ANTS: Articulated Nested Telescoping Simulations

An articulated multi-level particle system for the interactive simulation of large data sets

MARY DECKER
University of Cape Town
and
ALEXANDER KARPUL
University of Cape Town

This paper addresses the problem of scale in simulations by investigating a novel kind of interactive simulation system which can be run on desktop machines.

In this system, data is represented at various levels of abstraction. The system is navigated in such a way that the data can be viewed and modified at each of those levels. However, at deeper levels in the model, only a subset of the total system is represented in detail. This subset is generated, as the user moves down a level, out of aggregates in the level above and compressed into aggregates, when the user moves up a level. Simulation rules are applied to each level of the simulation in a modular manner, independent of the level of the simulation.

This system is found to increase efficiency dramatically, since only a subset of the data is represented in detail. A certain amount of data loss inevitably occurs as a result of data compression. The system is therefore only appropriate in those cases where data precision is not of paramount importance.

An interface was designed by means of the PICTIVE participatory design technique. A subset of this interface design was implemented. It was then evaluated using the constructive interaction method and was found to be generally intuitive and user friendly.

Categories and Subject Descriptors: E.1. [**Data Structures**]; I.6. [**Simulation and Modeling**]: Simulation Theory, Applications, Gaming
General Terms: simulation, data structure, multi-level
Additional Key Words and Phrases: abstraction, large

---

1 INTRODUCTION

As computing power increases and memory becomes available in larger quantities, the human appetite for computing power increases too. This trend is particularly apparent in the field of simulation where models can become increasingly large.

In fact, David Nicol [1] makes the assertion that to all challenges facing the world of simulation may be added the statement: This challenge becomes all the more formidable with increasing model size.

This project addresses the problem of scale by creating a novel kind of interactive simulation system and evaluating the advantages and limitations of such a system.

2 BACKGROUND

In order to gain a background to this problem it was helpful to examine a body of work from the Winter Simulation Conference. Those papers on the topic of simulation in general were particularly useful, such as the *Introduction to Simulation* by Shannon [2], which raised many of the general issues pertinent to the field of simulation.

Other extremely useful papers were those which addressed the challenges faced in the field of simulation and potential directions for research in the future. An example of such a paper is *Strategic Directions in Simulation Research* [1]. It is of particular interest with regard to assertions by David Nicol concerning the problem of scale which pervades the world of computer simulation and complicates all other challenges which simulationists face. Another such paper is the *Twenty-Fifth Anniversary Keynote Address* [3] for the Winter Simulation Conference of 1992. In this paper, Joseph Sussman makes mention of how levels of detail in simulation can have an impact on computational feasibility and numerical precision. More specifically he mentions how greater detail in simulation usually leads to greater precision in simulation results but at the expense of computational efficiency.

In designing and evaluating the interface, a user-centred design approach was adopted. The PICTIVE method of interface design was adopted and work by Muller [4, 5], Preece [6] and Spinuzzi [7] was particularly helpful in preparing for this. The interface evaluation was performed using the constructive interaction method and work by Kahler [8] and Marsden [9] was very useful in this regard.

3 APPROACH

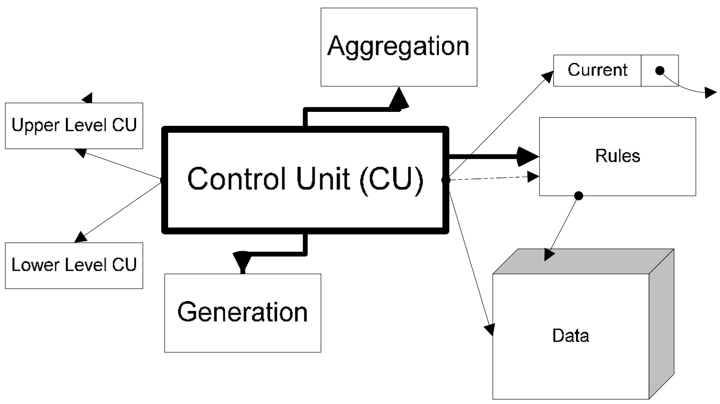
In the system created for this research, data is represented at various levels of abstraction. It is navigated in such a way that the current state of the data can be viewed and modified at each of those levels. However, at deeper levels, only a subset of the total system is represented. This subset is generated, as the user moves down a level, out of aggregates at the level above and compressed into aggregates when the user moves up a level. All the while, as the simulation runs, simulation rules are run on the data at different levels to update it in a manner that is consistent with the multi-level nature of the world being simulated.

