



## Dynamic Exploration

### 5.1 Real problems

When someone has a problem to solve it may not be clearly formulated, either consciously or unconsciously. The house-hunter<sup>1</sup> may wish to 'try to keep the cost below about £100,000, try for 3 bedrooms (but 4 would be nice), see if we can find something near a good school and, perhaps, close to Granny who might baby-sit for us'. As emphasized by the italics, the word 'precision' does not leap to mind. Nevertheless, a great many problems are often just as imprecisely articulated at first, and for good reason.

That reason is a lack of knowledge. The house-hunter does not know what houses are currently on the market, is possibly a stranger to the locality where a house is needed, and may have no idea whatsoever of the general level of prices. The first task, therefore, is to help the house-hunter to gain an understanding, not only of the relevant data, but additionally of any 'hidden' relationships such as a steep increase in price as one nears **Pearson's Park**. The next task is to empower them to solve their problem, that of finding a satisfactory house. Often, in the course of acquiring knowledge about available houses, the requirements may be modified ('looks like we could afford a 5-bedroom house if we

---

<sup>1</sup> Ideally, we should use the term 'home-hunting' to describe the process of looking for a new house, since it is the environment (a nearby park, graffiti on the walls, proximity to a good school), as well as the physical house, that will influence the selection of a particular house to buy. For simplicity we use the term 'house-hunting' throughout.

take in paying guests'). Thus, in practice, problem formulation is often equally as important as problem solution, and the two can usefully proceed together using, as they do, the same data. What is especially valuable to the person with a problem to solve is a means of visualizing the relevant data in such a way as to support these two activities, as well as a smooth transition between them.

To identify issues, and thereby help to establish guidelines and criteria for the application of visualization techniques, we first examine – perhaps paradoxically – a dated, textual command-based query system.

## 5.2 Command line query

With a conventional database query language the house-hunter might be expected to enter a query of the form:

```
Select house-address
From mydatabase
Where price <= 100,000 and
      Bathrooms = 2 and
      Bedrooms >= 3
```

to indicate that the most that can be afforded is £100,000, that two bathrooms are essential (no more, no less) and that the house should have at least three bedrooms. The result, of course, could be:

*0 HITS*

indicating that the estate agent can offer no houses satisfying these requirements. Equally unhelpful can be the response:

*1328 HITS*

especially if there is no indication as to how the query might usefully be modified. How much more or less should I offer?

There are, in fact, many disadvantages associated with conventional database query languages, and it is useful to identify them. Here are seven:

1. The discretionary user must learn a language. Except for specialist users, people are not prepared to do this. Even for query languages claimed to be simple (for example, Query by Example), controlled tests have shown that even after about an hour's tutelage insufficient familiarity was acquired (Borgman, 1986).<sup>2</sup>

---

<sup>2</sup> Borgman (1986) found that after a day's instruction only a quarter of the Stanford University students she tested could use the library's **online** query system. Additionally she found that users tend to make very simple queries and not use the full complexity of **Boolean** expressions.

2. Errors are not tolerated. Users are now becoming familiar with systems which do not permit syntactical errors, and would expect such a feature in any new tool.
3. Too few or too many 'hits' may occur. This is not surprising if the user has little or no knowledge of the database.
4. If there are too few or too many hits, there is no indication of how the query might beneficially be reformulated. By paying an extra £50 the 'house of my dreams' might be available: I would certainly want to be made aware of this.
5. There is a significant time delay between the formulation of a new query and the delivery of its result. There are two principal reasons. First, the user must formulate a new query in their mind. Second, they must enter it without error. A third possible reason may be an inherent delay in processing the entered query. Any delay discourages a 'let's see if . . .' approach which can efficiently lead to the acquisition of insight into the data being explored.
6. Useful contextual data is unnecessarily hidden. The importance of context – of a global model within which a local solution is to be found – has already been emphasized (see item 4 above).
7. As a result of all these drawbacks, it can be difficult for the user to build an internal model or cognitive map of the data, which is what information visualization is all about.

Behind all these drawbacks is the very basis on which a typical command-line query language is grounded: an assumption that a user knows precisely what single question needs to be asked and, additionally, knows a great deal about the database. The rejection of this assumption is fundamental to the developments we now describe. We first concentrate on the need to formulate a problem, and hence the need to acquire, by visualization, some idea of the content and characteristics of the database. What will permeate this chapter is the effectiveness of exploration – and in particular dynamic exploration – in gaining some qualitative idea of the database content while moving towards a satisfactory solution.

