# Discovery Assistance for Evidence Based Model Development

Catriona Kennedy[1], Georgios Theodoropoulos[1], Volker Sorge[1], Ed Ferrari[2], Peter Lee[3] and Chris Skelcher[4]

[1]School of Computer Science, University of Birmingham, UK

[2]Informatics Collaboratory for the Social Sciences, University of Sheffield, UK

[3]Centre for Urban and Regional Studies, University of Birmingham, UK

[4]Institute for Local Government Studies, University of Birmingham,UK .

Email address of corresponding author: cmk@cs.bham.ac.uk

**Abstract.** Agent-based simulation can help to predict policy impact and generate explanations of observed phenomena. However, the models underlying the simulations have to be based on evidence on the real world. The paradigm of Dynamic Data Driven Application Simulation (DDDAS) can improve the accuracy of a simulation's representation of reality by continually adjusting the simulation in response to incoming data. Conversely the simulation predictions can help to constrain and direct the data collection and analysis process. As part of the AIMSS project (Adaptive Intelligent Model-building for the Social Sciences) we investigated the feasibility of DDDAS for the social sciences using artificial intelligence (AI) techniques to interpret simulation predictions and to check for their consistency with real world data content. This paper describes the resulting software architecture and discusses its further development into an interactive “workbench” to assist with the discovery of evidence and the iterative development of models.

## Introduction

There is a substantial desire by policy makers to understand “what works” in order to increase the accuracy and impact of their interventions on society (Davies et al. 2000). Agent-based simulations can act as test environments to analyse the effects of such interventions (Happe et al., 2006; Entriken and Wan, 2005; Widergren et al., 2006).

Understanding “what works” involves not only predicting effects but also explaining *why* an intervention should work based on a deep understanding of social processes. Causal mechanisms are poorly understood prior to interventions by governments, as in the attempts to tackle social exclusion and disadvantage (Levitas 2005). Simulations may help to provide such explanations due to their ability to produce a causal sequence of events which can “generate” a social phenomenon (Epstein, 2006). For both prediction and explanation, however, the models underlying the simulations need to be accurate and based on evidence in the real world. For example, the agent behaviour rules should reflect the behaviour of real agents (such as house buyers).

The recently developed paradigm of Dynamic Data Driven Application Simulation (DDDAS) (Darema, 2005) can improve the accuracy of a simulation's representation of reality. DDDAS is a method by which a simulation is continually adjusted in response to real-time data. In some cases, the data can be directly absorbed into the simulation. The simulation predictions may also be used to control the data collection and analysis, forming a feedback loop. Not only the simulation parameters but also the underlying model (e.g. the physics equations or agent rules) may be adjusted so that it fits the data.

In earlier work (Lee et. al, 2006) we proposed an architecture for “intelligent assistance” for model building and revision based on DDDAS principles. This architecture was developed during the AIMSS project (Adaptive Intelligent Model-building for the Social Sciences) which was a feasibility study to investigate whether the DDDAS paradigm can be applied to the social sciences and whether it makes sense if the simulation model is abstract and does not run in parallel with live data updates from a real system.

In this paper we describe the AIMSS architecture and show how it was implemented as a proof-of-concept. Finally we discuss its usefulness for policy researchers and other social scientists.

## AIMSS Architecture

### Automating the scientific discovery loop

The accuracy of a simulation model is typically tested by validating its predictions with respect to the available data. In other words, it should be possible for the simulation to “reproduce” the current data and act as an explanation for the observations in the real world system. Model development and revision can be described as an iterative process involving the following stages:

1. Formulate initial model and run simulation;

2. Once the simulation has stabilised, inspect it visually and determine whether it makes interesting predictions which need to be tested;

3. Collect the relevant data and analyse it;

4. Determine if the simulation predictions are supported by the data;

5. If the data does not support the predictions, determine whether the model should be revised. Experiment with variations of the original simulation and return to Step 2.

