, 2000). On the other hand quantitative modelling of human social systems is still in its infancy. Part of the challenge lies in the immense difficulties of modelling human behaviour. But as we build larger and larger data warehouses of individual preferences and actions, our ability to model collections of people will continue to improve. The other part of the challenge lies in the domain expertise of sociologists and marketing managing theorists. To be able to play with new ideas in modelling they need tools which do not require programming expertise.

One way to approach emergent properties is to look for the simplest abstraction of the individual agents and how they interact. This approach has two advantages: it encapsulates the essential properties of system dynamics without extraneous detail and illuminates similarities of quite disparate applications. But it also saves on computing power in a big way. What it does not do, is enable us to refine agent behaviour based on the large information resources we are now starting to collect.

The advent of greater computing power and associated graphics has led to a rise in agent-based modelling, and the potential for integrating data warehouses and data mining tools. But there are still substantial problems. Data integration involves many issues of uniform standards and communication protocols, which is the first issue we address in this paper. The second issue is ease of use. Many business, health, environmental and other professionals would like to build and play with models, much as they use spreadsheets today. But the available software tools usually require programming expertise. We illustrate the approach with a simple, hypothetical example of bank closures in rural areas.

First we describe the ideas of scenario planning for organisations and the need for agent modelling tools usable by managers and executives. We then discuss a specific scenario of the threats to bank of new forms of retail banking and how this may be embedded in a simple agent based model.

The core of this paper is an XML framework which serves as the integration glue. XML descriptions of agents, their behaviour and interactions subserve two functions. They serve as templates for the automated generation of java code for RePast. But they also serve as templates for data mining agents which can parametrise agent behaviour against real data.

In the final version of the system, XML files will be generated by window interfaces, allowing interactive, graphical specification of agents and their behaviours and interactions.

2. Scenario planning in network organisations

2.1 Scenario planning: a process of strategic foresight

Companies approach analysis of the environment with various levels of rigour and comprehensiveness. Research indicates that macro-environment data is collected and synthesised by approximately fifty percent of organisations, with greater emphasis being placed on the synthesis of industry and market information (Hubbard, 2000). Strategists generally focus their assessment around known market forces, and existing value, created through current business partnerships and client relationships. However, new organisational partnerships, technology, and shifting demand may eliminate current bases of competition, and introduce new platforms that support new forms of value creation (Jarratt and Fayed, 2001; Morgan and Hunt, 2002).Thus, the question arises: ‘how can managers build new understandings of the emerging forces shaping their environment?’

In addition to the complexities of market, network and technology dynamism, an organisation’s ability to understand the future as it emerges is hindered by psychological, organisational and political barriers (Watkins and Bazeran, 2003). Cognitive biases, organisational silos and systematic flaws in decision-making processes lead to underestimating forces, dispersed information and overvalued interests of some groups.

Scenario planning was introduced as a technique to address some of the psychological, organisational and political barriers constraining a firm’s strategic foresight capability, and is commonly employed to identify forces shaping potential future business and government environments. As a pivotal component of organisational strategy development, scenario planning is the process of identifying internally consistent, plausible alternative futures (Krentz and Gish, 2000).

Typically, scenario planning purports to draw together tacit knowledge of employees, the knowledge that resides in an organisation’s knowledge management system, and opinions of external stakeholders and experts in fields thought to impact on those alternative futures. In addition, scenario planning has been applied in a vast array of contexts beyond its traditional underpinning of organisational strategy, for example, projecting emerging themes in the marketing services literature (Lewis et al., 2002).However, four important issues have emerged from its use that relate to both the process of scenario planning and how managers construct alternative futures.

First, scenario planning adopts an organisational rather than a network perspective. Today, many independent organisations are linked to other independent organisations either formally, through joint ventures and licensing agreements, or less formally, but equally in terms of strategic importance, through strategic alliances, value-adding partnerships and outsourcing agreements (Birkinshaw et al., 2000). Complexity theorists have been investigating the evolution of networks (Jeong et al., 2003; Robertson, 2003; Barabási, 2002; Wilkinson and Young, 2003), providing theory underpinning network expansion/contraction through preferential attachment and critical nodes. Thus, the process of scenario planning needs to be conducted within the context of the evolution of an organisation within a network of organisations, some core, some peripheral and should draw upon the explicit and tacit knowledge residing in that network.