### 5.3 Dynamic queries

Perhaps the most significant recent acknowledgement of the fact that a user formulates a problem concurrently with solving it was the concept of dynamic queries, first reported by Williamson and Shneiderman (1992). An effective illustration (Figure 5.1) is again provided by a lay person's search for a house. To the right of a map are scales with adjustable sliding limits, each associated with a particular attribute of a house: Price, number of Bedrooms and Journey time to work are typical examples. Adjustment of the upper **and/or** lower acceptable limits to any of these attributes causes the corresponding selection to be made from all houses in the database, and the result of that selection immediately displayed by dots on the map. A significant advantage is that responsive interaction, by which we mean that an effect occurs within less than about 0.1 seconds of its cause, not only allows answers to be obtained rapidly but, most importantly, supports

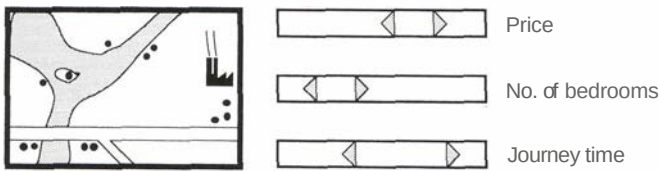


**FIGURE 5.1**  
The dynamic  
homefinder

dynamic exploration, often called the 'what if . . .?' activity. Even though no direct indication of how a query might usefully be modified is given, some indication can now be obtained manually by exploratory variation of the attribute limits: any advantage of paying £50 more is immediately indicated by the appearance of another dot on the map. The dynamic query technique, in fact, overcomes most of the seven drawbacks identified in the previous section.

Since the task to which the dynamic homefinder is addressed will be referred to again, it is useful to state it in generic terms:

Given a collection of objects, each described by the values associated with a set of attributes, **find** the most acceptable such object or, perhaps, a small number of candidate objects worthy of more detailed consideration.

### 5.3.1 To find a film

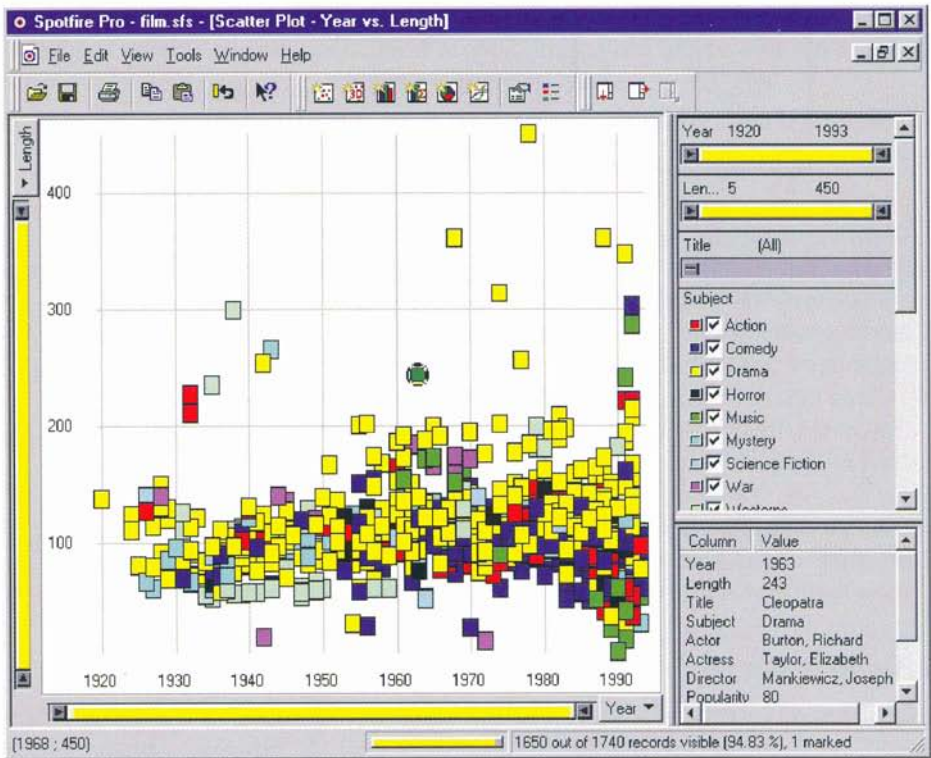
The same type of task is involved in the selection of a film to watch on video, for which the **Spotfire** display of Figure 5.2(a) is relevant (Ahlberg and Shneiderman, 1994). On the main display each coloured square identifies a film. Colour represents type (horror, musical, etc.), horizontal position indicates the year of production and vertical position indicates duration. On the right, sliders can be used to specify other attributes such as type, director, actor and other characteristics of a film. Scroll bars can be used to confine attention to a particular span of years and film length, allowing more detail to be displayed and explored (Figure 5.2(b)).

