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, UK

[b] People and Interactivity Research Centre, Brunel University, Uxbridge, UK

** Brunel Institute for Bioengineering, Brunel University, Uxbridge, UB8 3PH, UK,*

Tel: +44(0)1895 266856, Fax: +44(0)1895 274608, E-mail: joost.defolter@brunel.ac.uk

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. 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.

This form of two-way, dynamic interaction was shown to be effective, not only for circuit design, but also for a number of task domains including financial design [15] and structural design [16]. However, CCC users were less interested in parameter-performance relationships. The critical requirements were that users could explore the temporal progression of the model and, most challenging of all, the relationship between parameters and the impact of these configurations on the final chromatogram.

Temporal exploration of simulations using dynamic queries has already been explored by Bosch [5]. They used dynamic query controls to enable the user to cycle through the progression of a run of a SimOS machine simulation of the Argus parallel rendering library and also to focus in on particular attributes of the recorded output. In the current case, the Time mode Chromatogram in the system allows the user to replay, in a non-linear fashion, the results of a run of the CCC model. This is described in further detail in section three.

Enabling ad hoc exploration of the model was more challenging. Tweedie and Spence's [12] sampling approach was considered. However, the size of the parameter space was a concern (as the model has a varying number of independent input parameters; 10 at minimum). One way to reduce simulation run-time is to reduce the resolution of the simulation [17]. However, this was not necessary as a more flexible solution presented itself in the form of a simplified version of the model, based on a set of predictive linear models rather than iterative equations. In earlier work De Folter [8] reported that the simplified model gave acceptable results for most CCC flow modes in near real-time. It was anticipated that providing even approximate chromatograms in real-time would provide useful sensitivity information that would facilitate exploration and the acquisition of insight.

2.2. Model previews

The idea of using dynamic model summaries was inspired by the concept of query previews. Query previews were originally developed to facilitate dynamic query interaction within a networked database system [6]. At the time, slow network access speeds along with client-side processing and memory capacity were a bottle-neck to implementing dynamic query for information retrieval tasks operating on large databases. Query previews form part of a two-stage query formulation process. In the preview stage, the user forms a rough query over a small number of attributes, using dynamic query controls to select keywords, date and other quantitative ranges. As the user adjusts their query, real-time feedback is provided about the volume of likely matches, plus other high-level metadata. This interaction is driven by a volume preview table that is downloaded at the start of the session. Once the user is happy with the previewed subset, they move on to the second phase of query refinement. More detailed metadata relating only to the identified subset is retrieved from the server, which is usually small enough to allow dynamic query interaction to continue, but at a lower, more specific level of detail than the preview stage.

Hence the motivating problem behind query previews is very similar to the current case, however here the lag is caused by the run-time of the algorithm rather than network speed. In a model preview, unlike both Influence Explorer and query previews, there is no pre-computed database. Instead a simplified model substitutes the full model to provide the first stage of the formulation process, presented as a stream of chromatograms updated in real-time as the parameters are adjusted. As the results reported later show, valuable insights into the CCC model can be achieved using the Interactive Configuration Explorer (ICE) without progressing to the second stage (i.e. running the full model).

2.3. Evaluating insight

ICE was a response to evaluations of the earlier model interface, comprising the Chromatogram and Time mode Chromatogram views. Both of these views were dependent upon execution of the full model, with the former visualizing the end result. The Time mode view provides the opportunity to explore the effects resulting from a single parameter configuration, allowing the user to identify key transitional points during the CCC process. In contrast, ICE was developed to allow users to gain a better understanding of the relationships between parameter configurations and the end results of the process.

The purpose of the evaluation study reported here was to determine the added value of the ICE (dynamic query / model preview) approach to the existing suite of visualization tools. This required a methodology that went beyond simple evaluation of usability or performance on pre-defined tasks. In recent years, there has been a movement in the Information Visualization (IV) field to find a more suitable means of evaluating interfaces that are essentially geared towards discovery: the acquisition of insight. A key approach here is that of Insight Evaluation [9,18,19,20]).

Insight is frequently touted the main purpose of IV, yet has proved devilishly hard to define, particularly at the operational (task) level. The key problem is that when one defines any kind of objective, one is limiting the scope for participants to make (and for experimenters to record) unexpected discoveries i.e. the “aha” events that characterize insight. Hence, rather than trying to formally define what an insight is, North simply suggests characteristics that one might expect an insight to have [9]. First, insights tend to be complex, involving relationships between large amounts of cases and attributes. Insights also tend to be deep, accumulating over time, with the results of questions and hypotheses leading to further new queries. They are also qualitative, being hard to define or otherwise subjective, unexpected in the sense that insights are often serendipitous. Finally, insights are about relevance, which is closely dependent upon existing domain knowledge.

Hence utilizing benchmark tasks makes poor sense for evaluating exploratory interfaces. Clear procedures and objectives leave little room for complex inferences. Time limits on trials leave little scope for achieving depth. Instead, for Insight Evaluation, North advocates an open-ended protocol, whereby domain experts, following basic training, are allowed to explore an interface in an unconstrained fashion [9]. This presents some new difficulties for experimenters, particularly in terms of collecting and coding the data. The need for domain experts restricted participant recruitment. Deciding how to collect and code the data was difficult; a code-sheet based on North's suggestions [9] was devised, recording for each insight event, the category or type, complexity, depth and an estimate of the time taken to generate the insight. Due to the limited participant population, we did not categorize type post-hoc, instead we applied a general distinction between results that were different from expectation, a new discovery and the creation of a new hypothesis.