The AIMSS architecture provides automated assistance with this process. A schematic diagram of the AIMSS interpretation of DDDAS is shown in Figure 1. We can think of the simulation as running in parallel with events in the real world. Although in the social sciences, the data is likely to be historical, it could be reconstructed to behave like a stream of data being collected in parallel with the simulation. The simulation does not have to correspond to a particular observed system but can instead represent a class of systems with similar properties (e.g. the usual behaviour of shoppers in a crowd, not a specific crowd in real time). This means that the variable values read from the data cannot be directly absorbed into the simulation as might happen in a physical model. The data may be collected from multiple real world instances of the general class of systems represented by the simulation.

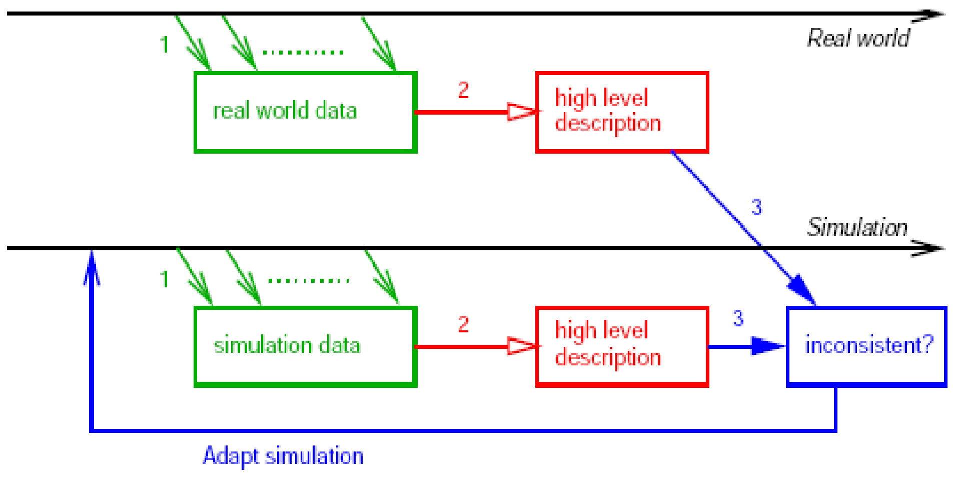

Figure 1: Data driven adaptation of a simulation

The figure can be divided into two processes: the first is a process of interpreting both the simulation predictions and the raw data content and determining whether they are consistent (arrows 1, 2 and 3 in the figure). The second involves adapting the simulation in the event of an inconsistency. This requires the following automated capabilities:

1. Interpretation of simulation states (at regular intervals or on demand): this is not just about reading the current variable values, but about generating a high level description summarising patterns or trends.

2. Interpretation of real world data: the same methods are required for data interpretation as for simulation interpretation, except that pre-processing of the raw data is necessary. This may involve the integration of data from multiple sources and the generation of higher level datasets that correspond to the simulation events.

3. Consistency checking to determine whether the simulation states mostly agree with descriptions of data content.

4. Re-direction and focusing of data collection in response to interpretation of simulation states or uncertainty in the data comparison.

Synthetic data is generated from the simulation as a trace of events. This may be a sequence of actions carried out by each agent and the circumstances of the action (e.g. moving house from source to destination). The real world data contains records of events that actually happened in the scenario being simulated.

Data mining can be applied to both datasets (simulation and real world) to produce two sets of high level descriptions for simulation and real world data respectively. These descriptions are general statements on the patterns and trends in each. We have investigated Association Rule Mining (Agrawal and Srikant, 1994) to produce generalised statements and logical satisfiability checking (Moskewicz et al., 2001) to look for inconsistencies.

## Role of Ontologies

The first process of Figure 1 is the generation of datasets from both the simulation and the real world. To define the structure of the data, an ontology is required to specify the entities and attributes in the simulation, and to define the state changes. There are actually two components required to define a simulation:

1. Static entities and the relations of the model. For example, households and homes exist and a household can move from one region to another; a household has a set of needs that must be satisfied;

2. Dynamic behaviour: the decision rules for the agent as well as probabilistic rules for dynamic changes in the environment and household status (ageing, having children, changes in income etc.) The way in which these entities are initialised should also be stated as part of the model, as this requires domain knowledge (e.g. initial densities of population and houses etc.) For more detailed models, this becomes increasingly non-trivial, see e.g. (Birkin et al. 2006).