The visualization tool shown in Figure 5.2 also contains an effective way of compactly **making** a selection from a particular class of menu, called **alphaslider**s (Figure 5.3). The **alphaslider** technique, first proposed by Osada et al. (1993), and described in detail by Ahlberg (1996) allows users to scan rapidly through, and select items from, lists of alphabetic data. The essential components of the **alphaslider** shown in Figure 5.3 are a slide area, a **slider** thumb, a text output and, running underneath the **slider**, an index of the elements over which the **slider** operates. Guided by the user's view of the index and knowledge of the required initial letter, a mouse-click on the **slider** causes the thumb to move appropriately and the selected text item to appear. In Figure 5.3 the user, wishing to examine film titles beginning with 'H' has clicked the slide area just above the 'H' displayed in the index, whereupon the thumb has repositioned and the output 'Hunt for Red October' has appeared. If this title was not the one sought, gentle direct manipulation (for example, by mouse drag) of the thumb allows exploration. Not all the letters of the alphabet need appear in the index: a combination of the most popular and a reasonable spread could be best. An improved **alphaslider** shown in Figure 5.4 allows a choice of fine and coarse tuning.

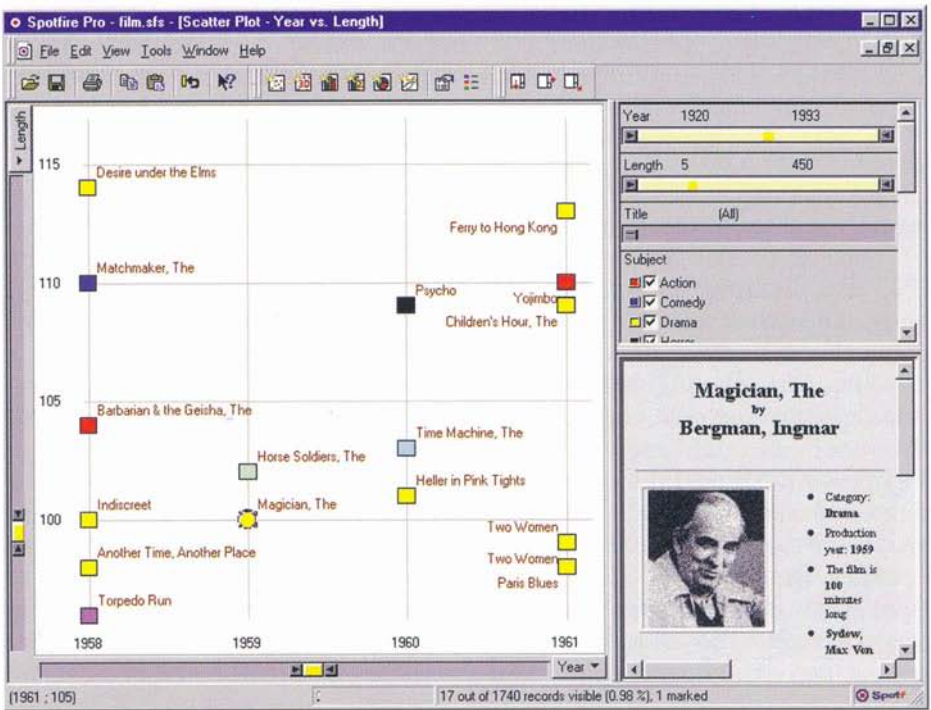
FIGURE 5.2

Dynamic queries applied to the selection of a film. Detail can be disclosed

Source: Spotfire AB



(a)



(b)

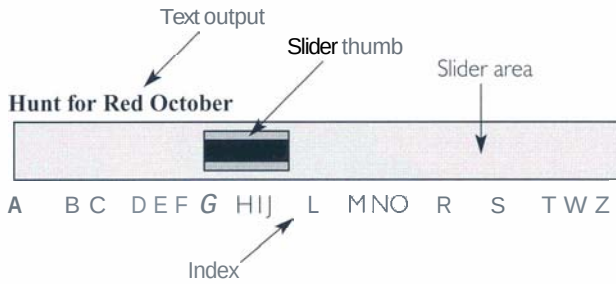


**FIGURE 5.3**  
An alph slider

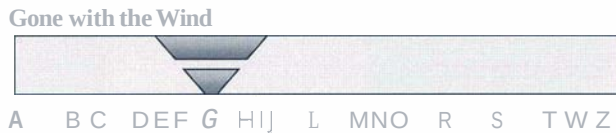


**FIGURE 5.4**  
An improved alph slider

### 5.3.2 Discovery

The dynamic queries concept has more general application than finding one or more desirable objects within a group. The illustration of Figure 5.5, for example, relates to demographic studies of Sweden, and in particular to the distribution of pollutants. Control panels on the right-hand side of the display allow the selection of the year involved (1975 and 1985 in the illustration) and,