Second, even though scenario planning encourages participants to question assumptions held by themselves and others (Lewis et al., 2002), cognitive biases continue to shape managers' determination of which forces will be dominant in the future. The challenge for organisations in addressing this issue is to access tacit (know-how and know-why embedded in an organisation's skills and routines, Mohr and Sengupta 2002) and explicit knowledge residing in managers within the network, while minimising the subjectiveness that surrounds the synthesis of that knowledge.

Third, there is little understanding of the interaction between forces purported to shape future environments, and fourth, scenario planning is undertaken as a single event, limiting a firm's ability to understand the changing trajectory and interaction of the forces shaping the future. Taken together these issues make the case for quantitative scenario planning informed by real, objective data. Agent-based modelling, as discussed herein, is thus the obvious choice.

2.2 The context of this simulation

Banks seek to achieve the joint objectives of maximising profitability, shareholder value added and survival in the marketplace (Heffernan, 1996). Those responsible for achieving those objectives locally, regionally or nationally will require strategic foresight about the market and behaviours of clients to select the most appropriate strategy that will provide a mechanism for achieving those objectives. Further, once selected the strategy takes on dynamic properties as adjustments are made according to mental models of those implementing the strategy, knowledge emerging about new forces shaping the environment, actions of competitors, changes in the behaviour of value-generating agents (adaptive model) and/or an imposed re-conceptualisation of the firm (interpretive model) (Robertson, 2003; Chaffee, 1985).

The dynamic nature of the environment is reflected through the changing nature and number of competitive configurations, each with the ability to access an expanding set of specific competencies and capabilities that an organisation must consider when shaping strategy (Chakravarthy, 1997). Traditional entry and mobility barriers have been lowered through technological advances and deregulation policies, and competitive advantages are rapidly created and eroded (Chakravarthy, 1997). Within this environment of innovation intensity, competition intensity, and information technology change, 'windows of opportunity will be narrower and more transitory'(Achrol, 1991).

In banking as well as other industries concern is raised about current managers' preoccupation with restructuring and re-engineering as opposed to energetically focussing their attention on the outward and forward. Any company that is a bystander on the road to the future will watch its structure, values, and skills become progressively less attuned to industry realities (Hamel and Prahalad, 1994). The current inward focus towards divest and downsize is certainly making an impact on asset productivity; however, are proponents of this de-layering attuned to the opportunities of the future and has this focus positioned the organisation to take advantage of the opportunities that might arise? Turbulence can result from changes to the links with value-generating agents (introduction of new technology or new policies for interaction), by the rapid introduction of new agents (supermarkets, banking co-operatives) or removal of agents (one of the major bank's exit from the physical local space) to the strategic space (Robertson, 2003). In assessing all these diverse forces, two needs become apparent. The range and subtlety is so great, that managers and executives need to be able to build models directly themselves, incorporating their extensive experience as they go

Customers will have a requirement for face-to-face interaction with the bank, and its importance will be traded off against other attributes of banks' offers (e.g. cost, expertise, convenience) as consumers make choices on where to bank. The nature of the trade off will depend on the choice model evoked by the customer. The complexity and multiple value exchanges associated with banking services (accounts, loans, insurance, superannuation) create an exit barrier for the customer, and encourage sufficing behaviour. However, when attributes that account for high levels of utility in exchanges are eroded beyond an individual customer's minimum level of tolerance, switching behaviour is likely to occur.

2.3 Research approach

Organisations, the networks in which they are embedded, and the forces which act upon them, form complex systems. Computer simulation is frequently the means employed to identify emergent properties, and, thus, promises to become a key tool in scenario planning. Quantitative simulation has traditionally involved formulation of a macro model and its subsequent solution. Agent-based modelling recognizes that for many complex systems this is either impractical or impossible. Instead it defines agents, their interactions, and steps through their behaviour in time.

Numerous packages for agent based modelling (ABM) exist (O'Sullivan and Haklay, 2000), such as Ascape (ASCAPE, 2004) and Ethos (ETHOS, 2004), however, they are either too simplistic for large-scale models or require low level programming. We propose to overcome these limitations by developing declarative techniques for describing agents at all levels using XML, graphical tools for describing agents' interaction, and algorithms for agents to learn automatically from real data. This will create a deeper understanding of ABM at different levels of complexity embedded in a changing environment. We will address issues such as sampling, and the number and diversity of each agent's category. We will develop algorithms to match agent descriptions to cognitive models and parameterise them by mining data warehouses. The entire framework, shown in Figure 1, we call, SMUSA (Scenario Modelling Using Software Agents).