The next section introduces the CCC model in more detail. This is followed by a description of the existing visual interface, followed by the improved interface, implemented using the simplified CCC model and the ICE. Finally, an insight evaluation based on North's proposed methodology [9] is presented.

3. Counter-Current Chromatography simulation model

Counter-Current Chromatography (CCC) is a liquid-liquid chromatography technique first introduced by Ito in 1966 [11]. The process is based on two separate media called phases. Although the phases are in constant contact with each other, they do not naturally mix. In general the phases might be solid, liquid or gaseous, although in CCC both phases are liquid. Usually there is a stationary phase and a mobile phase. In CCC the phases flow freely around the system in continuous tubing. However, the stationary phase is retained in the tubing due to a combination of hydrodynamic and hydrostatic forces. By introducing varying G-forces, mixing and settling steps promote the distribution of sample components between the phases. The different components of a sample are separated due to differences in their affinity (attraction) to each phase.

For the purpose of simulating CCC, the continuous tubing is replaced by discrete test tubes. This representation is also more suitable for illustrating the CCC process. Figure 1 provides a simple example in this format.

<Figure 1>

A universal CCC model was created to satisfy the need for a model versatile enough to take into account the numerous operating modes that can be used with CCC (i.e. normal flow, reverse flow, dual flow, co-current flow and intermittent flow mode) [8]. This model was validated with real separations from the literature and against established CCC partition theory. The software consists of a mathematical chromatography model realized as a C++ .NET application. A number of visual enhancements were added to the model using a combination of .NET and custom user interface components. The model is simulated by arrays of numbers representing the mass concentrations of sample components. For the arrays a dynamic storage type is used. Allowing array length to be dynamically changed provides a greater flexibility and optimization of available computer memory.

The value of the computational model is that it allows researchers to explore potential experimental scenarios without incurring the expense and time of a physical experiment. The standard visual output from the described model is the Chromatogram, showing the component concentrations coming out of the chromatography system. However, this only shows the end-state of the process. New views were developed by integrating the time dimension, providing the user with better comprehension of the separation process.

3.1. Chromatogram view

The visual output of any chromatography process, is called the chromatogram. The results of the model are displayed in a chromatogram style view (fig. 2). In the chromatography process, a number of components are first inserted at the start of the chromatography system (or column). At this point the component concentration is very high, so this would be represented as a very tall narrow concentration peak in the chromatogram. As the components travel through the column, the components are diluted in the process, reducing their concentration, resulting in broader peaks. Due to the difference in affinity, different components travel at different speeds through the column, whereby the different components are separated. In addition to the components that have come out of the chromatography system, it also shows the components still inside the system. Furthermore, a clear distinction is made of how the components are distributed over the phases [21], where the upper half of the screen represents the upper phase, and the lower half of the screen, the lower phase. It is also possible to show only the total components over both phases. This view is highly customizable, including the units (model steps, volume or time) and zoom level on the horizontal axis and the scaling on the vertical axis can be changed.

<Figure 2>

3.2. Time mode chromatogram view

A way to show how the CCC process develops over time was realized by adding time as a parameter to the chromatogram view (fig. 3). A slider control allows interactively moving to any point in time in the CCC process. Gradually moving the slider control shows how the chromatographic peaks develop as the process continues.

<Figure 3>

3.3. Rendering Performance Optimization

After the simulation model completes, its internal raw data is the main model output. The rendering process of this data is optimized for efficiency (fig. 4). First the internal raw data is converted to a generalized format. This format is highly compatible with a basic spreadsheet format, and at this point can be saved to a file for processing in a spreadsheet program like MS Excel. Next, the data is normalized into a vector format and is bound to a visualization component. This enables the component to render to screen very efficiently, rapidly handling any update (such as resizing and zooming in/out).

<Figure 4>

3.4. Simplified model

Running the model typically takes from a few seconds up to a few minutes to complete*. Shneiderman states that truly dynamic interaction means updates of less than 100 ms [1]. As reducing the complexity of the model would directly affect a key model parameter and would therefore not result in a representative model, a different approach had to be used. Instead, predictive equations were chosen to represent a simplified model [8]. This simplified model does not iterate through component movement and redistribution. A comparison between the complexity of the original model and simplified model using Landau notation (also known as 'Big O notation') follows here:

Normal model complexity: O(CNT)

Simplified model complexity: O(C)

Where C is the number of input components in a mixture (in the order of $10^{0\ldots1}$), N is the number of chromatography system cells (in the order of $10^{2\ldots3}$), and T is the total number of iterative steps to separate the components out of the chromatography system (in the order of $10^{3\ldots4}$). The estimated total difference in complexity is therefore in the order of $10^{5\ldots7}$. As the simplified model removes most of the iterations, it is therefore not only much faster, it also removes most of the variability of the calculation times the normal model has. This enables real-time interaction, as it typically only takes 1 to 10 ms* to complete. It is important to note though, that the simplified model is an approximation, and cannot substitute the normal model, which simulates the process in a natural way.