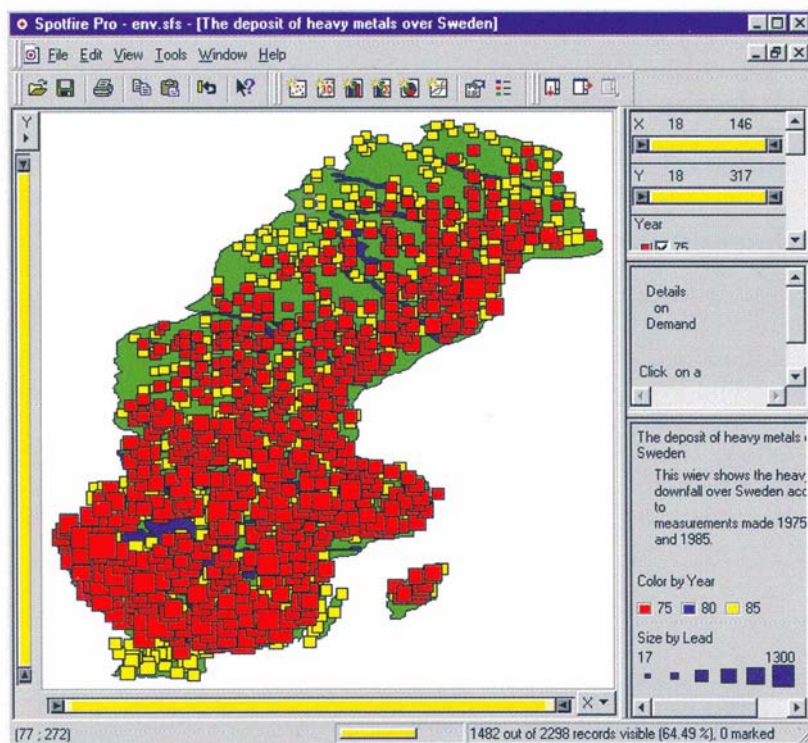
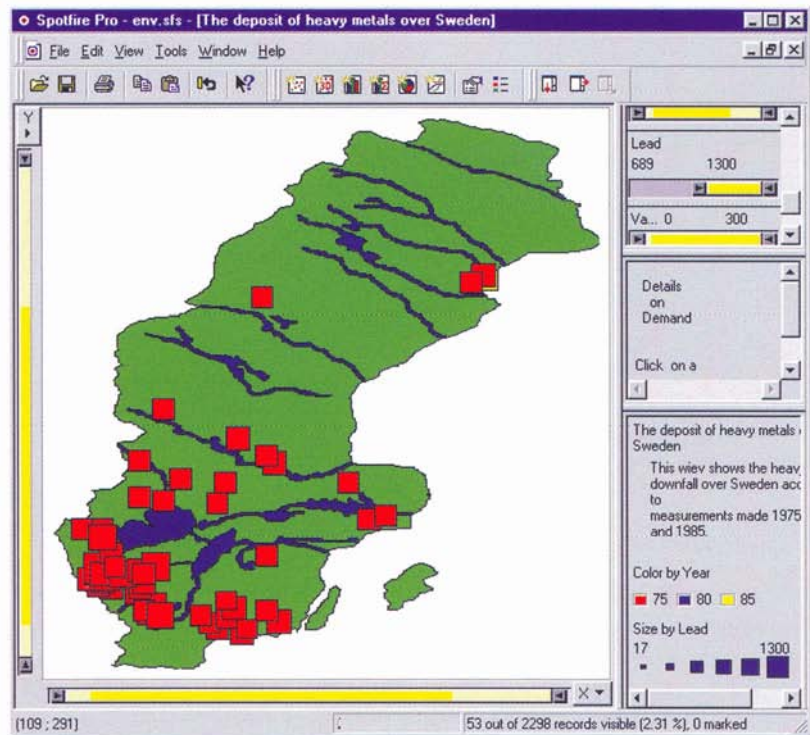