In the AIMSS prototype, both these components are specified in XML. In future work we will consider the use of OWL. The XML specification also includes the agent rules. These specifications are machine-readable and may potentially be modified autonomously.

The entities and attributes are used to define the structure of data to be sampled from the simulation as well as the structure of a high level dataset to be derived from the pre-processing of the raw data. At the end of this process, we have two datasets, one records a sequence of simulated house moves, the other contains a sequence of actual moves.

## Data Mining: Recognising General Patterns

The second stage in Figure 1 is the generation of high level descriptions. These are general statements about the developments in the simulation and in the real world. For this purpose, we are investigating data mining tools. We have done some initial experimentation with Association Rule Mining using the *Apriori* algorithm (Agrawal and Srikant, 1994), which is available in the *Weka* Machine Learning package (Witten and Frank, 2005).

Association rules are a set of “if ... then” statements showing frequently occurring associations between combinations of “attribute = value” pairs. This algorithm is suited to large databases containing qualitative data which is often produced in social science research. Furthermore, it is “unsupervised” in the sense that predefined classifications are not given. This allows the discovery of unexpected relationships.

An association rule produced by *Apriori* has the following form:

*if* ($a_1$ *and* $a_2$ *and* ... $a_n$) $s_1$ *then* ($c_1$ *and* $c_2$ *and* .. $c_m$) $s_2$ *conf*(c)

where $a_1$, ..., $a_n$ are antecedents and $c_1$, ..., $c_m$ are consequents of the rule. Both antecedents and consequents have the form “attribute = value”. $s_1$ and $s_2$ are known as the support values and c is the confidence. The support value $s_1$ is the number of occurrences (records) in the dataset containing all the antecedents on the left side. $S_2$ is the number of occurrences of both the right and left sides together. Only those collections of items with a specified minimum support are considered as candidates for construction of association rules. The confidence is $s_2/s_1$. It is effectively the accuracy of the rule in predicting the consequences, given the antecedents. An example minimum confidence may be 0.9.

The higher the support and confidence of a rule, the more it represents a regular pattern in the dataset. If these measures are relatively low, then any inconsistency would be less “strong” than it would be for rules with high confidence and high support. The values of attributes are mutually exclusive. They are either strings or nominal labels for discrete intervals in the case of numeric data.

Effective use of data mining requires considerable preprocessing of the data in order to minimise errors and remove entries that contain no useful information (e.g. the selection “other” from a list of preferences in a survey). Similarly the resulting rules require post-processing on order to remove redundant and irrelevant associations.

### Consistency Checking: Testing Hypotheses

The third stage in Figure 1 is consistency checking. The two high level descriptions are sets of logical statements in the form of “if ... then ...” rules. Logical satisfiability checking is a method where a statement has all of its components checked to see if there is any combination of values (true or false) which make the whole statement true. In our case the statement (or *formula*) is a set of logical assertions connected by “and”. In logical terminology such a formula is *satisfiable* if there is a possible situation in which the statement is true. A simple example of an unsatisfiable formula is “(if p then not(q)) AND (if p then q)”.

Satisfiability checking is necessary when there is no advance knowledge of the form of the logical statements. In the AIMSS case, these “formulae” are conjunctions of logical statements that have been *discovered* to be true in both the simulation data and in the real world data using association rule mining. They are not known in advance. The input to the satisfiability checker is a conjunction of both sets of rules (from simulation and real world) resulting in a combined formula of the form: “(first conjunction of association rules) AND (second conjunction of association rules)”. If a rule is found in the first list that is inconsistent with one or more rules in the second list then the whole formula is unsatisfiable. We have done initial experiments with an algorithm called *zChaff* (Moskewicz et al., 2001) which has successfully detected some inconsistencies.

Detecting unsatisfiability means that there is an inconsistency between a prediction of the theory and what appears to be the case in the real world. Clearly the “weight” given to the inconsistency depends on the strength of the association rules involved. It may be necessary to detect several such inconsistencies to conclude that the theory should be revised. Different methods and parameter settings for data mining should also be used to try to replicate the inconsistencies independently. It may be possible to use other kinds of pattern recognition and mining, provided that their results can be expressed as logical formulae.