3. Simple scenario model for bank use

As an illustration of the SMUSA philosophy, we describe a simple cellular automata (CA) simulation to demonstrate criticality effects in bank closures within small communities. In this simulation the rules of consumer change are fixed, Boolean predicates, but the more general framework uses fuzzy logic, see Section 5.3, as more appropriate to human decision making. Consumer's financial services depend upon

1. whether there is a bank, B, sufficiently "close". This is their first preference, derived largely from tradition.

2. a sufficient number of close neighbours who use the new alternatives, which might be credit unions or financial services offered within a supermarket. For example, Tesco, the UK's largest supermarket chain, has recently introduced in-store banking. So we model the alternative as a supermarket bank S, which is always close at hand, offers slight cost savings, but is new and perceived to be slightly "risky" or suspect.

The initial configuration has most customers using the bank, with some supermarket users in local clusters. There is a supermarket for every local neighbourhood (the grid and diagonal nearest neighbours (9 including the centre)).

The customers have several states, reflecting the distance of the nearest bank. T_0,T_1,T_2..., where T_n indicate a bank n steps removed from the local neighbourhood. Customers have a decision thresholds, θ_0,θ_1,θ_2... to switch to the supermarket thresholds, reflecting the willingness customers to travel balanced against perceived risk.

The rules take the form:

- if there is a bank in the local neighbourhood, then stay with it, unless the number of local supermarket users, N_s, is greater than some threshold, η_0. – no change;
- if there is no local bank, but there is one within the extended neighbourhood, T_k, then if N_s is greater than some threshold, θ_k, switch to the supermarket;
- if the switch to S does not occur, then set customer state to $T_{k+1}$where k is the minimum of the neighbourhood.

Now we can look for some starting configurations and bank densities, which create a flash point, where most or all customers switch over, or perhaps where the customers fluctuate wildly in their preferences. Formally this is analogous to the sort of CA models which have been used for simulating bushfires and epidemics.

Although the positive feedback effects in a model of this kind are not hard to see, in practice they are easily overlooked. The Economist in a special report in April this year described how a new wave of banks is taking over the United States –banks which relish in their local branches, offer extended opening hours more suitable to retail customers, provide toys for kids and in general dispel the formal/hostile nature of many traditional banks. In Australia we see small operations such as the Bendigo Bank, spreading outside the small Victorian town of its origin across south east Australia.

4. The XML framework

An agent based model requires the specification of three sets of information:

1. the agents themselves
2. the connectivity matrix
3. the simulation parameters (initial conditions etc.)

Packages such as RePast (REPAST, 2004) or SWARM (SWARM, 2004) will create and run the simulation given these code plug-ins. We want to specify these parameter sets in a code independent way. In principle a simulation should be runable under a number of simulation environments. At the same time, we need the framework to be sufficiently close to a particular application domain that it is practical to use, i.e. it does not require endless details defining the nature of the simulation (climate modelling, enterprise planning, epidemiology etc.) In this sense it is effectively a subclassing of general specifications such as FIPA (FIPA, 2004).

To achieve this level of abstraction we want the agent specifications to be in a universal language, which is an open standard and vendor independent. XML satisfies these requirements and has grown rapidly in the last few years to encompass many domains of activity and expertise. Apart from its value as a universal glue, it has another advantage – it is self documenting. XML files representing agents or simulations can be mapped in a straightforward way using XSL to printed or online documents. Figure 1 shows the mechanism of integration with modelling packages such as RePast.

XML has become the lingua franca of web services and widescale integration of data and software. We choose XML for the specification of agents, rather than a software tool such as the Unified Modelling Language (UML) for this reason. UML, although a very powerful tool, has the disadvantage, too, in that it requires some software engineering experience to use it effectively. The client users of SMUSA are unlikely to have this experience.

The SMUSA framework (Scenario Modelling Using Software Agents)embodies these ideas as Figure 2 shows. SMUSA has three levels of increasing complexity:

1. Level 1 is a cellular automata simulation. The advantages of cellular automata are that most of the analysis and structuring of the simulation is done in the modeller's domain of expertise. So, for example in the modelling of lava flows (Gregorio and Serra, 1999), the work goes into mapping the equations of lava flow, cooling and solidification, to a set of cellular update rules. Nevertheless, powerful results are frequently obtainable. Each kind of agent, e.g. a customer, is represented by its own schema, embodying their attributes, with appropriate type and value constraints, and their behavioural model. Level 1 is a prerequisite for level 2.