(* on a PC with a 3GHz Intel Core 2 Duo CPU)

4. The Interactive Configuration Explorer (ICE)

The motivation for the interactive configuration explorer (ICE) was to allow users to explore these interactions in real-time. ICE makes the query-output process more interactive by applying an established IV interaction technique known as dynamic querying (dynamic query) [1,13]. A dynamic query interface employs graphical ‘widgets’ (such as check-boxes and sliders) that allow the user to continuously manipulate the constraints of the query, whilst the output is presented visually with updates occurring in virtually real-time (less than 100 ms [1]). To improve the understanding of how the model responds to each parameter, the dynamic queries and query previews were combined into the interface of the computer model, where the dynamic queries provide the interactive parameter selection, and the query previews enable the real-time updates of the model. We call this dynamic model previewing.

The interactive configuration explorer shows the output of the simplified model in a chromatogram view, which is updated real-time when the input parameters are changed (fig. 5). Traditionally, slider controls are used to manipulate each input parameter. Though the sliders controls visually provide the parameter range and indicate their absolute value, it is difficult to allow for finer adjustments or non-linear scales. Many of the model parameters require a logarithmic adjustment and are furthermore very sensitive to small changes. To meet these requirements, a new type of control was designed, allowing for relative changes. The visual interface of this control is a springed ‘jog’ wheel, allowing increments and decrements, ranging from very course to very fine changes (fig. 5). Furthermore, manual numerical input was integrated into the current parameter value, allowing a specific value to be set directly. As the wheel is selected and ‘dragged’ towards the right of its normal position, the value is increased, and as it is turned towards the left, the value is decreased. The further the wheel is turned from its normal position, the more the value is changed. The equation for the factor that the value is multiplied with is:

$$factor = 10^{\frac{jogpos}{10}}$$

where *jogpos* is the position of the jog wheel ranging from the centre to its maximum towards the right: 0, 0.25, 0.5, 0.75, 1; and ranging from the centre to its maximum towards the left: 0, -0.25, -0.5, -0.75, -1.

This corresponds to a value increase of approximately 6%, 12%, 19%, 26%, and a decrease of approximately 6%, 11%, 16%, 21% respectively. Because of this scaling, after any number of varied increases and decreases using the control, the value that was started with can easily be reached again using the jog wheel only. The result is shown using a chromatogram view (in the centre of the screen), following each parameter change in real time. Key numerical results at the bottom of the screen are updated in real time as well.

<Figure 5>

5. Evaluation

North suggested that traditional evaluation methods, using detailed controlled tasks, potentially inhibit the generation (or at least recording) of insights [9]. In contrast, an open-ended protocol, resulting in a qualitative analyses rather than a quantitative, allows greater domain relevance and therefore a higher likelihood of inducing and recording the occurrence of insight.

Insight evaluation was originally proposed to deal with shortcomings in the traditional empirical method. Unlike traditional evaluation protocols, both formative and summative [22], insight evaluation does not rely on a set of predefined tasks. Instead users (generally domain experts) are left reasonably free to explore the system, relying on their own knowledge and interests to drive their activity. Insight evaluation has been applied successfully in a number of contexts [18,19,20].
We felt that insight evaluation was ideally suited for measuring the effect of the added interactivity and visualization in the user interface.

5.1. Experimental design and procedure

Five subjects took part in the experiment. All were experts in the CCC field, rather than usability experts. The goal was to determine the potential for insight generation, rather than usability per se, and thus required users with strong domain knowledge. The use of a user-centered, evolutionary design process meant that major usability issues had already been addressed by this stage. A ‘think-aloud’ protocol was used, where subjects were invited to provide a narrative on their experience and, in particular, mention insights as they occurred. As insights occurred, subjects paused their activity to allow the experimenter to record the details. The format for the recording sheet is shown in Table 1. Note that data collection was carried out by the primary author. Whilst this may have introduced some risk of experimenter bias, this was minimized by instructing subjects to explore the model according to their own interests and all reported insights were recorded and included in the analysis.

<Table 1>

The procedure for recording insights was as follows. First the visual element (e.g. Chromatogram, Timeline, ICE) of the interface where the insight occurred was recorded. Then a short description of the insight was recorded. The insight was then categorized into three categories. The first category was: different from expectation (‘Dif’), where a (model) behavior different from the subject’s expectation was observed. The second category was: a new discovery (‘New’). The third category: the creation of a new hypothesis (‘Idea’) meant (model) behavior was observed, resulting in the subject to form a new hypothesis that could subsequently be investigated. Then domain value, or importance of the insight was marked in a range from 1 (low) to 5 (high). The same scale was used for the insight complexity or depth. A rough indication of the time it took the subject from using a particular element to having the insight was marked as short, medium or long. And finally a note was made in case an insight was actually erroneous, or incorrectly interpreted. The subjects were also recorded, using a webcam, recording sound in case necessary to later on verify any comments the subject made, and the subjects’ face in case any additional information was provided by the facial expression indicating an insight or unexpected behavior of the interface.

To quantify the contribution of the visual enhancements, a basic version of the CCC computer model that included just the non-interactive Chromatogram, was used as a control. An open task scenario was used which allowed subjects to ‘explore’ the model, rather than simply completing set tasks. Two simple data sets of input parameters were initially loaded into the computer model. The working of the computer model, its interface, its features and the loaded parameters were then explained to the subject. The subject was then encouraged to freely use the computer application and explore the behavior of the model while changing parameters.

Although subjects were free to move to any of the elements of the interface, the general sequence of use was as follows. First a simple data set would be loaded with some basic model parameters displaying a plain form with numerical input values. These could then be modified, and using the basic version, subjects could then proceed to run the model with set parameters to obtain the model result. In the enhanced version, subjects would first move from the parameter input to the ICE interface. The subject could then dynamically manipulate the interface and see the preview results as described in previous sections, and then continue to run the model to obtain final model results. However in the enhanced version, subjects would likely spend more time using the ICE interface.

