Journal of Visual Languages and Computing 23 (2012) 344–353

Contents lists available at SciVerse ScienceDirect

Journal of Visual Languages and Computing

journal homepage: www.elsevier.com/locate/jvlc

Facilitating insight into a simulation model using visualization and dynamic model previews ☆

Joost de Folter [a,*], Timothy Cribbin [b]

[a] Brunel Institute for Bioengineering, Brunel University, Uxbridge, UB8 3PH, UK
[b] People and Interactivity Research Centre, Brunel University, Uxbridge, UB8 3PH, UK

ARTICLE INFO

Article history:
Received 16 June 2011
Received in revised form
31 January 2012
Accepted 23 August 2012
Available online 18 September 2012

Keywords:
Visualization
Simulation
Dynamic query
Query preview
Insight evaluation
Counter-current chromatography

ABSTRACT

This paper shows how model simplification, by replacing iterative steps with unitary predictive equations, can enable dynamic interaction with a complex simulation process. Model previews extend the techniques of dynamic querying and query previews into the context of ad hoc simulation model exploration. A case study is presented within the domain of counter-current chromatography. The relatively novel method of insight evaluation was applied, given the exploratory nature of the task. The evaluation data show that the trade-off in accuracy is far outweighed by benefits of dynamic interaction. The number of insights gained using the enhanced interactive version of the computer model was more than six times higher than the number of insights gained using the basic version of the model. There was also a trend for dynamic interaction to facilitate insights of greater domain importance.

1. Introduction

Simulation is critical to solving many types of problems in science, design and engineering, providing a cheap and non-intrusive means of exploring the behavior of real-world systems under different conditions. However, simulation models are typically defined by a multi-dimensional parameter space wherein parameters interact in complex, often non-linear ways. This makes it difficult and time consuming to understand the behavior of a model across the full scope of the parameter space. An attractive solution is to replace command-line or form fill-in interfaces with dynamic query interaction. Dynamic querying is defined as "…the interactive user control of visual query parameters that generate rapid (100 ms update) animated visual display of database search results" [1]. Although database retrieval was the original target application, the interactive visualization technique dynamic queries has since been applied to many other domains including data-mining [1–3] and simulation [4,5].

Whilst dynamic query is perfectly feasible means of interacting with pre-computed datasets [4,5], generating ad hoc results in real-time is not possible using a typical simulation model, as complex calculations need to be applied to the state over many iterations. In this paper the authors propose that the concept of query previews [6] can be extended to the task of exploring the parameter space of a simulation model. Query previews provide only a summary indication of the effect of query change on the results of a database search, which is much faster to process than a full retrieval operation. To the authors' knowledge the concept of previews has not yet been applied to simulation modeling. This paper focuses on the case of a counter-current chromatography (CCC) model [7,8]. The target users of this model expressed a desire to explore parameter-performance relationships in real-time. However, the full CCC model can take anything between several seconds to minutes to compute, prohibiting the use of dynamic query over the parameter space.

☆ This paper has been recommended for acceptance by S. Levialdi.

* Corresponding author. Tel.: +44 1895 266856; fax: +44 1895 274608.
E-mail address: joost.defolter@brunel.ac.uk (J. de Folter).

1045-926X/$ - see front matter
http://dx.doi.org/10.1016/j.jvlc.2012.08.001

A method is described for creating model previews from discrete-event model, by substituting the full, iterative algorithm with predictive equations based on a subset of key parameters. Using this technique, a solution was developed in the form of the interactive CCC explorer (ICE). ICE achieves an acceptable estimate of the full model prediction in real-time, allowing the use of dynamic query control over a limited range of key parameters. As a further contribution the relatively untested insight evaluation methodology [9] was used to compare user experience on ICE with the existing (full) model interface. Whilst the users had control over fewer parameters, the direct manipulation afforded by the dynamic model previews resulted in a substantial increase in the rate of insight events of various types, including hypothesis generation and unexpected observations.

