Two Design Patterns for Visualising the Parameter Space of Complex Systems

Nic Geard*
ACCS Technical Report

November 9, 2006

A key feature of complex systems is that their behaviour can vary significantly depending on their location in parameter space. A major challenge for researchers is to understand how combinations of system parameters influence behaviour; that is, to understand the shape of parameter space. Tools for visualising the structure and dynamics of complex systems and the shape of their parameter spaces play an important role in addressing this challenge. Many of these tools are developed to address problems in specific domains. If complex systems share certain general properties that transcend their specific domain, it should be possible to share tools for understanding these systems between domains. One technique that has been proposed for achieving this is the use of design patterns (Wiles and Watson, 2005).

Patterns are a tool that enables the collective knowledge of a particular community to be recorded and transmitted in an efficient manner. Initially developed in the field of architecture and later developed by software engineers (Gamma et al., 1995), they have now been adopted by the complex systems modelling community (Wiles and Watson, 2005). It can be argued that, while most complex systems models are idiosyncratic and highly specific to the task for which they are constructed, certain tools and methodologies may be abstracted to a level at which they are more generally applicable.

This report describes two patterns – the Recursive Parameter pattern (Section 1) and the Interactive Heatmap pattern (Section 2) – that provide techniques for managing the exploration and visualisation of large parameter spaces. These patterns are based on research reported in (Geard, 2006) and a visualisation tool that implements these patterns (in the context of exploring the parameter space of developmental genetic systems) is available from `http://www.itee.uq.edu.au/~nic/_linmap`.

* `nic@itee.uq.edu.au`

1 Recursive Parameter Pattern

1.1 Problem

Model systems typically contain one or more parameters whose value affects the behaviour of the system. For simple systems it may be possible to characterise the relationships between system parameters and behaviour analytically. Complex systems consisting of many nonlinear interactions between components can be less amenable to analytic approaches, and empirical simulation is often used to investigate the relationships between system parameters and behaviour.

A common solution to this problem is to define the range of interest of the Target Parameter $[p_{min}, p_{max}]$ and a step size δp. The parameter range is then sampled $(p_{max} - p_{min})/\delta p$ times at intervals of δp. There are two problems with this approach:

1. large homogeneous regions will be repeatedly sampled; and
2. regions in which system behaviour changes rapidly with the Target Parameter will be undersampled.

The parameter spaces of many systems are frequently very large. Searching such spaces as described above can be time consuming and resource intensive.

1.2 Context

The Recursive Parameter pattern can be applied when the following preconditions are satisfied:

- A parameterised model exists. Specifically, one parameter (or meta-parameter), the Target Parameter can be identified whose impact on the behaviour of the system is of interest.
- An Identity Function can be identified that uniquely quantifies the behaviour of the system. Note that this quantity need not necessarily be meaningful, or otherwise of interest; it must however, be monotonic. That is, the following conditions must hold:
    1. **if** $IF(a) = IF(b)$ **for** Target Parameter values a and b, **then** $IF(c) = IF(a)$ must hold for all values of c in the range $[a, b]$;
    2. **if** $IF(a) \neq IF(b)$ **for** some b incrementally greater than a, **then** no $c > b$ exists for which $IF(a) = IF(c)$;
    3. **if** $b > a$ **then** $IF(b) > IF(a)$.

1.3 Forces

The primary goals of the Recursive Parameter pattern are:

- understanding the influence of the Target Parameter on model behaviour;
- automatic location of regions of interest (*e.g.*, phase transitions);
- variable resolution of exploration: high detail in regions of interest; low detail elsewhere; and
- time and resource efficient exploration of system behaviour.