**FIGURE 5.5 (a)**  
A display of heavy metal deposits in Sweden

Source: Spotfire AB

(a)

FIGURE 5.5 (b)



(b)

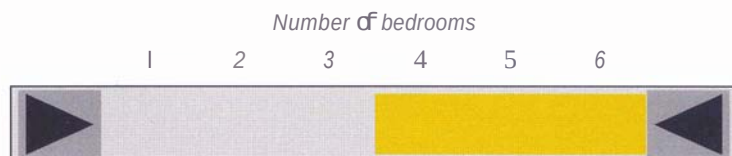
via the sliders, the thresholds above which a given heavy metal is to be displayed on the map. It is not difficult to appreciate that the technique of dynamic queries can support discovery and the formulation of hypotheses leading to a rearrangement of relevant data (Figure 5.5(b)).

### 5.3.3 Insensitivity

An interesting feature (Figure 5.6) can be associated with the sliders in Figure 5.5. Within each **slider**, a gray area indicates that range within which no objects are to be found as a result of other limits placed on other attributes. Thus, if a limit placed on Price is such that all 1-, 2- and 3-bedroom houses are thereby excluded, then this fact is registered on the Bedrooms scale, as shown in Figure 5.6. It follows that the selection is now insensitive to movement of the lower limit on Bedrooms as far up as 4 bedrooms. Such information can be of considerable value to the user.

FIGURE 5.6

An **alphaslider** feature indicating insensitivity to limit variation



## 5.4 The Attribute Explorer

A drawback to the dynamic queries concept as proposed by Williamson and Shneiderman (1992) is that data is disclosed only when it satisfies the query. In other words, only those objects are identified whose attribute values satisfy all limits. At first this sounds reasonable, especially in those cases where the display would otherwise be crowded. Nevertheless, as we shall now show, there are situations in which the display of all data provides **contextual information** yielding extremely valuable guidance for exploration.

To illustrate the value of contextual information we return to the house-hunting scenario where there is a (usually large) collection of objects, each described by a number of attributes, and where the task is that which was stated earlier:

Given a collection of objects, each described by the values associated with a set of attributes, find the most acceptable such object or, perhaps, a small number of candidate objects worthy of more detailed consideration.

As originally pointed out in Chapter 3, the data associated with an attribute can conveniently be displayed (Figure 5.7) in the form of a histogram composed of rectangles, each associated with one of the houses in the collection. A mouse-hit on the histogram can reveal, as shown in the figure, the value of the attribute for that house. Below the histogram is a scale whose extremities indicate the range of attribute values associated with the objects. Located on the scale are two limits which can be adjusted interactively (Figure 5.8) to specify a range of acceptable attribute values, and hence a subset of the collection of objects. This subset can be encoded with colour, as shown in Figure 5.8. It is, nevertheless, often valuable – as in Figure 5.8 – to retain the (white) display of the unselected objects since, as will soon be apparent, they provide valuable contextual information.

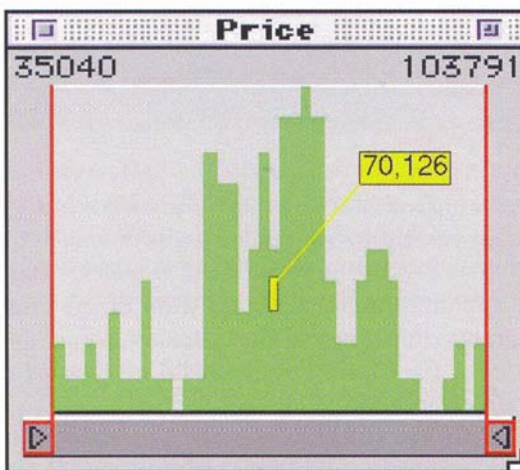


**FIGURE 5.7**  
A histogram of house prices

**FIGURE 5.8**

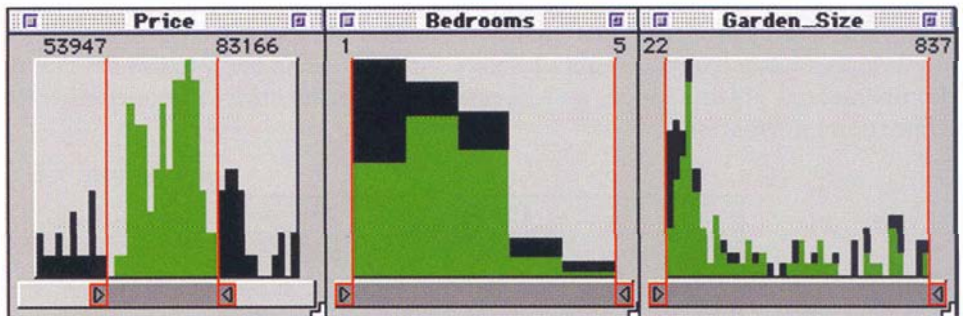
Placement of lower and upper limits to identify a set of houses