Using the think-aloud method, the subject indicated when gaining an insight (as previously defined). At that time, details about the insights were recorded by the experimenter. The basic version of the CCC computer model, the control condition, was evaluated first. This version of the model did not include any of the visual enhancements described earlier, but only had a simple user interface with a basic chromatogram output showing units in model steps on the horizontal axis. After the subject satisfactory finished exploring the two data sets, the subject was asked to switch to the enhanced version of the model. In this way only new insights gained by the enhanced version were recorded, excluding any insight already gained using the basic version.

6. Results and discussion

Tables 2, 3 and 4 summarize the main results of the study. Table 2 shows the total insights accrued by each subject using both the basic and enhanced interfaces. It can be seen that the proposed visualizations resulted in a relative increase in insights of 6.5 times over the basic chromatogram view.

<Table 2>

Table 3 goes into more detail, showing the number of insights accrued for each visual element of the enhanced interface. Whilst the time-mode view accounted for some of this increase, it can be seen that the ICE mode accounted for the majority (77%) of all significant insights achieved, with the time-mode view accounting for 15% and the basic chromatogram accounting for the remaining 8%.

<Table 3>

Table 4 describes the most important insights (as defined by domain value) and the visual element with which they were achieved. The ICE mode generated five out of total six most important insights. It is apparent that most of these insights relate to what effect the input parameters have on the separation results.

<Table 4>

Each subject showed unique insights, as well as insights that subjects had in common. The most common insight was that particular input parameters critically influenced the quality of the result of the model. Furthermore, the most important insights (with the highest domain value) occurred using the ICE mode. A relationship between the time before the insight occurred, and the complexity of the insight was also seen. In general the more complex insights were recorded after a longer time. Subject 1 had no insights using the basic version of the model, and five using the enhanced version (all in the ICE mode). Three of these triggered the formation of a new hypothesis, which the subject subsequently tested. Most insights were of high domain value. Subject 2 had one insight using the basic version, of a low complexity and domain value, and one insight using the enhanced version (in the Chromatogram mode), also of low domain value. Subject 3 had one insight of high domain value using the enhanced version (in the ICE mode). Subject 4 had one insight using the enhanced version (in the Time mode) with a high domain value. Subject 5 had one insight using the basic version of a low complexity, and five insights with high domain value using the enhanced version (four in the ICE mode and one in the Time mode).

The insight classification and scaled values (complexity and domain value) proved adequate for this study, and provided good detail and relevance. Although the general insight evaluation method was followed, subjects still seemed reluctant to exercise their freedom. Subjects possibly had an objective oriented mindset and/or had set a time limit in the context of the study. Another problem that was encountered was that some subjects tended to focus on the interface and its usability rather than the model itself, in spite of efforts made to encourage focusing on the behavior of the model. This was a problem with subject 3 in particular. Overall, the number of recorded insights, whilst not as high as initially expected, was still sufficient to identify clear differences in the efficacy of each visual element.

7. Conclusions and future work

This paper has shown how model simplification, by replacing iterative steps with unitary predictive equations, can enable dynamic interaction with a complex simulation process. Model previews extend the concept of dynamic querying and query previews to the context of ad hoc simulation model exploration. A case study was 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. In other words, dynamic model previews of simulations that give an indication of the effect of parameter changes are clearly advantageous to users who are seeking to learn from or explore the behavior of a simulation model. The number of insights gained using the enhanced interactive version of the computer model was some 6.5 times higher than the number of insights gained using the basic version of the model. There is a clear contribution of the visual elements and interactivity that were successfully implemented in the computer model. The interactive configuration explorer (ICE) generated the majority of the insights recorded when using the enhanced interface.

Insight evaluation is a relatively new approach to evaluation. Our experience with this protocol suggests that in order to achieve best results, it is important to impress upon the user that the study is about exploring the underlying domain (CCC in this case), rather than evaluating usability. Additionally, it is likely that adopting a longitudinal study design would produce richer and more reliable results. This would allow users to fully familiarize themselves with the interface and to incorporate the interface into their natural work activity. Previous research [19] has shown that non-supervised recording of insights over a period of time provides subjects with significantly more freedom to achieve relevant and significant insights. In future work we plan to distribute the model to domain experts, inviting them to record their insights using a similar template table as was used in this study. As users become more familiar with the user interface, they should become less focused on interface usability and more focused on using the software as a research tool.

Nevertheless, the results of this study are still encouraging, demonstrating that even a small-scale application of the insight evaluation methodology can provide a valuable means of measuring the value added by the introduction of dynamic query tools.

References

[1] B. Shneiderman, dynamic queries for visual information seeking, IEEE Software 11 (1994) 70-77.

[2] J. Seo, Information visualization design for multidimensional data: integrating the rank-by-feature framework with hierarchical clustering, in Department of Computer Science, University of Maryland, College Park, 2005, 210.

[3] J. Seo and B. Shneiderman, A rank-by-feature framework for interactive exploration of multidimensional data, Inf. Vis. 4 (2005) 99-113.

[4] B. Spence, L. Tweedie, H. Dawkes and H. Su, Visualisation for functional design, Atlanta, GA, USA, IEEE Conf. Inf. Vis. (1995) 4-10.

