AIMSS: An Architecture for Data Driven Simulations in the Social Sciences

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

[1] School of Computer Science, University of Birmingham, UK
[2] School of Public Policy, University of Birmingham, UK
C.M.Kennedy,G.K.Theodoropoulos@cs.bham.ac.uk

Abstract. This paper presents a prototype implementation of an intelligent assistance architecture for data-driven simulation specialising in qualitative data in the social sciences. The assistant architecture semi-automates an iterative sequence in which an initial simulation is interpreted and compared with real-world observations. The simulation is then adapted so that it more closely fits the observations, while at the same time the data collection may be adjusted to reduce uncertainty. For our prototype, we have developed a simplified agent-based simulation as part of a social science case study involving decisions about housing. Real-world data on the behaviour of actual households is also available. The automation of the data-driven modelling process requires content interpretation of both the simulation and the corresponding real-world data. The paper discusses the use of Association Rule Mining to produce general logical statements about the simulation and data content and the applicability of logical consistency checking to detect observations that refute the simulation predictions.

Keywords: Architecture,Data Driven Simulations, Social Sciences.

1 Introduction: Intelligent Assistance for Model Development

In earlier work[1] we proposed a conceptual architecture for the intelligent management of a data driven simulation system. In that architecture, a software "assistant" agent should compare simulation predictions with data content and adapt the simulation as necessary. Similarly, it should adjust the data collection depending on simulation predictions. In this paper, we present a proof-of-concept prototype that is being developed as part of the AIMSS project[1] (Adaptive Intelligent Model-building for the Social Sciences). This is an exploratory implementation of the conceptual architecture.

A key issue the AIMSS project is trying to address is "evidence based model development": this can be understood 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;

---

[1] http://www.cs.bham.ac.uk/research/projects/aimss/

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 goal of the AIMSS project is to investigate the role of DDDAS in the automation of this process for the social sciences. The project is focusing on qualitative data and agent-based models.
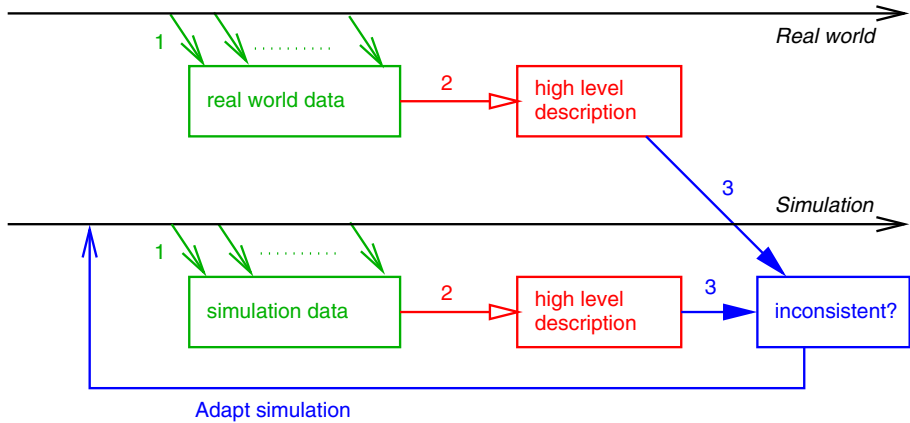
Fig. 1. Data driven adaptation of a simulation

A schematic diagram of the AIMSS concept of data-driven adaptation is shown in Figure 1. We can think of the simulation as running in parallel with events in the real world, although in a social science scenario, there are two important differences:

1. The simulation does not physically run in parallel with the real world. Instead, the real world data is usually historical. However, the data input could be reconstructed to behave like a stream of data being collected in parallel).

2. The simulation is abstract and does not correspond to a particular observed system. This means that the variable values read from the data cannot be directly absorbed into the simulation as might be typical for a physical model. Instead, the simulation represents a sequence of events in a typical observed system. The data may be collected from multiple real world instances of the general class of systems represented by the simulation. (For example, the simulation could be about a typical supermarket, while the data is collected from multiple real supermarkets).

The figure can be divided into two processes: the first is a process of interpreting both the simulation predictions and the 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. In the current version of the prototype we have focused on the first process. This requires the following automated capabilities:

- Interpretation of simulation states (at regular intervals or on demand): due to the abstract and qualitative nature of the simulation, this is not just about reading the current variable values, but about generating a *high level description* summarising patterns or trends.
- Interpretation of real world data: the same methods are required as for the simulation interpretation, except that the data is often more detailed and is usually noisy. Therefore pre-processing is required, which often involves the integration of data from multiple sources and the generation of higher level datasets that correspond to the simulation events.
- Consistency checking to determine whether the simulation states mostly agree with descriptions of data content.
- Re-direction and focusing of data collection in response to evaluation of simulation states or uncertainty in the data comparison.