1.4 Solution

The Recursive Parameter pattern uses binary recursion to automatically focus exploration of system behaviour on interesting regions of parameter space:

$\textsc{RecursiveExplore}(p_a, p_b, \textit{depth})$:

> generate system behaviours $IF(a)$ and $IF(b)$
> **declare** $p_c = \frac{p_a + p_b}{2}$
> generate system behaviour $IF(c)$
> **if** $(IF(a) \neq IF(c))$ **and** $(\textit{depth} > 0)$:
> $\textsc{RecursiveExplore}(p_a, p_c, \textit{depth} - 1)$
> **if** $(IF(c) \neq IF(b))$ **and** $(\textit{depth} > 0)$:
> $\textsc{RecursiveExplore}(p_c, p_b, \textit{depth} - 1)$
> **else return**

Initially, the RecursiveExplore procedure is called with p_a equal to p_{min} and p_b equal to p_{max}. As the procedure is called recursively, this range is continually subdivided, with regions of equivalent system behaviour being ignored and regions of varying system behaviour being explored in greater detail. The *depth* parameter imposes a limit on the level of recursion. Increasing *depth* results in a map with greater resolution along the p axis, at the expense of increased processing time (Figure 1).

1.5 Examples

An example application of the Recursive Parameter pattern is described in Geard (2006). The model system in this instance is a network of interacting genes that controls the division and differentiation of cells in a developing organism. The behaviour of this system is represented as a cell