Usually, many attributes will be of concern to the user, but in the interests of space we shall consider only three: Price, number of Bedrooms and Garden size. Figure 5.9 shows the corresponding attribute histograms, but with the same limits on Price as in Figure 5.8. Each house appears once on each histogram. The effect shown is that of 'brushing' discussed in Chapter 3, in that the objects selected according to the Price limits are encoded green not only in the Price histogram but additionally in the remaining histograms. Such linking between histograms now allows the interrelation between the three attributes to be easily explored. It is for this reason that a set of interactive linked attribute histograms such as that of Figure 5.9 is called the Attribute Explorer (Spence and Tweedie, 1998).

**FIGURE 5.9**

A selected Price range is brushed over the Bedrooms and Garden Size histograms



A rapid way of gaining initial insight into the interrelations is to use the fact that the 'bar' between the limits on each attribute scale can itself be located manually (Figure 5.10): a quick movement of the Price bar up and down the scale, for example, and simultaneous observation of the green components on the other two histograms, might reveal that cheaper houses tend to have smaller gardens (but perhaps, at the same time, indicate interesting exceptions such as cheap houses with large gardens).

What makes the Attribute Explorer especially valuable is the fact that the set of objects encoded by colour on all the histograms can be determined by limits placed on all attribute values, as shown in Figure 5.11, making it possible for a

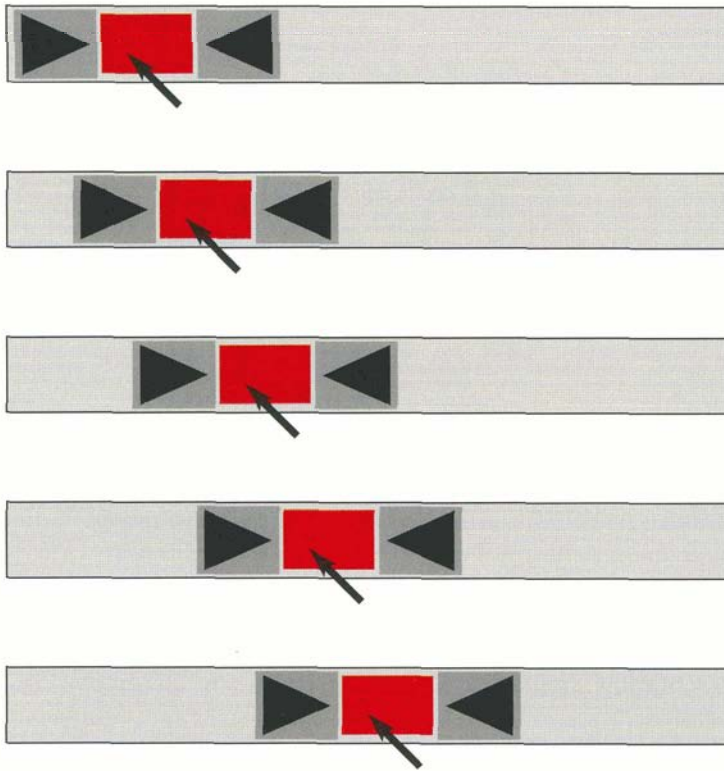


**FIGURE 5.10**  
A bar on a histogram scale can be moved as a single entity

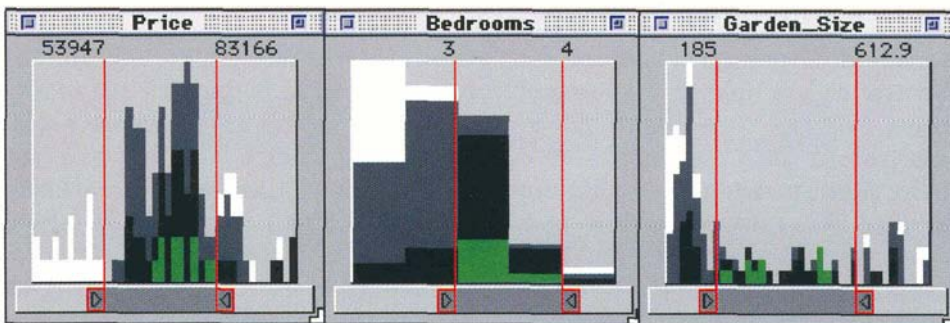


**FIGURE 5.11**  
In the Attribute Explorer separate limits on three attributes combine to identify houses that satisfy all limits

user to explore the interrelations between attributes very thoroughly.<sup>3</sup> Stated formally, the object selection shown in green is the result of performing a **Boolean AND** operation on the objects identified by the separate attribute ranges. Figure 5.12 provides a reminder that latitude and longitude form an interesting pair of attributes that can be handled together, allowing the user to