It is expected that the simulation and data interpretation will be complemented by human visualisation and similarly the consistency/compatibility checking may be overriden by the "common sense" judgements of a user. Visualisation and compact summarisation of the data are important technologies here.

2   Social Science Case Study

As an example case study, we are modelling agents in a housing scenario, focusing on the circumstances and needs of those moving to the social rented sector, and emphasising qualitative measures such as an agent's perception of whether its needs are met.

We have implemented the agent-based simulation using RePast[2]. The environment for the agents is an abstract "housing space" that may be divided into subspaces. One example scenario is where the space is divided into 4 "regions" (R1-4): R1: expensive, small city centre apartments; R2: inexpensive cramped city towerblocks in a high crime area; R3: Modest suburb; R4: Wealthy suburb (large expensive houses with large gardens). At initialisation, homes are allocated randomly to regions with largest number in inner city and city centre. Households are allocated randomly to regions initially with varying densities. A household is represented by a single agent, even if it contains more than one member. Precise densities and other attributes of each region (such as crime level etc.) can be specified as parameters.

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

---

[2] http://repast.sourceforge.net/

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 satsified 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.). We have used a default scenario where the conditions are 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. The decision rules do not depend on the actions of neighbours. Since this is a proof-of-concept about intelligent management of a simulation, the actual simulation model itself is a minor part of the prototype. According to the incremental prototyping methodology, we expect that this can be gradually scaled up by successively adding more realistic simulations. More details on the simulation are in [2].

2.1 Data Sources

For the feasibility study, we are using 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).

3   Interpretation and Consistency Checking

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. [3]).

In the AIMSS prototype, both these components are specified in XML. Later we will consider the use of OWL[3]. The XML specification also includes the agent rules. These specifications are machine-readable and may potentially be modified autonomously. We are building on existing work on this area [4].

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.

3.1   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 [5], which is available in the WEKA Machine Learning package [6]. Association rules are a set of "if ... then" statements showing frequently occuring 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 classes 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 ... 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.