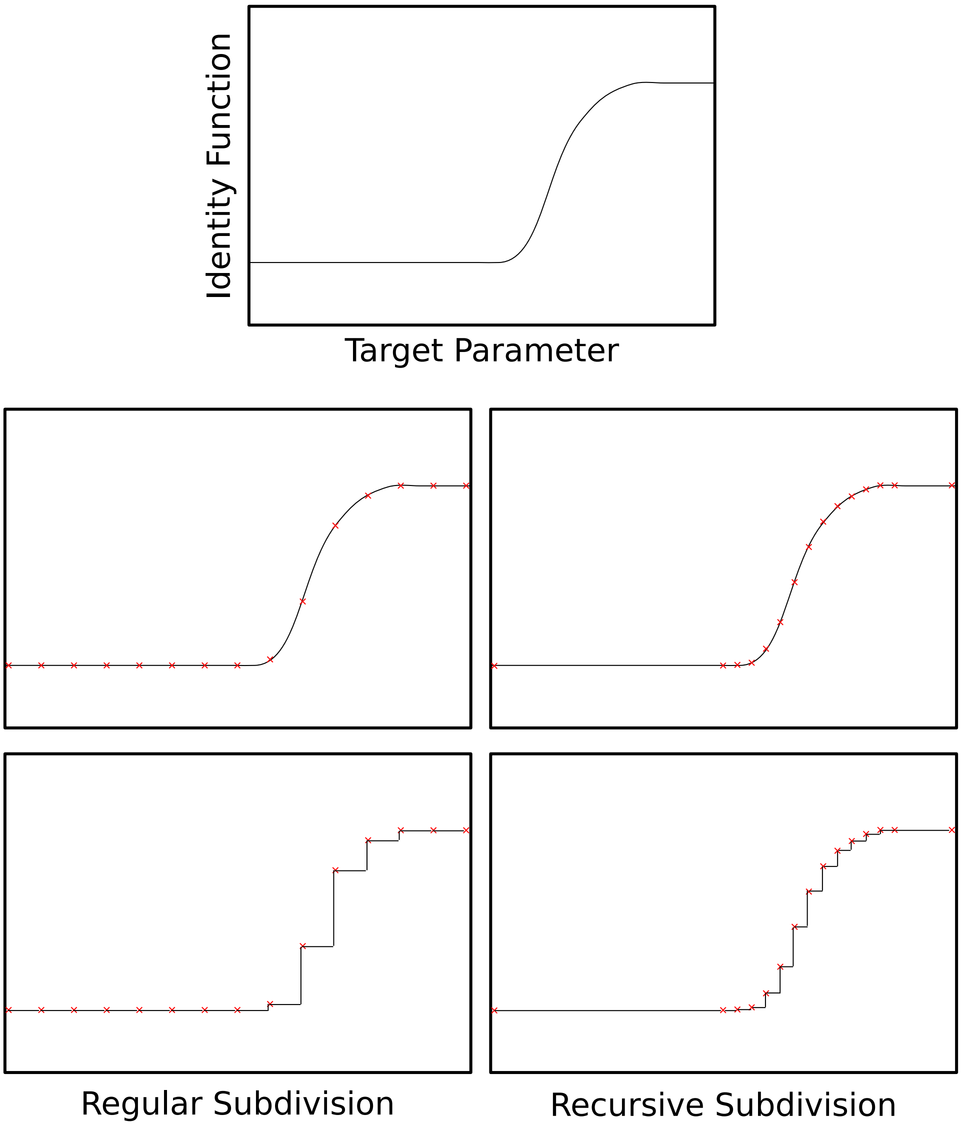

Figure 1: In the top plot, the phase transition in the Identity Function is clearly visible. The plots below demonstrate the reconstructed functions generated by regular and recursive subdivision and sampling of the Target Parameter. In both cases, 15 sample system behaviours have been generated. Recursive subdivision results in an increased resolution of sampling in the heterogeneous transitional region and a reduced level of sampling in the homogeneous regions.

lineage – a binary tree in which the root is the initial fertilised egg cell and the leaves are the differentiated terminal cells of the final organism.

This model system involves a large number of parameters. The gene network can be characterised by the number and connectivity of its genes, as well as their pattern and strength of interaction. The developmental process used to generate a cell lineage also contains parameters that control the rate at which the division threshold is scaled and the maximum depth of evaluation. One of these – the lineage scaling parameter, λ – satisfies the requirements necessary for allowing recursive exploration of parameter space.

At very low values of λ, a cell will always divide, generating a proliferating lineage in which cells never differentiate. At very high values of λ, the initial cell will never divide. In between these two extremes, a wide variety of different cell lineage structures are observed. These cell lineage structures are not distributed evenly along the λ parameter range however; large λ ranges map to a single lineage, while dense concentrations of different lineage structures are found within a small distance from certain 'transitional' values of λ.

By recursively subdividing the λ parameter, homogeneous regions are rapidly characterised as such and ignored, while more computational resources are allocated to generating cell lineages in the diverse transitional regions.

A further application of the Recursive Parameter pattern in this context is to provide an efficient graphical representation of how the complexity of cell lineages is distributed around a phase transition (Figure 2).

1.6 Resulting Context

The primary outcome of applying the Recursive Parameter pattern is a map of Target Parameter values to system behaviours. Each entry maps the lowest value of the Target Parameter at which a particular system behaviour was observed to that behaviour.

One side effect of the Recursive Parameter pattern is that the map may not be minimal, in that it may contain sequential entries that map to the same value. If this duplication is a problem, one straightforward solution is to perform a sequential sweep through the map and merge or remove sequential duplicate entries.

1.7 Rationale

The rationale for the Recursive Parameter pattern is that pertinent features of a parameter space (such as monotonicity) can be exploited to enable that space to be explored and characterised more efficiently.

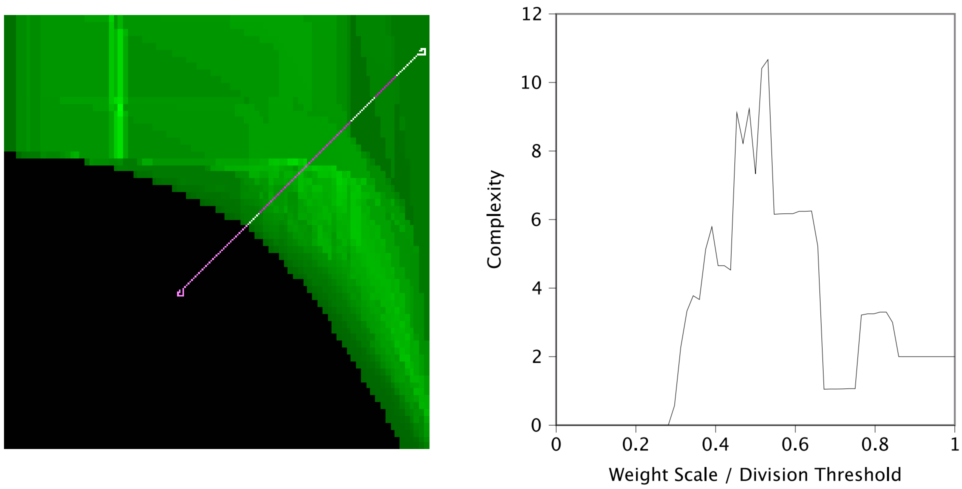

Figure 2: An example complexity profile showing the increase in complexity and diversity of cell lineages around a phase transition region. The heatmap on the left indicates the region from which the profile has been sampled. Section 2 below provides more information on the heatmap visualisation.