A simple example of such a system (and the example we chose to model) is that of an ant world . In such a world, data can be represented at three different levels of abstraction: world level, land mass level and colony level. The example of an ant world was chosen for this project because of the relative simplicity of the system. The ant world represented was not intended to be biologically accurate but was intended rather as a simple and intuitive example. The methodology presented is applicable to a far wider variety of systems.

This methodology makes use of an interlinking and integration of levels which we term articulating the levels . This term derives from the work of Willem Doise [10], who formalised this concept in the field of social psychology. We use this term to indicate the navigation between levels, together with inter-level update of information. By the method of articulation, we hope to deal with the problem of scale.

Because of the way in which data is condensed into aggregates and constructed out of aggregates, it is understood that results of the simulation produced are intended to be plausible, rather than accurate. If a high degree of accuracy is required in simulation, it would be better to use supercomputers and parallel processing methods.

4 CONTROLLER DATA STRUCTURE



The Control Unit (CU) has the following components:

- *Data*: this is where all data that is used for the simulation is stored.

- *Rules*: these rules change the state of the data at the appropriate steps.
- *Generation*: this is a generic function that generates a lower level CU with data when called.
- *Aggregation*: this is also a generic function, but aggregates the data into a higher level piece of data.
- *Upper Level CU*: this stores a link to the higher level CU that spawned this CU
- *Lower Level CU*: this stores a link to a CU that this CU has generated

5 GENERATION AND AGGREGATION OF DATA

When the method for generation is called it creates a new CU, storing new objects and a reference to the rules that act upon them. The new CU also stores a pointer to the previous CU which spawned it. This concept allows the levels to be completely distinct but still linked so that information can pass back up. The new CU also holds a link to aggregation and generation methods specific to the level. The CU stores a list of which original higher level data instances generated the classes that are stored inside the data block for this new CU. This is important in order to determine what has to be changed at the higher up CU. The higher level data is prevented from being accessed by the rules at that level, since the information has become too specific for those rules to act upon. Each level is updated, using the rules, before updating the levels above it.

Moving up the levels, the data structure needs to aggregate the data that was generated, back into a more general form. This allows the simulation to use the newly generated data in the rest of the simulation. In order to do this, it first calls the aggregation function. This function compresses the generated data into its aggregated form, also called the higher level data, for which the lower level CU has a link stored. The aggregation function moves the aggregated data back into the higher level CU in the correct place in the higher level CU data block. This function then sets the lower level CU pointer in the higher level CU to null. Finally, it deletes the lower level CU and destroys all the data it points to.

6 SIMULATION RULES

In the interests of general applicability, it was important that the interface to the rules be as general as possible. It was decided that a class would be created for each level of the simulation which would invoke the rules in a black box fashion, so that the calling class would not need to know the internal workings of this class. The calling class would merely pass this class a link to the controller data structure as well as a time step variable to indicate the time in the simulation.

These level-specific classes all inherit from a common abstract class. Within these classes, work is allocated to functions in other defined classes which model behaviour initiated by a specific entity, such as an ant.

In order to determine whether the behaviour of the data in the system is consistent from one level to the next, it is necessary to run the simulation at each of the levels in the model and compare results. Tests of this nature were performed on the pilot simulation and showed that a high level of consistency could be attained between levels, with slight variations between levels where the data is updated in a different manner, such as updating age for each individual ant or for the entire population.

7 EFFICIENCY AND DATA LOSS

As a result of compressing up a level to aggregates, certain types of data in this simulation are automatically lost, such as the exact position of individual ants. Other kinds of data are retained, such as the number of ants in a colony which is stored as a population. Some attributes, such as the strength of an ant, were stored as averages and statistical deviations when compressed up, so that they are stored only in a general, aggregated sense.

The chief rationale behind settling for this data imprecision is an improvement in time efficiency. In the pilot project created, the simulation ran 20 times faster than a conventional simulation running all of the data at a detailed level.

If data accuracy is important in the simulation, in cases where the results of the simulation are to be used to make important decisions of policy, then it is advisable to rather make use of a simulation system in which data compression is not used. However, in situations such as teaching (where being able to illustrate a concept on a desktop machine is the focus) or game playing (where real-time interactivity is more important than accuracy) this system is ideal.

8 INTERFACE DESIGN, IMPLEMENTATION AND EVALUATION

The interface was designed using the PICTIVE design technique. In this technique, users, developers and other stakeholders collaborate on an equal footing in designing the functionality and the look and feel of the software, making use of low-fidelity design materials such as pens, paper, post-it notes, etc [4].