2. Level 2 moves from a cellular automata model to more powerful agents, whose preferences and activities are expressed by fuzzy logic or neural networks. These agents can be parameterised using data mining. The XML specification is crucial to being able to interface with data warehouses automatically.

This paper discusses our implementation of Level 1.

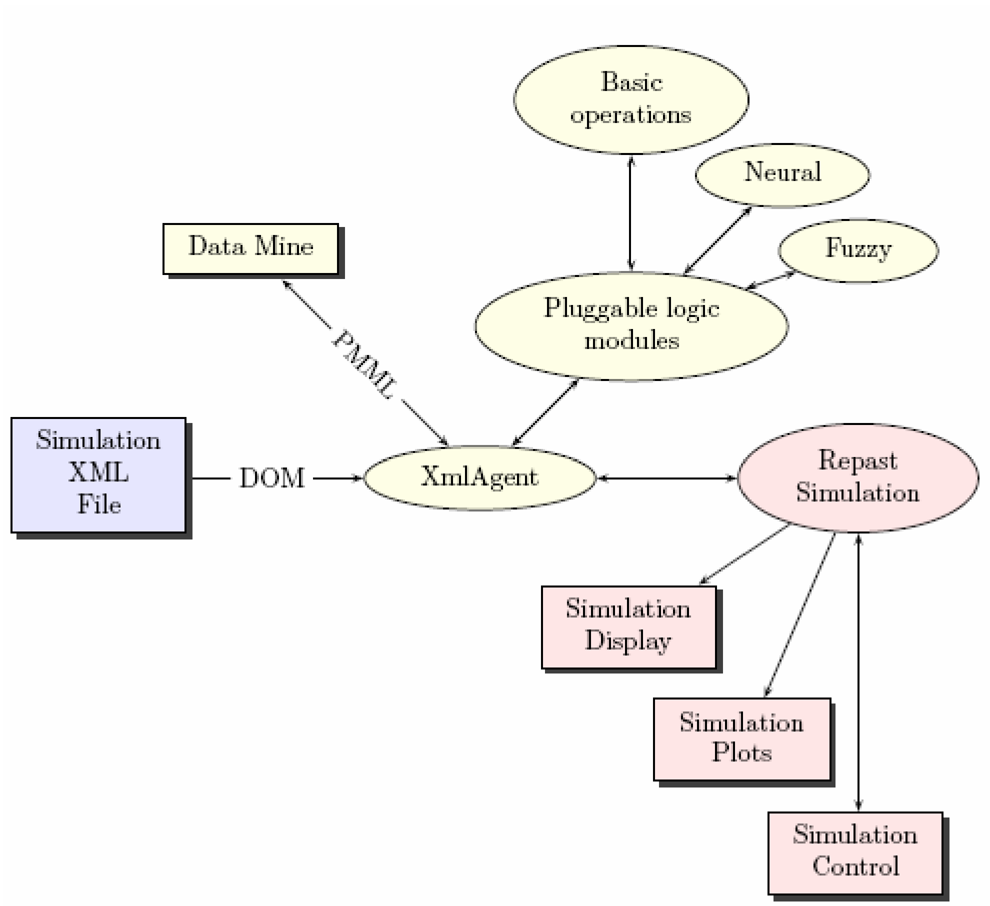

Figure 1. Mechanism of incorporation of XML into RePast: Shaded boxes show the components under executive control. Other components will be brought in automatically.

5. Implementation of Level 1

5.1 RePast

Initial design for the SMUSA system was conducted in RePast. RePast is a flexible agent programming toolkit which takes care of onerous tasks such as the graphical display of results and scheduling. RePast is a mature agent based programming toolkit with a high degree of flexibility in how a simulation is constructed. It offers features such as:

•Synchronous or asynchronous scheduling

•Grid or network connection

•Loosely bound agents

•Platform independance through Java

The loose and flexible integration of agents in a RePast framework allows the use of a selection of techniques for utilising XML in the description of agents. Illustrative portable frameworks for working with XML files are the W3C's DOM (DOM, 2004) as represented in JDOM (JDOM, 2004) and David Megginsons defacto standard SAX (SAX, 2004).