Many phenomena of interest in complex systems are transitional – that is, they occur around the phase transition from one state to another. Depending on other system parameters, the location of this phase transition may vary. The Recursive Parameter pattern enables transitional regions to be automatically detected and explored in greater detail than surrounding regions of lower interest.

1.8 Related Patterns

The Recursive Parameter pattern can be combined with the Interactive Heatmap pattern (Section 2) to enable more rapid generation of parameter ranges.

The Recursive Parameter pattern is conceptually related to dynamic exploration methods in several other domains. Response Surface Methodology (Box and Wilson, 1951, Box and Liu, 1999) uses iterative sampling of parameter spaces to identify optimal features of produce and process design. Recursive subdivision of topological surfaces is used in computer graphics to generate efficient approximations to arbitrary topological shapes (Catmull and Clark, 1978).

2 Interactive Heatmap Pattern

2.1 Problem

Many complex systems are capable of generating a wide range of possible behaviours depending on the value of one or more structural or dynamic parameters of the system. It is often desirable to understand both the range of a system's possible behaviours and how they are related in parameter space.

Several problems hamper such investigations:

1. There is often little or no *a priori* knowledge about where in parameter space interesting behaviours will be observed.

2. Different instances in a class of parameterised systems often display variable patterns of behaviour.

3. System behaviours can be highly complicated, making them difficult to quantify and compare in an automatic fashion. Often visual inspection is required to characterise and understand parameter space.

4. Individual instances of system behaviour can be computationally intensive to generate and/or visualise.

5. If a sufficient range of parameter combinations, class instances and behaviours are to be visualised and explored, the amount of data generated can be large and difficult to organise and inspect in an ordered fashion.

2.2 Context

The Interactive Heatmap pattern can be applied when the following preconditions are satisfied:

- A parameterised model exists. Typically the Interactive Heatmap pattern is applied when two independent Target Parameters of the model can be identified or defined.

- One or more Height Metrics can be defined that quantify some aspect of system behaviour.

- Optionally, system behaviour is also amenable to some form of additional visualisation (*e.g.*, network structure, expression pattern, morphology, *etc.*)

2.3 Forces

The primary goals of the Interactive Heatmap pattern are:

- Rapid and intuitive exploration of the parameter space of complex systems such that trends, gradients, transitions and other features can be easily identified.
- Intuitive visualisation of parameter space, as characterised by gradients in the defined Height Metrics.
- Rapid generation of new maps.
- Individual system behaviours can be easily mapped to their location in parameter space and neighbouring behaviours.
- Ability to 'drill down' into interesting areas of particular maps.

The primary constraints that can potentially operate are:

- Parameter spaces can be very large.
- Behaviour evaluation can be computationally intensive.

The aim of the Interactive Heatmap pattern is therefore to allow structured exploration of a parameter space in a way that is both intuitive and computationally efficient.

2.4 Solution

The core feature of the Interactive Heatmap pattern is the use of interactive visualisation to allow both system behaviours and parameter spaces to be viewed and explored in a structured fashion.

2.4.1 Basic system

The Interactive Heatmap pattern consists of two components. The first component is the heatmap itself. A heatmap is a two-dimensional representation of an abstract space in which each axis maps to one of the two Target Parameters. In addition, the value calculated by the Height Metric at a given point is indicated by a colour gradient. The second component is a graphic representation of the system behaviour at a given point in parameter space. The screenshot in Figure 3 illustrates how the Interactive Heatmap pattern may be implemented.

By selecting different points in the heatmap component, the corresponding system behaviour can be displayed, enabling an intuitive understanding of the relationship between a system's parameters and its behaviour to be generated.