## Proof of Concept Implementation

The simulation component of the architecture in the AIMSS prototype is a simple proof-of-concept, based on an example housing scenario. This scenario is about the circumstances and needs of households moving to the social rented sector, and emphasises qualitative measures such as an agent’s perception of whether its needs are met.

The environment for the agents is an abstract “housing space” that may be divided into regions. Households are allocated randomly to regions initially with varying densities.

Precise densities and other attributes of each region (such as crime level etc.) can be specified as parameters.

For the feasibility study, we used a database of moves into the social rented sector for the whole of the UK for one year. This is known as CORE (Continuous Recording) dataset. Each CORE record include fields such as household details (age, sex, economic status of each person, total income), new tenancy (type of property, number of rooms, location), previous location and stated reason for move (affordability, overcrowding etc).

The simulation is a sequence of moves from one house to another for a typical housing scenario over a period of time measured in “cycles”. The CORE data contains actual moves that were recorded in a particular area (England).

## Ontology and model details

Static entities in the simulation include households, homes and regions. All these entities have a current state, which can change. For example, households have income, household size, and current home. Homes are of different types and sizes, can be expensive or cheap and have good or bad condition (among other properties). They are located in different neighbourhoods (regions) which themselves have properties such as population density, level of crime and availability of services. Households are agents. Each household has a set of needs that must be satisfied.

Dynamic behaviour comprises the decision rules for an agent as well as probabilistic rules for dynamic changes in the environment and household status (changes in income etc.) The way in which these entities are initialised should also be stated as part of the model, as this requires domain knowledge (e.g. initial densities of population and houses etc.)

An example configuration is where the space is divided into 4 “regions” (R1-4): R1: expensive, small city centre apartments; R2: inexpensive cramped city tower blocks; R3: Modest suburb; R4: Wealthy suburb (large expensive houses). The simulation is a sequence of steps in which agents decide whether they want to move based on a prioritised sequence of rules. These rules are simplified assumptions about decisions to move. Each rule has the form:

if (condition i not satisfied) then look for homes satisfying conditions 1, .., i

where i is the position in a priority list which is indexed from 1 to n. For example, if condition 1 is “affordability”, this is the first condition to be checked. If the current home is not affordable, the agent must move immediately and will only consider affordability when selecting other homes (as it is under pressure to move and has limited choice). The agent will only consider conditions further down the list if the conditions earlier in the list are already satisfied by its current home. For example, if the agent is relatively wealthy and its current housing is good, it may consider the pollution level of the neighbourhood as being too high. Before moving into a new home, it will take into account the pollution level of the new area, along with all other conditions that were satisfied in the previous home (i.e. it must be affordable etc.).

In an example parameter setting, the conditions may be prioritised in the following order: “affordability”, “crime-level”, “living-space”, “condition-of-home”, “services-in-neighbourhood” and “pollution-level”. Those agents that find available homes move into them, but only a limited number are vacant (depending on selected parameters). An agent becomes “unhappy” if it cannot move when it wants to (e.g. because its income is too low).

Clearly, the above scenario is extremely simplified. For example, the decision rules do not depend on the actions of their neighbours. Since this is a proof-of-concept, the simulation component was kept simple in order to test the feasibility of the AIMSS architecture as a whole. According to the incremental prototyping methodology, we expect that the simulation can be gradually scaled up by successively adding more features. More details on the simulation are in (Kennedy et al., 2007).

### Data mining

The following are some example rules that were mined from the simulation data using the agent rules above and environmental parameters guided by domain specialists

*S1: if (incomeLevel=1 and moveReason=affordability) 283*
*then newHomeCost=1 283 conf(1)*

This specifies that if the income level is in the lowest bracket and the reason for moving was affordability then the rent to be paid for the new home is in the lowest bracket.