2. Exploring simulation models

Simulation is the act of imitating one process with another process, whereby a process is some system in which the state changes over time [10]. The process being imitated is generally some real-world system, represented in the simulation by a dynamic mathematical model. Discrete event simulation is a common approach wherein a simulation involves resolving a set of equations in an iterative fashion across a series of time intervals. At each subsequent interval the outputs from the previous iteration become the inputs. The investigator is able to specify the starting parameters of a model to the desired configuration and observe the change in system state over a specified time period. Furthermore, visualization of the system state (or constituent variables) can be employed to permit rapid perception of emergent patterns and trends as the process unfolds. Common simulation applications include testing theories, solving design problems and for running experiments that would be otherwise impractical due, for instance, to cost, time, or ethical issues.

In this paper we focus upon a simulation of a chemical process called counter-current chromatography [11]. The dynamic CCC model [7,8] describes the process of mixing and settling of a liquid or analyte in order to separate it into its component parts. The distribution of components is normally displayed as a line graph called a Chromatogram. In the original version of the model software, the user could set parameters and run the simulation and was able to visualize the results from a number of perspectives, including temporal and summative. These are described in more detail in section three. However, testing of this prototype amongst domain experts revealed that they wanted to be able to use the model in a more exploratory fashion. What they wanted was a means of dynamically exploring the interactions between the numerous parameters of the model. Unfortunately, as with many models, the run-time of the CCC simulation was quite lengthy taking from seconds to minutes depending upon the specification of parameters.

Whilst simulations entice the user to explore, they do not naturally lend themselves to this purpose. A natural question for a user to ask of a simulation model is: 'How do I achieve this desirable outcome?' Ironically, this is a particularly difficult question to answer because there is generally only a one-way relationship between parameters (input) and performances (output). All performances are functionally dependent upon the parameters; it is not possible to simply state a desired outcome and to identify a corresponding parameter configuration [4,12]. Instead, the user must adopt an exploratory strategy in order to establish the relationship between parameter space and the desired performances, which involves trial and error: testing a tentative hypothesis, evaluating the result and reformulating the hypothesis based on the feedback. This requires a fluid interaction cycle of querying (parameter specification) and feedback (performance results). Unfortunately, the lengthy run-time of a typical simulation model prohibits the ad hoc application of dynamic queries.

2.1. Dynamic queries

Despite these barriers, previous research has found ways to enable dynamic interaction with such models. A pervasive theme is the use of dynamic queries [1,13]. Dynamic queries were originally designed as an alternative to the traditional form fill-in database interface. Shneiderman [1] described the concept of dynamic queries as the interactive control of visual query parameters that generate rapid (<100 ms) animated visual display of database search results. Key to the approach is the application of the principles of direct manipulation and tight-coupling, or linking, whereby users incrementally modify parameters using graphical controls (e.g. range sliders, check-boxes and buttons) and the resulting change is shown immediately in a visual (generally graphical) overview of results. Shneiderman described the interaction experience as akin to "flying through the database", in order to emphasize the natural flow induced by the technique.

A logical solution to the problem of implementing dynamic query in a simulation environment is to pre-compute a database of simulation results. This approach is exemplified by Tweedie and Spence's Influence Explorer [4,12], an interface originally designed to support the task of finding an optimal electronic circuit design according to customer-specified performance criteria. Clearly, when a model is defined by multiple, continuous performance variables it is not practical to pre-compute all possible scenarios. Dix and Ellis [14] explain how random sampling can provide a solution to a large variety of hard problems, including those relating to visualization of large spaces. Influence Explorer solves the problem by sampling a random distribution of points (usually a few hundred) within a user-specified region of the parameter space. Simulations are then run using each of these configurations. The resulting database of parameter–performance relationships can then be queried in real-time. Perhaps most valuable, from a designer's perspective, was that this approach makes it possible to formulate performance queries and instantly see the parameter configurations that meet these constraints.