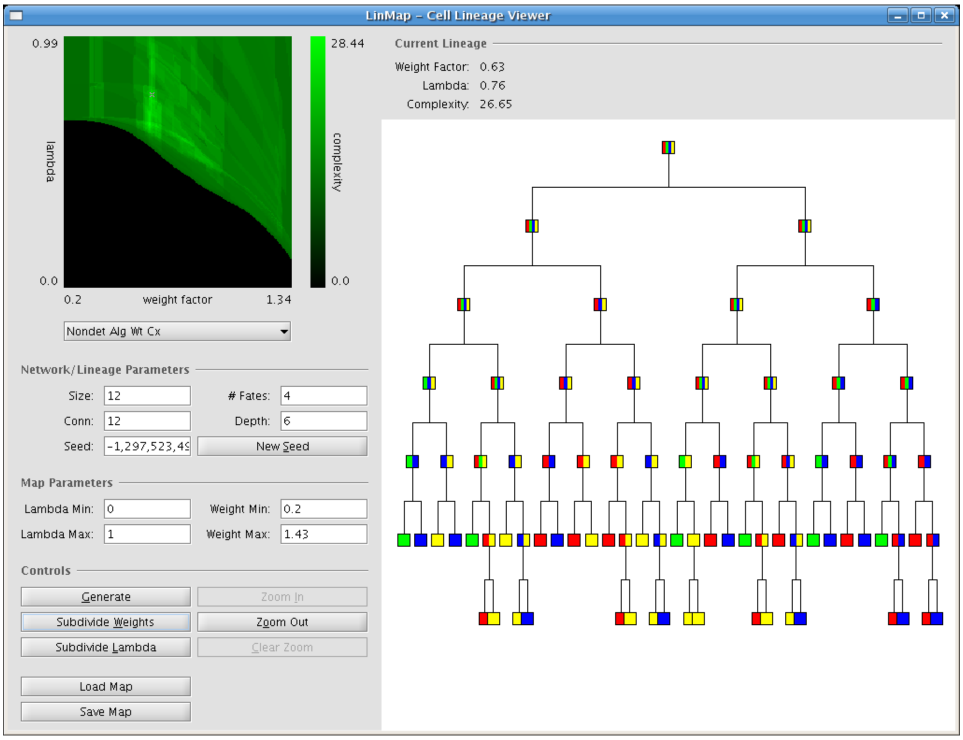

Figure 3: An example implementation of the Interactive Heatmap pattern. The heatmap component in the top left shows how the complexity of a developmental system varies with two underlying parameters. The cell lineage component illustrates the currently selected developmental system. Clicking at any point on the heatmap causes the corresponding cell lineage to be displayed. The LinMap visualisation tool also incorporates the Recursive Parameter pattern described above: while the Target Parameter mapped by the x-axis is sampled at uniform intervals, each individual column is sampled recursively along the y-axis. Therefore, the large black area (corresponding to a homogeneous region of proliferating lineages) is sampled very sparsely, while the more diverse regions in the centre of the heatmap are sampled much more densely.

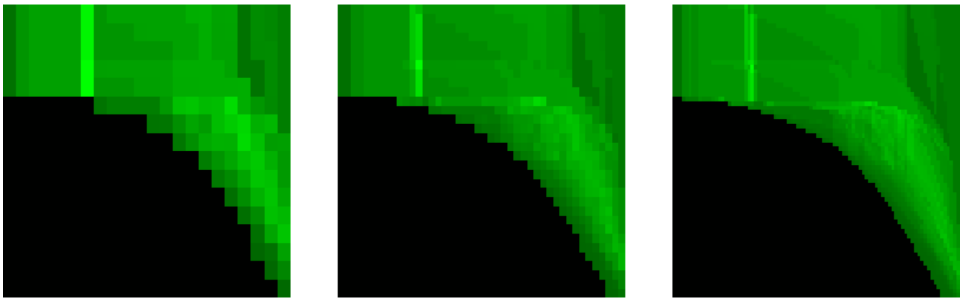

Figure 4: Three instances of the same heatmap at increasing levels of resolution. The initial heatmap (left) provides a rapid means of assessing whether this particular region of parameter space is interesting and worth exploring further. Resolution may then be interactively increased (centre and right) by interpolating between existing Target Parameter values.