The following is an example from the CORE data:

*D1: if (moveReason=affordability and incomeLevel=1) 102*
*then newHomeCost=2 98 conf(0.96)*

This has a similar form to S1 above, except that the new home cost is in the second lowest bracket instead of the lowest. Use of the satisfiability checker *zChaff* detected this inconsistency. This is, however, a very weak inconsistency because the data is numeric and had to be divided into intervals to be processed by Association Rule Mining. This kind of data mining and the satisfiability checking have no way of taking into account how close the prices actually are, only that they fall under different labels. To overcome these limits, a suite of different algorithms and methods would be required in an operational system.

## How can this help the social sciences?

The AIMSS architecture is intended to be useful for sociologists and policy researchers attempting to understand causal interactions in order to advise policy decision-makers. Causal explanations are difficult, since an extremely large number of variables may be possible contributors to an explanation of an observed phenomenon. Significant data acquisition costs make it infeasible to conduct surveys on all of them.

Simulations can help to manage this kind of complexity because they can help to visualise the consequences of a theory unfold over time and may generate unexpected emergent properties from an initial set of assumptions about agent behaviour. Such assumptions can include responses to needs (such as living space and affordability etc.) and reactions to the behaviour of others. This leads to the possibility of testing whether these emergent properties from the simulation actually exist in the real world, therefore focusing the data collection efforts in order to specifically find these properties. The resulting data collection may find some support for the simulation predictions but it may also point to additional variables that should be investigated, leading to ontology extension.  If the data seem to contradict the predictions, the assumptions themselves may be revised.

### Benefits of Automation

The user can compare the simulation predictions with real world observations and iteratively adapt the model according to the discovery loop above. However, this process may be infeasible to do manually due to the complexity of the simulation and data. Therefore the following advantages from automation may be expected:

1. Automated interpretation: Users may overlook an emergent property of the model that is difficult to detect visually. This may be due to preconceptions or the limits of the visualisation itself. The use of machine pattern recognition (such as data mining) applied to simulation-generated data can draw attention to these hidden developments (Kennedy et al., 2007).  In the same way, data mining and content extraction may be applied to real-world data. Since the results of these automated processes are sets of logical statements (such as association rules), the processes are expected to complement visual inspection.

2. Automated data selection and configuration: since the interpretation of the simulation content can be automated, the result can be used to steer the data selection. This may lead to more effective exploitation of available data (for example, online social networks). In addition, the automated data selection may be combined with data integration tools, resulting in new connections being found between apparently unrelated data sources.

3. Automated consistency-checking can reduce errors and may reveal hidden inconsistencies. Detailed comparisons between simulation predictions and data analysis results can be error prone and time-consuming.

All of the above are methods of assisting with the discovery process.  Since the DDDAS process is a feedback loop, the redirection of data collection and analysis may lead to surprising outcomes and lead to unexpected theory revision trajectories. The processes will not be completely automated, however, since expert knowledge and "common sense" is often necessary. Instead, we plan for the system to be used as an interactive  "workbench" with options for different levels of automation.

### Towards autonomous adaptation

We did not include the autonomous adaptation of the model itself within the AIMSS project, as this is expected to be more relevant to numerical data in physical systems than for social science data. One possibility is that the model may be adapted by running many simulations in parallel and applying an evolutionary algorithm to find the best fit for the data, thus greatly increasing the space of possibilities that can be searched realistically. This may yet be possible with qualitative content, where we define the "best fit" as the one that minimises logical inconsistencies between simulation content and data content. However, we do not yet know the feasibility of this in practice.

## Conclusions and Future Work

The proof-of-concept has shown the feasibility of applying DDDAS to qualitative data in the social sciences. This is demonstrated by the use of association rule mining and logical consistency-checking, both of which are suited particularly to this kind of data. One limitation of the current prototype is that the database we used was not sufficiently qualitative to thoroughly test these tools.

Further work will involve the addition of online data selection to the interactive workbench, along with a repository showing the relevant grid-enabled data sources and a selection of tools for their preprocessing and analysis. A major challenge is the automated search for data sources that potentially “match” the simulation content. These may, for example, contain some relevant details that can be integrated together from multiple sources. It is necessary to ensure that the ontology used to interpret simulation content is interoperable with ontologies used to classify data content.