[5] R. Bosch, C. Stolte, G. Stoll, M. Rosenblum and P. Hanrahan, Performance analysis and visualization of parallel systems using SimOS and Rivet: a case study, 6th International Symposium on HPCA, Toulouse, France, IEEE (2000) 360-371.

[6] C. Plaisant, B. Shneiderman, K. Doan and T. Bruns, Interface and data architecture for query preview in networked information systems, ACM T. Inform. Syst. 17 (1999) 320-341.

[7] I.A. Sutherland, J. de Folter and P. Wood, Modelling CCC using an eluting countercurrent distribution model, J. Liq. Chromatogr. Relat. Technol. 26 (2003) 1449-1474.

[8] J. de Folter and I.A. Sutherland, Universal counter-current chromatography modelling based on counter-current distribution, J. Chromatogr. A 1216 (2009) 4218-4224.

[9] C. North, Toward measuring visualization insight, IEEE Comput. Graph. Appl. 26 (2006) 6-9.

[10] S. Hartmann, The World as a Process: Simulations in the Natural and Social Sciences, in: R. Hegselmann, U. Mueller, K.G. Troitzsch (Eds.), Simulation and Modelling in the Social Sciences from the Philosophy of Science Point of View, Kluwer Academic Publishers, Dordrecht, 1996, pp. 77-100.

[11] Y. Ito, M. Weinstein, I. Aoki, R. Harada, E. Kimura and K. Nunogaki, The coil planet centrifuge, Nature 212 (1966) 985-987.

[12] L. Tweedie, B. Spence, H. Dawkes, H. Su, The Influence Explorer, CHI Conf. Proc., ACM, Denver, Colorado, USA (1995) 129-130.

[13] C. Ahlberg, C. Williamson and B. Shneiderman, dynamic queries for information exploration: an implementation and evaluation, CHI Conf. Proc., ACM, Monterey, California, USA, 1992, 619-626.

[14] A. Dix, G. Ellis, by chance - enhancing interaction with large data sets through statistical sampling, AVI Conf. Proc., ACM, Trento, Italy (2002) 167-176.

[15] L. Tweedie, Exploiting interactivity in graphical problem solving: From visual cues to insight. Unpublished Doctoral, London University, London (1997).

[16] H. Su, J. Nelder, P. Wolbert, R. Spence, Application of generalized linear models to the design improvement of an engineering artifact. Qual. Reliab. Eng. Int. 12 (1996) 101-112.

[17] S.A. Herrod, Using Complete Machine Simulation To Understand Computer System Behavior, in Department of Computer Science, Stanford University, Stanford, 1998, 121.

[18] P. Saraiya, C. North and K. Duca, An evaluation of microarray visualization tools for biological insight, IEEE Symposium on Inf. Vis., Austin, TX, USA, (2004) 1-8.

[19] P. Saraiya, C. North, V. Lam and K.A. Duca, An insight-based longitudinal study of visual analytics, IEEE Trans. Vis. Comput. Graph. 12 (2006) 1511-1522.

[20] C. Plaisant, J.D. Fekete and G. Grinstein, Promoting insight-based evaluation of visualizations: From contest to benchmark repository, IEEE Trans. Vis. Comput. 14 (2008) 120-134.

[21] A. Berthod, M.J. Ruiz-Angel and S. Carda-Broch, Elution-Extrusion Countercurrent Chromatography. Use of the Liquid Nature of the Stationary Phase To Extend the Hydrophobicity Window, Anal. Chem. 75 (2003) 5886-5894.

[22] M. Scriven, Beyond formative and summative evaluation, in: D.C.P. M. W. Mclaughlin (Ed.), Evaluation and education : at a quarter century, University of Chicago Press, Chicago, Illinois, USA (1991) 19-64.

Figure 1. Illustration of the liquid-liquid chromatography process [8]. The system alternates between two stages. First there is a transfer step (T), where the upper (mobile) phase is advanced one step. Then there is a mixing & settling step (D) where the sample components redistribute over the phases according to their affinity (K-value). K refers to the ratio of distribution between the two phases. The component shown as red squares (K=2) has a greater affinity to the upper phase, whilst the other component (K=0.5) has a greater affinity for the lower phase. As the red component prefers the upper phase, it moves through the tubes faster than the blue component. Hence the components become separated.

Figure 2. Annotated screenshot of model: Chromatogram, showing different sample components in different colors; X: X-axis (volume or time), Y: Y-axis (component concentration); P: Chromatography Peak values, UP: Upper Phase (top half); LP: Lower Phase (bottom half); COL: Chromatography system (Column)

Figure 3. Superimposed screenshots of model: Time mode chromatogram where different views are shown in different opacity in this figure, illustrating the view at different points in time of the simulation process (at different positions of the slider control).

Figure 4. Rendering optimization process

Figure 5. Screenshot of model: Real time interactive configuration explorer; with interactive controls (top), where far left control is being rotated using the mouse causing its value to increase; chromatogram view (centre); key numerical results (bottom).