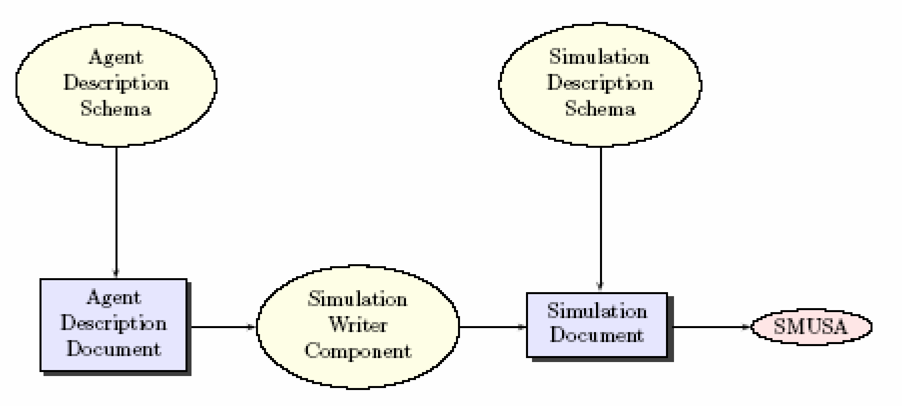

Figure 2. The SMUSA XML file framework: Schema are created to describe both agents and the structure of the simulation. Each simulation is then captured (as shown in the grey boxes) by a document describing the particular agents and the simulation parameters.

5.2 Integration of XML

DOM will build a structure in memory representing the XML file, the programmer can then navigate this structure. DOM offers the benefit of random access to an XML file and the ability to pass entire subsections of the document tree as an argument to a system such as that responsible for mining data.

The initial method of integrating XML into RePast was to create an Xml Agent class. This Xml Agent is responsible for building its internal data structures based on the XML element it is presented. XmlAgent keeps track of what type of agent it is and its id, if any. XML agent nodes contain two main types of information, attributes and rules. Each attribute contains a name a data type and a value. Each attribute in the XML file is enumerated by the XmlAgent object and the values are entered in a hashtable indexed on each attributes name. Values are entered into the table as an object of the type specified (int, float, string, etc).This approach is easy to implement but it lacks the declarative control possible with a schema for each agent class and also creates additional run-time overheads.

Implementing rules in XML presents the greatest challenge. Rules in the system need to be flexible enough to provide a variety of behaviours while retaining a simple structure. Rules are broken into several sub-sections each represented by a class responsible for its area of the rule logic. A class exists for each rule implemented in the system. (IfRule for conditional structures, GTRule to determine the highest value argument). Rules contain argument classes which wrap argument values and provide functionality such as counting neighbours or finding the closest agent matching a set criteria. As the XML structure is navigated these classes are instantiated and added to their parent agents. Each rule class encapsulates the entire logic for their operation and can simply be executed at any specified interval.

```
void tick(int tickno)
{
    for all rules
        if rule[i].interval % i == 0
            rule[i].execute()
}
```

Finely encapsulating rules in classes enables the use of modular rulesets a basic logic ruleset is included in SMUSA by default, but as requirements grow, extra argument subclasses can be written to implement complex functions such as fuzzy or neural operations.

5.3 *Specification of fuzzy logic rules*

Fuzzy logic is a well established AI procedure for specifying human judgements. The rules for agent behaviour may also be easily represented in XML as shown below, for the banking example.

```
<fuzzy>
   <output calculation>
        <inference engine>
             <model>MAMDANI</model>
             <type>PRODUCT</type>
        </inference engine>
        <defuzzification>CENTROID</defuzzification>
   </output calculation>
   <rules>
        <rule>
            <antecedent>
                 <if> <input_name = "bank_distance"> SMALL </if>
                                  </and>
                 <if> <input_name = "supermarket_customers">LARGE</if>
             </antecedent>
             <consequent>
                 <then> <output name = "switch_to_or_remain_with">SUPERMARKET</then>
             </consequent>
        </rule>
            .
            .
            .
        <rule>
        </rule>
   </rules>
</fuzzy>
```