A future challenge for the simulation is to add more cognitive richness to the agents so that their attitudes and preferences become more complex. This opens up the possibility of using text mining to extract content from real-world data showing attitudes and emotions. These may be the results of survey interviews corresponding to a scenario being simulated (e.g. attitudes of people who are unable to move from a high crime housing area). Such surveys may themselves be prompted by emergent properties in the simulation that were not foreseen. Another possibility is the use of online textual content as data sources (e.g. blogs, forums etc.) in order to test theories about attitudes. In the case of automated search for such sources, the repository (or index) of available data would be dynamic, as new data sources are actively discovered and classified according to the various ontologies.

### Ontological pluralism

Another limitation of the approach we have taken is that the model-building process is determined by a single initial model and set of concepts (ontology). The interpretation of the data is also constrained by these concepts. The limitations of a single ontology to classify and interpret social science data have already been identified (e.g. Edwards, 2006). In future work we plan to capture multiple ways of describing the events to be modelled by involving representatives of different social groups (stakeholders) in the initial model-building process. Multiple ontologies can lead to multiple ways of generating data from a simulation (or possibly even multiple simulations). Analysis of simulation predictions and real world observations is then not dependent on a single interpretation. Therefore the fault-tolerance of the system can be enhanced.

## References

Agrawal, R. and Srikant, R. (1994): Fast Algorithms for Mining Association Rules in Large Databases. In *Proceedings of the International Conference on Very Large Databases*, pages 478–499, Santiago, Chile: Morgan Kaufmann, Los Altos, CA, 1994.

Birkin, M., Turner, A. and Wu, B. (2006): A Synthetic Demographic Model of the UK Population: Methods, Progress and Problems. In: *Second International Conference on e-Social Science,* Manchester, UK (2006)

Darema, F. (2005): Grid Computing and Beyond: The Context of Dynamic Data Driven Applications Systems. *Proceedings of the IEEE: Special Issue on Grid Computing,* volume 93, number 3, March 2005, pages 692—697.

Davies, H. T. O., Nutley, S. and Smith, P. C. (2000): *What works? Evidence-based Policy and Practice in Public Services*. Bristol: The Policy Press, 2000.

Edwards, P. (2006): Cloudbuilding: Semantic Imprecision in Social Science Data (and how to live with it). In *Second International Conference on e-Social Science*, Manchester, UK, June 2006.

Entriken, R. and Wan, S. (2005): Agent-Based Simulation of an Automatic Mitigation Procedure. *Proceedings of the 38th Hawaii International Conference on System Sciences*, 2005.

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

Happe, K., Kellermann, K. and Balmann, A (2006): Agent-based analysis of Agricultural Policies: an Illustration of the Agricultural Policy Simulator AgriPoliS, its Adaptation, and Behavior. *Ecology and Society* 11(1): 49. URL: http://www.ecologyandsociety.org/vol11/iss1/art49/

Kennedy, C., Theodoropoulos, G., Ferrari, E., Lee, P., and Skelcher, C. (2007): Towards an Automated Approach to Dynamic Interpretation of Simulations. In: *Proceedings of the Asia Modelling Symposium* 2007, Phuket, Thailand (2007)

Lee, P., Ferrari, E., Kennedy, C., Theodoropoulos, G., and Skelcher, C. (2006): Assisted Model Building in the Social Sciences using Data Driven Simulation. *Second International Conference in e-Social Science*, Manchester, UK. June 2006.

Levitas, R. (2005): *The Inclusive Society? Social exclusion and New Labour*, Basingstoke: Palgrave, 2005

Mitchell, M. (1998): *An Introduction to Genetic Algorithms.* MIT Press.

Moskewicz, M., Madigan, C., Zhao, Y., Zhang, L. and S. Malik (2001): *Chaff:* Engineering an Efficient SAT Solver. In: *Design Automation Conference (DAC 2001)*, Las Vegas (2001)

Widergren, S., Sun, J., and Tesfatsion, L. S. (2006): Market Design Test Environments. *Proceedings of the IEEE Power Engineering Society*, Piscataway, NJ, 2006.

Witten, I.H. and Frank, E. (2005): Data Mining: Practical Machine Learning Tools and Techniques. Elsevier, San Fransisco, California (2005)