Table 1 Template for recorded data for each insight

Visual element	Insight	Category			Domain value	Complexity / Depth	Time	Errors
		Dif	New	Idea			Short/Med/Long	
					1 2 3 4 5	1 2 3 4 5	S M L	

Table 2 Total number of insights for version and enhanced version of model

Subject	Basic	Enhanced
1		5
2	1	1
3		1
4		1
5	1	5
Total	**2**	**13**

Table 3 Insights for each visual element

Subject	ICE	Chromatogram	Time-mode
1	5		
2		1	
3	1		
4			1
5	4		1
Total	**10**	**1**	**2**

Table 4 Description of most important insights

Insight	Visual Element
Determine critical amount of phase distribution for good resolution	ICE
Resolution not directly dependant on retention	ICE
Number off system cells big impact on resolution; New hypothesis	ICE
Hypothesis: Real experiment must have very inefficient mixing	ICE
Flow rates critical effect on separation	ICE
Start of separation does most of the separation	Time-mode

Highlights

- Model simplification enables dynamic interaction for complex simulation process.
- Model previews extend dynamic querying and query previews in simulation model.
- A case study is presented within the domain of counter-current chromatography.
- The number of insights was 6 times higher using the enhanced computer model.

Figure1

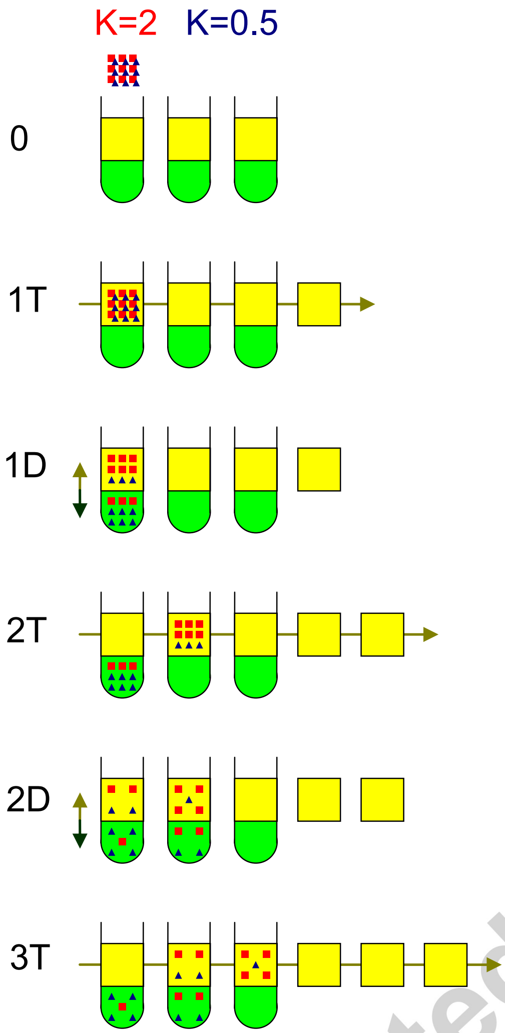
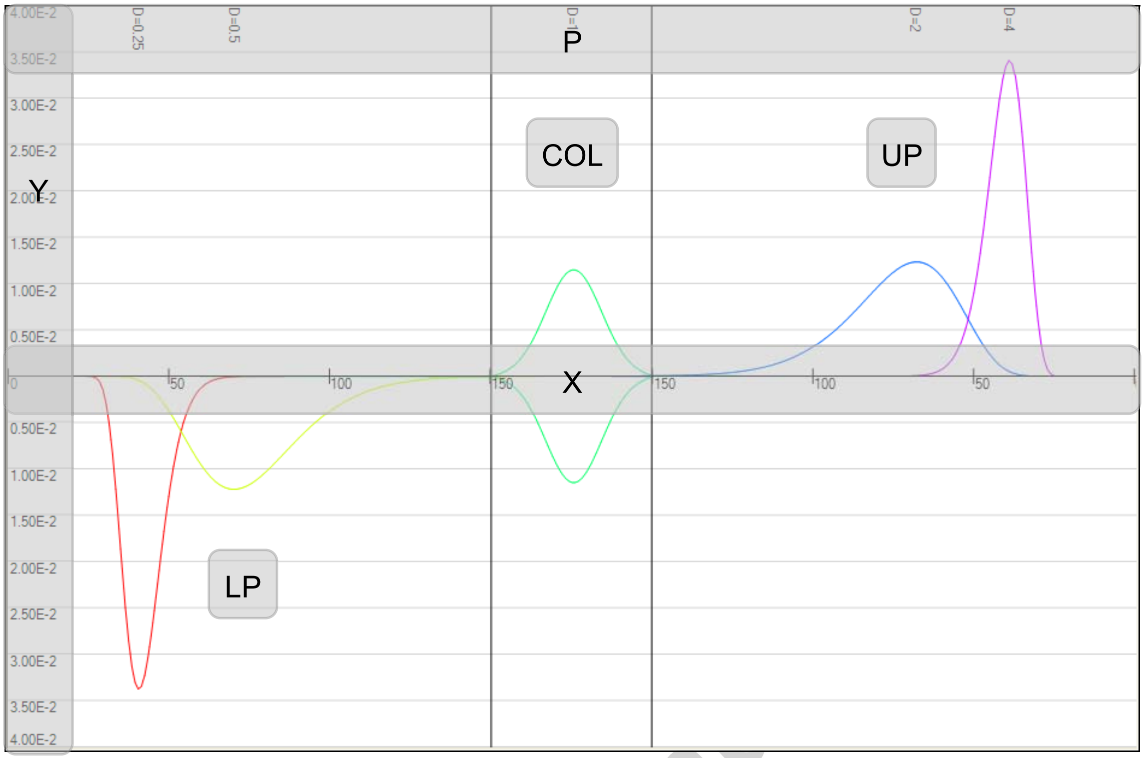