5.4 *Simulation description*

As well as the ability to represent a simulation in an XML file, a mechanism for creating such a file is required. The simulation may have thousands of each type of agent and manually writing an XML file of such length is impractical. In the simulation we can specify the network connection, simulation name, agent distribution and other simulation parameters. The schema and files for this description are quite similar to those describing the instance of a simulation with the addition of simulation and agent 'meta-information'. Values for agent attributes are designated in one of several ways:

- •A bound random number eg. "Any number between 20 000 and 200 000";
- •A specific value;
- •As a member of a set or a standard distribution;
- •By mining for the information;

Several efforts to describe the parameters of a data mining query in XML such as PMML (PMML, 2004) already exist, the modularity of XML Schema allows implantation of an XML data mining query into an attribute value node. The system will act as a broker between the ABM and Data Mining software.

6. Sample XML listings

Below we give the XML schemas for the ‘Bank’ and ‘Customer’ agents as well as the schema for the ‘Bank Closure Simulation’ as a whole.

6.1 XML schema for a bank agent

As the bank does not encompass any logic in the simple simulation that has been chosen for this paper it is a very simple agent, listing the bank agent merely reserves a bank grid cell.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="bank">

    <xs:sequence>
        <xs:element name="identification"/>
        <xs:element name="x_value"/>
        <xs:element name="y_value"/>
        <xs:element name="agent_interaction">
            <xs:sequence>
                <xs:element name="agent_type"/>
                <xs:element name="rule"/>
            </xs:sequence>
        </xs:element>
    </xs:sequence>
    </xs:element>
</xs:schema>
```

6.2 XML schema for a customer agent

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="customer">
    <xs:sequence>
        <xs:element name="identification"/>
        <xs:element name="x_value"/>
        <xs:element name="y_value"/>
        <xs:element name="customer_type"/>
        <xs:element name="bank_distance"/>
        <xs:element name="agent_interaction">
            <xs:sequence>
                <xs:element name="agent_type"/>
                <xs:element name="rule"/>
            </xs:sequence>
        </xs:element>
    </xs:sequence>
    </xs:element>
</xs:schema>
```