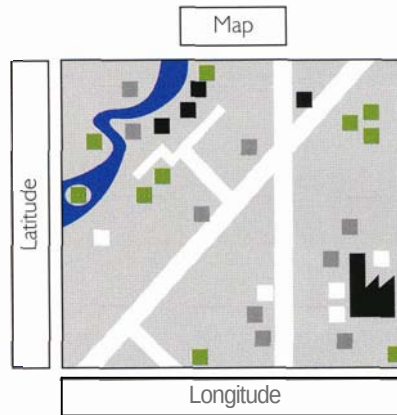
---

The colour-coded rectangles representing houses in the histogram are continuously 'lower justified', for the following reason. If a limit on price is changed it may render unacceptable a house in the Bedrooms histogram, represented perhaps by a rectangle below which there are acceptable houses. Such a change is difficult to follow, so it is arranged that in all columns of all histograms the ordering (starting at the bottom) is green, black, dark gray, light gray and so on.

**FIGURE 5.12**

Latitude and longitude are attributes that can usefully be combined to allow the selection of an area on a map

Source: Keim



select a particular area of the map. There is no limit to the number ( $N$ ) of attributes that can be handled, the number of histograms required being no more than  $N$ : recall that a scatterplot matrix increases in size as  $N(N-1)/2$ .

The fact that each house contributes to each histogram suggests an alternative presentation of histograms, as shown in Figure 5.13. Here, lines ('house-lines') join the contribution of each house to four attribute scales, leading to a presentation similar to Inselberg's parallel coordinate plots (Chapter 3). Animated enhancement of each line (that is, house) in turn can sometimes be helpful in gaining understanding of the displayed data (Tweedie *et al*, 1994).

### 5.4.1 Sensitivity information

A major disadvantage of both the dynamic queries interface (Figure 5.1) and the illustrative example of a command-line query system is the absence of any direct guidance as to how a query might be modified to lead to more useful information. That guidance is especially required when a 'zero hit' or a 'too many hits' situation is encountered. Such a disadvantage can be markedly reduced by 'additive encoding' in which colour coding is applied, not only to those houses satisfying all attribute limits but, additionally and separately, to those failing one, two, three or more such limits. Thus, in the illustration of Figure 5.11, green indicates houses satisfying all limits, black those houses that fail one limit, dark gray those that fail two limits, and so on. It is now apparent how a limit may be usefully changed: a limit extended to include a black house will cause it to turn green. Even (Figure 5.14) for a query which identifies no acceptable (green) houses (that is, the 'zero hit' condition), the additional colour coding shows how limits can be relaxed to discover an acceptable house. Such information, shown by gray-scale encoding, is called sensitivity information, since it indicates the reduction in the number of violated limits achieved by the movement of a limit. The sensitivity information is an excellent example of the value of context: if only the green houses were displayed, no sensitivity information would be available and no guidance provided concerning possibly beneficial exploration.