Figure2


4.00E-2
3.50E-2
3.00E-2
2.50E-2
2.00E-2
1.50E-2
1.00E-2
0.50E-2
0.50E-2
1.00E-2
1.50E-2
2.00E-2
2.50E-2
3.00E-2
3.50E-2
4.00E-2
Y
D=0.25
D=0.5
D=1
D=2
D=4
P
COL
UP
X
LP
0
50
100
150
150
100
50

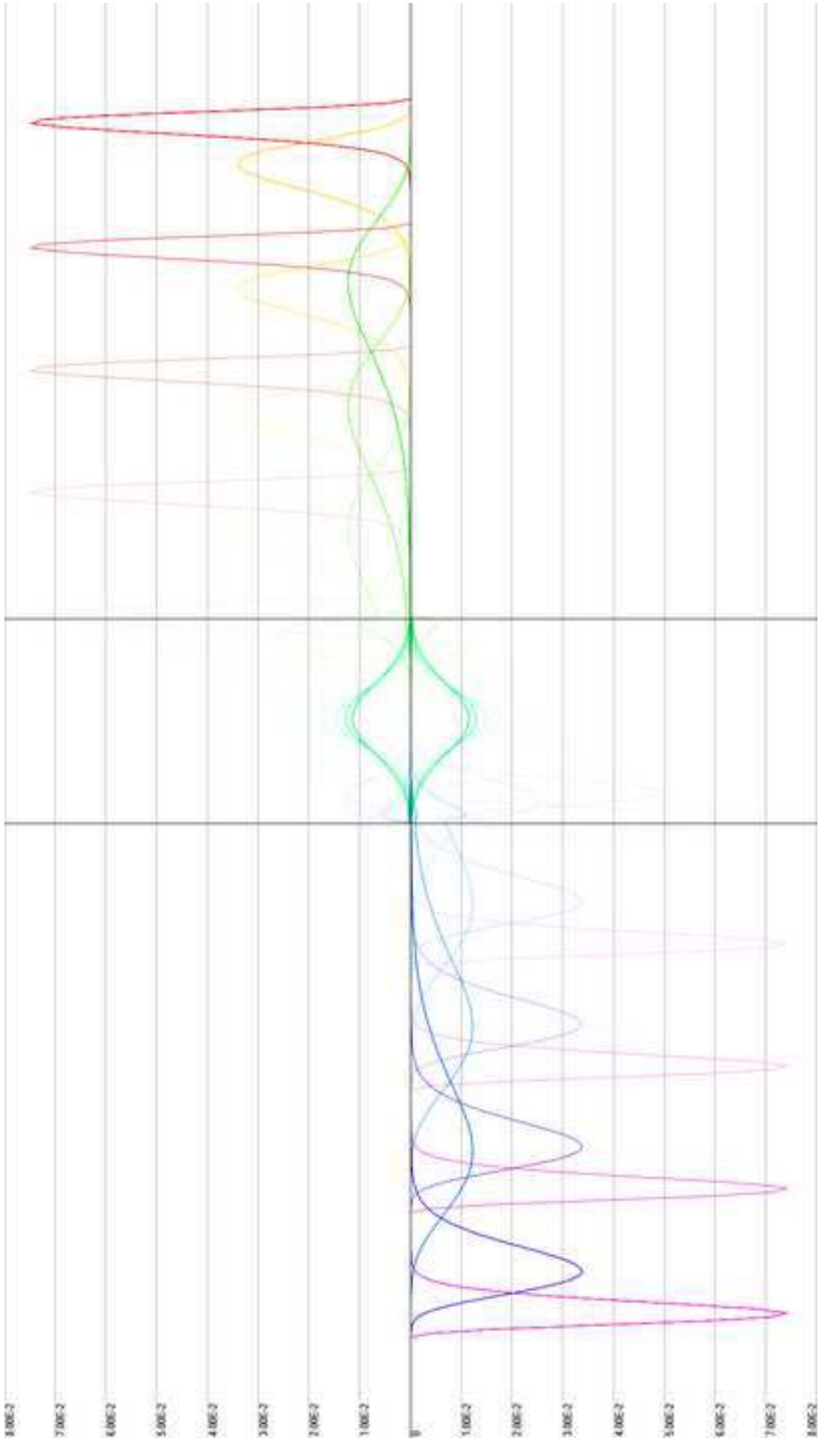

8.00E-2
7.00E-2
6.00E-2
5.00E-2
4.00E-2
3.00E-2
2.00E-2
1.00E-2
0
1.00E-2
2.00E-2
3.00E-2
4.00E-2
5.00E-2
6.00E-2
7.00E-2
8.00E-2