6.3 XML schema giving the meta description of the simulations

An intermediary program will take an XML file similar to that following and create a simulation XML file with the appropriate number of agents.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="simulation">
    <xs:sequence>
        <xs:element name="name"/>
        <xs:element name="seed"/>
        <xs:element name="steps">
            <xs:sequence>
                <xs:element name="number"/>
                <xs:element name="name"/>
            </xs:sequence>
        </xs:element>
        <xs:element name="display_type">
            <xs:sequence>
                <xs:element name="x_size"/>
                <xs:element name="y_size"/>
            </xs:sequence>
        </xs:element>
        <xs:element name="agents">
            <xs:sequence>
                <xs:element name="agent_type"/>
                <xs:element name="max_number"/>
                <xs:element name="min_number"/>
            </xs:sequence>
        </xs:element>
    </xs:sequence>
    </xs:element>
</xs:schema>
```

6.4 Bank simulation

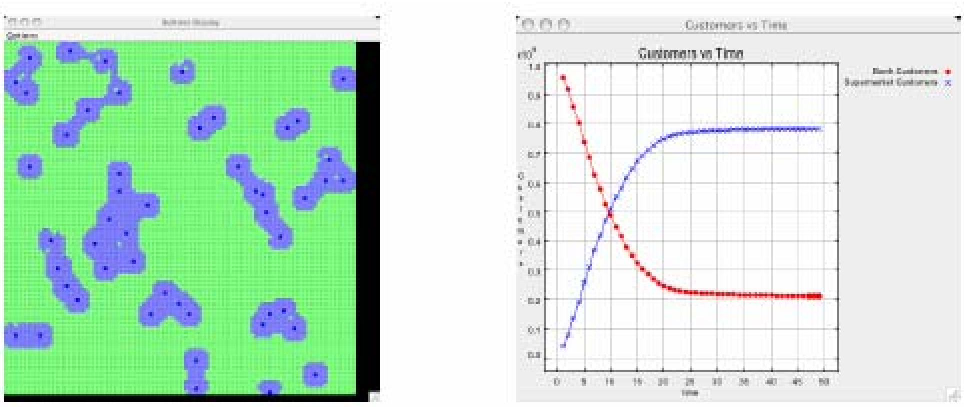

Figure 3. RePast running the bank simulation (In the screen shot on the left, the grey areas controlled by the banks, shown as black dots, decrease as shown on the right, where crosses shown supermarkets and circles the banks.)

7. Discussion

Whereas in engineering and manufacturing, computer simulation has been a core research and development tool for many years, in social systems it is just beginning. In this paper we have shown how agents can be described in a architecture and software independent way using XML Schemas. The XML descriptions then subserve the automatic creation of agents for modelling software and provide hooks to search agents which obtain parameters from diverse data warehouses.

The paper describes a methodology for bringing agent based modelling to within reach of organisational strategists, managers and executives with no programming knowledge. By representing each component by XML files, intermediate software modules can build simulations and data mining parameterisation automatically, which are, at the same time, self-documenting. Future developments will provide GUI interfaces for building agents, behavioural rules and simulations.

References

Achrol, R. S. (1991). Evolution of the marketing organisation: new forms for turbulent environments. *Journal of Marketing*, 55:77–93.

ASCAPE (2004). Ascape. http://www.brook.edu/es/dynamics/models/ascape/default.htm.

Barabási, A.-L. (2002). Linked. *Perseus*, Massachusetts.

Birkinshaw, J., Bresman, H., and Hakanson, L. (2000). Managing the post-acquisition integration process:

How the human integration and task integration processes interact to foster value creation. *Journal of Management Studies*, 37:395–425.

Bossomaier, T. R. J. and Green, D. G., editors (2000). *Complex Systems*. Cambridge University Press.

Chaffee, E. E. (1985). Three models of strategy. *Academy of Management Review*, 10(1):89–98.

Chakravarthy, B. (1997). A new strategy framework for coping with turbulence. *Sloan Management Review*, 38(2):69–82.

DOM (2004). Document object model. http://www.w3.org/DOM/.

ETHOS (2004). Ethos. http://centria.di.fct.unl.pt/jsimao/ethos/.

FIPA (2004). Foundation for interacting physical agents. http://www.fipa.org/.

Gregorio, S. D. and Serra, R. (1999). An empirical method for modelling and simulating some complex macroscopic phenomena by cellular automata. *Future Generation Computer Science*, 16:259–271.

Hamel, G. and Prahalad, C. K. (1994). Competing for the future. *Harvard Business Review*, pages 122–128.

Heffernan, S. (1996). Modern Banking in Theory and Practice. Wiley, Chichester, UK.

Hubbard, G. (2000). Strategic Management: thinking, Analysis and Action. Pearson Education Pty Ltd, Frenchs Forest, Australia.

Jarratt, D. G. and Fayed, F. (2001). The impact of market and organizational challenges on marketing strategy decision-making. *Journal of Business Research*, 51(1):61–72.

JDOM (2004). Java document object model. http://www.jdom.org/.

Jeong, H., Neda, Z., and Barbarasi, A. (2003). Measuring preferential attachment for evolving networks. http://www.nd.edu/ alb/public.html accessed 30/06/04.

Kauffman, S. (1993). The Origins of Order. Oxford University Press.

Krentz, S. and Gish, R. (2000). Using scenario analysis to determine managed care strategy. *Healthcare Financial Mgt*, pages 41–43.

Lewis, B., Foxall, G., and Hogg, G. (2002). Predicting a diverse future. *European Journal of Marketing*, 36(4):479–494.

Morgan, R. E. and Hunt, S. D. (2002). Determining marketing strategy: a cybernetics systems approach to scenario planning. *European Journal of Marketing*, 36(4):450–478.

O’Sullivan, D. and Haklay, M. (2000). Agent-based models and individualism: is the world agent-based? *Environment and Planning A*, 32(8):1409–1425.

PMML (2004). Predictive markup and modelling language. http://www.dmg.org/pmml-v2-0.html.

REPAST (2004). Recursive porous agent simulation toolkit. http://repast.sourceforge.net/.

Robertson, D. A. (2003). Agent-based models of a banking network as an example of a turbulent environment: the deliberate vs emergent strategy debate revisited. *Emergence*, 5(2):56–71.

SAX (2004). Simple api for xml. http://www.saxproject.org/.

SWARM (2004). Swarm. http://wiki.swarm.org/.

Watkins, M. and Bazeran, M. (2003). Predictable surprises: the disasters you should have seen coming. *HBR*, pages 72–80.

Wilkinson, I. and Young, L. (2003). A view form the edge. Marketing Theory, 3(1):179–195.