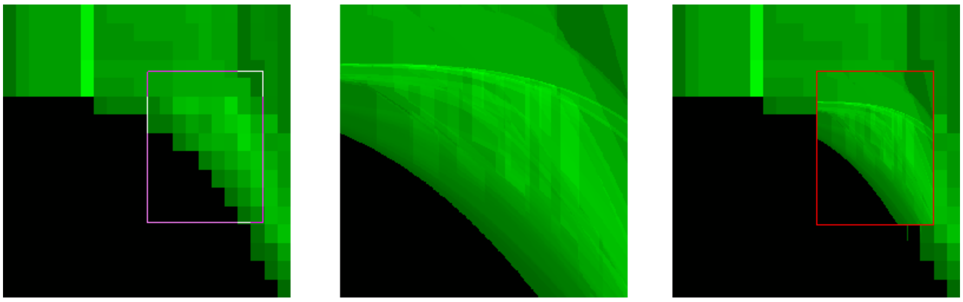

Figure 5: An example of increasing the resolution in a chosen region. First, a region of interest is identified and selected (left). The chosen region can then be zoomed in upon and sampled at a higher resolution (centre). Upon zooming back out to the original map, the zoomed region is highlighted and displayed at the increased resolution (right).

2.4.2 Extensions

The conflicting forces of speed and detail can be resolved in two different ways: by allowing the sampling resolution to be increased dynamically and by providing the ability to 'zoom in' on regions of a heatmap. Initial heatmaps can be generated quickly at low resolution (i.e., with a limited number of steps for each Target Parameter). This initial overview image should be sufficient to allow Target Parameter bounds to be adjusted and areas of possible interest to be identified. The 'resolution' of the heatmap can be increased by interpolating between the current Target Parameter steps (Figure 4). Alternatively, a sub-area of the heatmap can be selected and zoomed in upon to provide more detail in a chosen region (Figure 5).

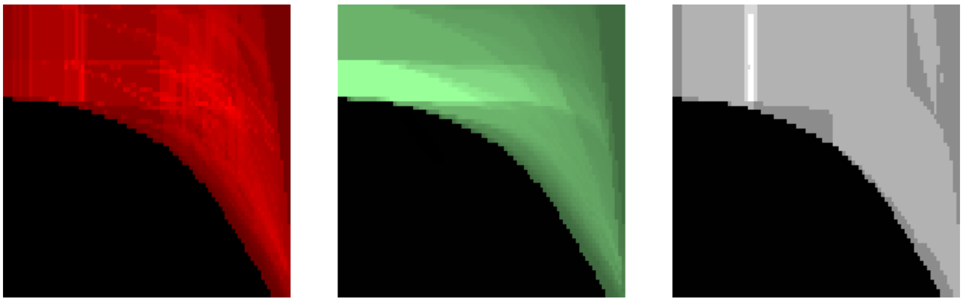

Figure 6: Three different Height Metrics applied to the same sample set shown in Figures 4 and 5. Details of the specific metrics used can be found in Geard (2006); however, the important observation is that different metrics result in different gradients across the parameter ranges.

2.5 Examples

An example application of the Interactive Heatmap pattern is described by Geard (2006) (Figure 3). The model system in this instance is a network of interacting genes that controls the division and differentiation of cells in a developing organism. The behaviour of this system is represented as a cell lineage – a binary tree in which the root is the initial fertilised egg cell and the leaves are the differentiated terminal cells of the final organism.

This model systems involves a large number of parameters: the size and connectivity of the gene network are parameters, as are each of the weights that modulate the interactions between genes; furthermore, the developmental mapping embodies parameters that determine the conditions under which division and differentiation take place. In this example, the two Target Parameters chosen were a scaling parameter applied to the strength of network interactions and a scaling parameter applied to the division threshold during development.

The Height Metric is defined in terms of cell lineage complexity, which can be quantified in several different ways ranging from the size of the lineage through to the number of recursive rules needed to describe the lineage. The use of multiple Height Metrics enables a further analysis of the way in which different aspects of system behaviour (in this case different definitions of complexity) vary across a common sample set (Figure 6).