Figure3

Figure4

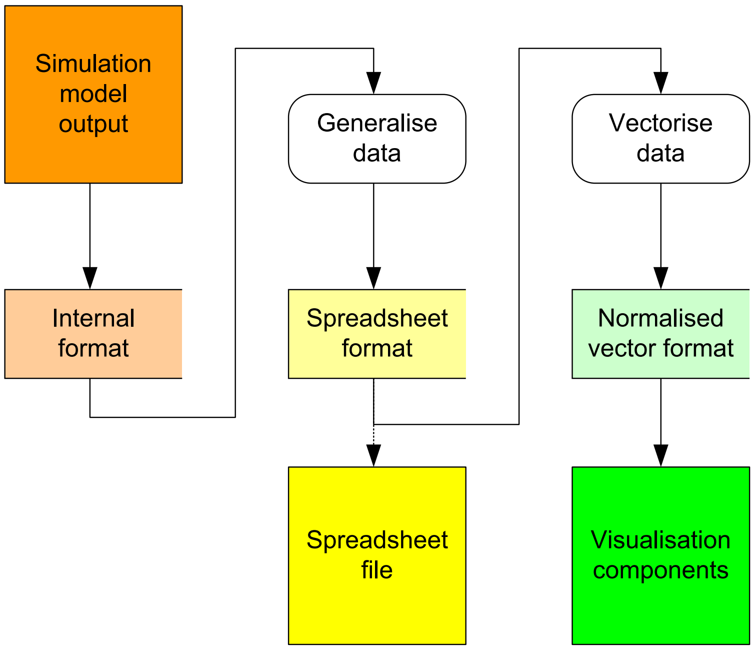
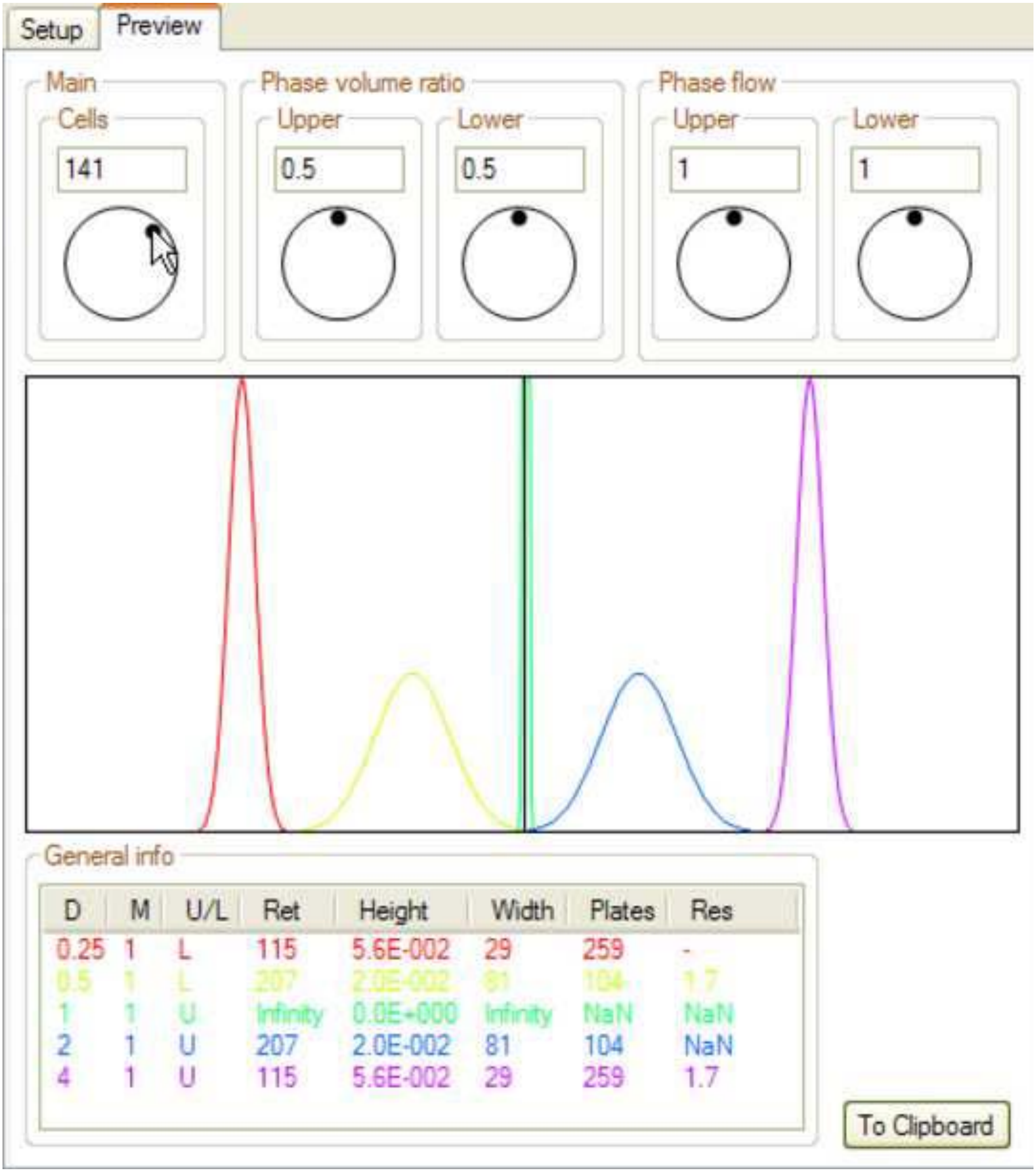

Simulation model output
Generalise data
Vectorise data
Internal format
Spreadsheet format
Normalised vector format
Spreadsheet file
Visualisation components

Figure5

Click here to download high resolution image



Main

Cells: 141

Phase volume ratio

Upper: 0.5

Lower: 0.5

Phase flow

Upper: 1

Lower: 1

General info

D	M	U/L	Ret	Height	Width	Plates	Res
0.25	1	L	115	5.6E-002	29	259	-
0.5	1	L	207	2.0E-002	81	104	1.7
1	1	U	Infinity	0.0E+000	Infinity	NaN	NaN
2	1	U	207	2.0E-002	81	104	NaN
4	1	U	115	5.6E-002	29	259	1.7