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)
```

---

[3] Ontology Web Language: http://www.w3.org/2004/OWL/

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.

3.2 Consistency-Checking

Assuming that the CORE data is "typical" if sampled for a minimum time period (e.g. a year), the simulation can also be sampled for a minimum number of cycles beginning after a stabilisation period. The simulation-generated rule above is an example prediction. To test it, we can apply consistency checking to see if there is a rule that was discovered from the data that contradicts it. This would indicate that the available data does not support the current model. Some existing work on postprocessing of association rules includes contradiction checking. For example, [7] uses an "unexpectedness" definition of a rule, given previous beliefs. These methods may be applied to an AIMSS type architecture, where the "beliefs" are the predictions of a simulation.

Efficient algorithms for general consistency checking are available, e.g. [8]. We are currently investigating the application of such algorithms to our work and have so far detected simple inconsistencies of the type between S1 and D1 above.

4 Towards Dynamic Reconfiguration and Adaptation

Work is ongoing to develop mechanisms to dynamically adjust the data collection and the simulation. Data mining often has to be fine-tuned so that the analysis is focused on the most relevant attributes. The rules generated from the simulation should contain useful predictions to be tested and the rules generated from the data have to make statements about the same entities mentioned in the prediction. Data mining parameters may be adjusted, e.g. by selecting attributes associated with predicted negative or positive outcomes.

The consistency checking may still be inconclusive because there is insufficient data to support or refute the prediction. In this case the ontology should contain pointers to additional data sources, and these may be suggested to the user before data access is attempted.

The dynamic adjustment of data collection from the simulation is also important so that focusing on particular events is possible (as is the case for the real world). Currently the data generated from the simulation is limited and only includes house moves that have actually taken place. This can be extended so that data can be sampled from the simulation which represents different viewpoints (e.g. it may be a series of spatial snapshots or it focus on the dynamic changes in the environment instead of actions of agents).

4.1   Adaptation and Model Revision

The ontology and behaviour model may be adapted, since they are represented in a machine-readable and modifiable form. Possible forms of adaptation include the following:

- Modify the initial values of parameters (such as e.g. initial density of homes in a particular kind of region) or the probabilities used to determine the initial values or to determine when and how they should change.
- Add new attributes or extend the range of values for existing attributes as a result of machine learning applied to the raw data.
- Modify agent behaviour rules or add new ones;
- Modify the order of execution of behaviour rules.

Note that behaviour rules are intended to give a causal explanation, while association rules merely show correlations. Furthermore, association rules may represent complex emergent properties of simple behaviour rules.

Populations of strings of behaviour rules may be subjected to an evolutionary algorithm (such as genetic algorithms [9]) to evolve a simulation that is most consistent with the reality in terms of behaviour. Behaviour models that are most "fit" can be regarded as good explanations of the observed data. However, domain experts would have to interact with the system to filter out unlikely behaviours that still fit the available data.

4.2   Limitations of the Current Approach

One limitation of the current prototype is that the pre-processing of the raw data is too much determined by artificial boundaries. For example, Association Rule Mining requires that numeric values are first divided into discrete intervals (e.g. "high", "medium", "low" for income and house prices). The problem of artificial divisions can be addressed by the use of clustering [10] to generate more natural classes, which can then be used as discrete attribute values for an Association Rule miner. Conceptual Clustering [11] addresses the need for clusters to relate to existing concepts. Instead of just relying on one method, a combination of different pattern recognition and machine learning methods should be applied to the different datasets.

Another limitation of the approach we have taken is that the model-building process is determined by a single interpretation of the data (i.e. one ontology). 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.

5   Conclusion

As part of the AIMSS project we have developed a simple prototype demonstrating some of the features required for the assistant agent architecture presented in an earlier

study. Although this prototype is far too simple to be used operationally to generate real-world models, it serves as a proof-of-concept and can be used as a research tool by social scientists to help with exploratory model building and testing. Future work will involve the development of more realistic simulations, as well as the use of a wider range of data analysis and machine learning tools.

Acknowledgements

References

1. Kennedy, C., Theodoropoulos, G.: Intelligent Management of Data Driven Simulations to Support Model Building in the Social Sciences. In: Workshop on Dynamic Data-Driven Applications Simulation at ICCS 2006, LNCS 3993, Reading, UK, Springer-Verlag (May 2006) 562–569
2. Kennedy, C., Theodoropoulos, G., Ferrari, E., Lee, P., Skelcher, C.: Towards an Automated Approach to Dynamic Interpretation of Simulations. In: Proceedings of the Asia Modelling Symposium 2007, Phuket, Thailand (March 2007)
3. Birkin, M., Turner, A., Wu, B.: A Synthetic Demographic Model of the UK Population: Methods, Progress and Problems. In: Second International Conference on e-Social Science, Manchester, UK (June 2006)
4. Brogan, D., Reynolds, P., Bartholet, R., Carnahan, J., Loitiere, Y.: Semi-Automated Simulation Transformation for DDDAS. In: Workshop on Dynamic Data Driven Application Systems at the International Conference on Computational Science (ICCS 2005), LNCS 3515,, Atlanta, USA, Springer-Verlag (May 2005) 721–728
5. Agrawal, R., Srikant, R.: Fast Algorithms for Mining Association Rules in Large Databases. In: Proceedings of the International Conference on Very Large Databases, Santiage, Chile: Morgan Kaufmann, Los Altos, CA (1994) 478–499
6. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. Elsevier, San Fransisco, California (2005)
7. Padmanabhan, B., Tuzhilin, A.: A Belief-Driven Method for Discovering Unexpected Patterns. In: Knowledge Discovery and Data Mining. (1998) 94–100
8. Moskewicz, M., Madigan, C., Zhao, Y., Zhang, L., Malik, S.: Chaff: Engineering an Efficient SAT Solver. In: Design Automation Conference (DAC 2001), Las Vegas (June 2001)
9. Mitchell, M.: An Introduction to Genetic Algorithms. MIT Press (1998)
10. Jain, A.K., Murty, M.N., Flynn, P.J.: Data Clustering: A Review. ACM Computing Surveys **31**(3) (September 1999)
11. Michalski, R.S., Stepp, R.E.: Learning from Observation: Conceptual Clustering. In Michalski, R.S., Carbonell, J.G., Mitchell, T.M., eds.: *Machine Learning: An artificial intelligence approach*. Morgan Kauffmann, Palo Alto, CA:Tioga (1983) 331–363