2.6 Resulting Context

The primary outcome of applying the Interactive Heatmap pattern is a combined global and local view across parameterised slices of a complex system's parameter space.

Some experimentation may be required in order to identify Target Parameters and Height Metrics that (a) capture interesting slices of the total space; and (b) characterise gradients across these slices in a useful fashion.

2.7 Rationale

The primary rationale for the Interactive Heatmap pattern is that providing a dynamic interface to information makes exploration easier and more intuitive:

> "The best visualizations are not static images ... but fluid, dynamic artifacts that respond to the need for a different view or for more detailed information." (Ware, 2000)

By enabling a large number of systems to be explored very rapidly, the time taken to develop insights into the range of behaviours of a system is reduced. Providing an intuitive representation of the relationship between different system behaviours enables an enhanced understanding of how a parameter space is structured.

2.8 Related Patterns

The Interactive Heatmap pattern can potentially be combined with the Recursive Parameter pattern described above (Section 1) to provide more efficient exploration of heatmaps. Note that this is subject to at least one of the Target Parameters satisfying the requirements described in the Recursive Parameter pattern. Also, the Height Metric need not be identical to the Identity Function used in the Recursive Parameter pattern.

It is possible that the Height Metric used in the Interactive Heatmap pattern may rely on other patterns, such as the Perturbation Analysis pattern (Geard et al., 2005), to quantify system behaviours.

Visually, the application of the Interactive Heatmap pattern results in diagrams that bear a resemblance to the diagrams used to visualise and analyse patterns of gene coexpression from microarray data Eisen et al. (1998). Gehlenborg et al. (2005) have incorporated interactive elements into a visualisation framework for microarray data that enables the inclusion of additional meta-information to the basic representation. A related framework developed by Seo and Shneiderman (2005) provides an interactive means of exploring which dimensions of a high-dimensional parameter space provide the most informative view of a complex data set. Finally, Plumlee and Ware (2006) have explored some of the cognitive implications of zooming versus multiple views in the exploration of large information spaces.

References

Box, G. E. P. and Liu, P. Y. T. (1999). Statistics as a catalys to learning by scientific method part I—An example. *Journal of Quality Technology*, 31(1):1–15.

Box, G. E. P. and Wilson, K. B. (1951). On the experimental attainment of optimum conditions. *Journal of the Royal Statistical Society B*, 13:1–45.

Catmull, E. and Clark, J. (1978). Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer-Aided Design*, 10(6):350–355.

Eisen, M. B., Spellman, P. T., Brown, P. O., and Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences of the USA*, 95:14863–14868.

Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995). *Design Patterns*. Addison Wesley, New York, NY.

Geard, N., Willadsen, K., and Wiles, J. (2005). Perturbation analysis: A complex systems pattern. In Abbass, H. A., Bossamaier, T., and Wiles, J., editors, *Recent Advances in Artificial Life*, volume 3 of *Advances in Natural Computation*, pages 69–84, Singapore. World Scientific Publishing.

Geard, N. L. (2006). *Artificial Ontogenies: A Computational Model of the Control and Evolution of Development*. PhD thesis, School of Information Technology and Electrical Engineering, The University of Queensland.

Gehlenborg, N., Dietzsch, J., and Nieselt, K. (2005). A framework for visualization of microarray data and integrated meta information. *Information Visualization*, 4:164–175.

Plumlee, M. D. and Ware, C. (2006). Zooming versus multiple window interfaces: cognitive costs of visual comparisons. *ACM Transactions on Computer-Human Interaction*, 13(2):179–209.

Seo, J. and Shneiderman, B. (2005). A rank-by-feature framework for interactive exploration of multidimensional data. *Information Visualization*, 4:96–113.

Ware, C. (2000). *Information Visualization: Perception for Design*. Morgan Kaufman, San Francisco, CA.

Wiles, J. and Watson, J. (2005). Patterns in complex system modeling. In Gallagher, M., Hogan, J., and Maire, F., editors, *Intelligent Data Engineering and Automated Learning – IDEAL 2005: 6th International Conference*, Berlin. Springer-Verlag.