Two design sessions were held and the following most prominent suggestions were made for the interface:

- A central play area should exist in which the user can interact with entities at a specific level.
- Modification of entity attributes should be possible, which would require pausing of the simulation.
- Navigation should take place by clicking the appropriate entity to examine in greater detail or by making use of navigation buttons alongside.
- A databoard , similar to the clipboard in popular word-processing packages, would be desirable. This would allow the user to take snapshots of an entity at a specific point in time.

For the pilot project, a subset of this functionality was implemented. The interface produced was then evaluated using constructive interactive method. Users found it to be user-friendly and intuitive. Users were easily able to identify possible applications to such a system in their fields of study, which included finance and social sciences.

9 RESEARCH CONCLUSIONS

- The system implemented is proof that the architecture envisioned can indeed be constructed:
  - The control unit data structure envisioned was shown to be practically implementable.
  - Data generation and compression were successfully performed on this data structure by means of statistical aggregation.
  - Rules were implemented in separate units at different levels of abstraction with a common mode of access, independent of the level.

- Issues of representation of such a system were addressed:
  - An appropriate user interface was designed with the input of user-centred design methodologies.
  - A subset of the designed features was implemented and met with general user approval.
- The following advantages and limitations of the system were observed:
  - The major advantage of the system produced is that it was found to improve dramatically on time efficiency for simulations of large data sets.
  - The major disadvantage of the system is that the statistical aggregation, which is necessary, does result in some data loss. This limits the applicability of the system to those simulations where data precision is not of paramount importance.

ACKNOWLEDGMENTS

Many thanks go to Mr Dave Nunez and Dr Patrick Marais for their suggestions, advice and support in this research. In addition, in the area of participatory design, Dr Gary Marsden and Mr Donald Cook are also thanked for their suggestions, advice and for provision of materials for the participatory design session. Any errors or omissions in this project are entirely those of the authors.

REFERENCES

[1] PAGE, E., NICOL, D., BALCI, O., FUJIMOTO, R., FISHWICK, P., L ECUYER, P. AND SMITH, R. 1999. Strategic Directions in Simulation Research in *Proceedings of the 1999 Winter Simulation Conference*. (Phoenix, Arizona, 1999), ACM Press, 1509 - 1520.

[2] SHANNON, R. 1992. Introduction to Simulation in *Proceedings of the 24th conference on Winter simulation*. (Arlington, Virginia, 1992), ACM Press, 65 - 73.

[3] CRAIN, R., SUSSMAN, J., SCHRIBER, T., HENRIKSEN, J. AND ROBERTS, S. 1992. The Winter Simulation Conference: Celebrating Twenty-Five Years of Progress in *Proceedings of the 1992 Winter Simulation Conference* (Arlington, Virginia, 1992), ACM Press, 3 - 33.

[4] MULLER, M. 1991. PICTIVE -An Exploration in Participatory Design in *Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology*. (New Orleans, Louisiana, 1991), ACM Press, 225 231.

[5] MULLER, M. 1992. Retrospective on a Year of Participatory Design using the PICTIVE Technique. *Proceedings of the SIGCHI conference on Human factors in computing systems*. Monterey, California, ACM Press, 1992, 455-462.

[6] PREECE, J., ROGERS, Y. AND SHARP, H. 2002. *Interaction Design: beyond human-computer interaction*. John Wiley and Sons, Hoboken, NJ, 2002.

[7] SPINUZZI, C. 2002. A Scandinavian challenge, a US response: methodological assumptions in Scandinavian and US prototyping approaches in *Proceedings of the 20th annual international conference on Computer documentation*. Toronto, Ontario, ACM Press, 2002, 208 215.

[8] KAHLER, H., KENSING, F. AND MULLER, M. 2000. Constructive Interaction and Collaborative Work: Introducing a Method for Testing Collaborative Systems. *Interactions*, May / June, ACM Press, 2000, 27 - 34.

[9] MARSDEN, G. AND JONES, M. 2006. *Mobile Interaction Design*. John Wiley & Sons, January 2006.

[10] DOISE, W. (TRANSL. ELIZABETH MAPSTONE). 1986. *Levels of explanation in social psychology*. Cambridge University Press, 1986.

Authors' addresses: Mary Decker, Department of Computer Science, University of Cape Town, ZAF; 2, Alexander Karpul, Department of Computer Science, University of Cape Town, ZAF.