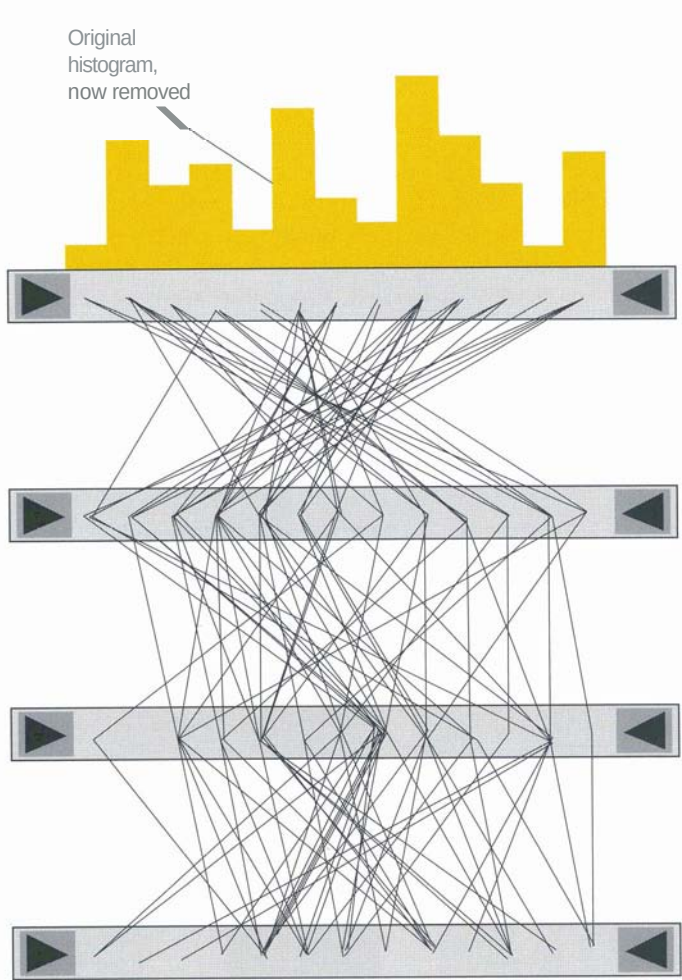


**FIGURE 5.13**  
The principle of parallel coordinate plots can be combined with the Attribute Explorer

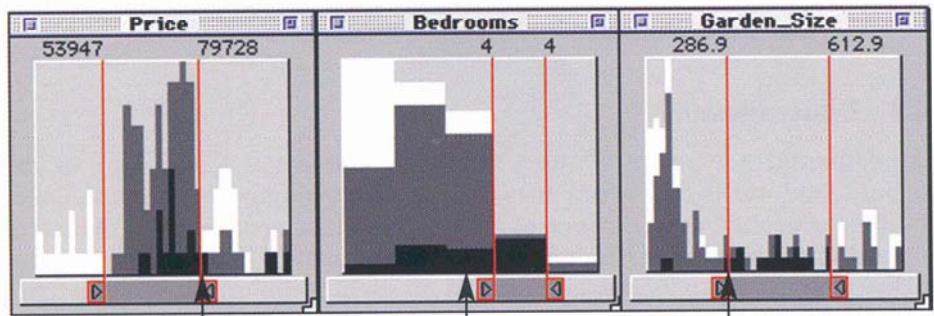


**FIGURE 5.14**  
The 'zero hits' situation, but with black houses indicating possible limit relaxations

by spending a little more...

by accepting three bedrooms instead of four.

by accepting a smaller garden...

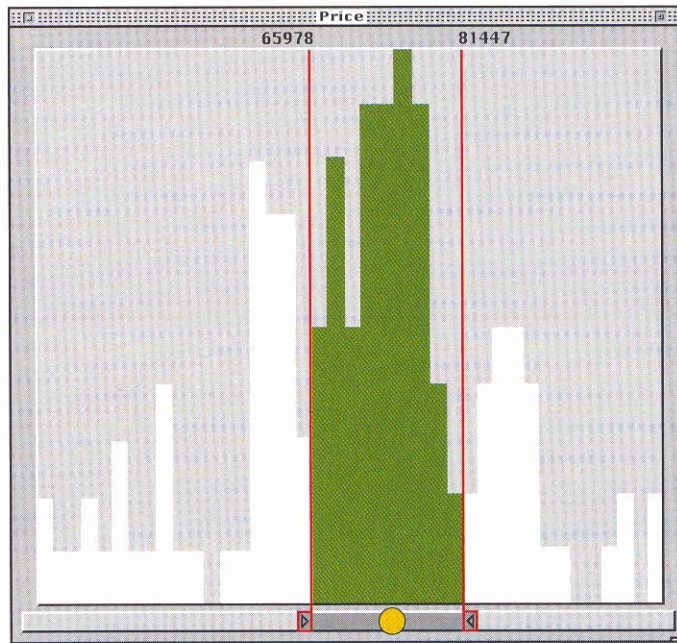
... an acceptable (green) house will appear

### 5.4.2 Summary information

It is not always easy, during exploration, to interpret a dynamic alteration of the color of the separate elements of a histogram. To aid such interpretation, summary information can be made available in the form of a yellow circle (Figure 5.15) positioned on the scale at the average of the attribute values of the selected group of objects. Now, when the range of **Price** is quickly moved up and down the corresponding scale, the yellow average circles on the Bedrooms and Journey time scales move correspondingly, allowing the relation between these attributes to be more readily discerned. Properties such as variance could similarly be encoded for easy understanding (Eick, 1994). In some circumstances, where display area is at a premium (see Chapter 9) the scale alone, without the associated histogram, might suffice

**FIGURE 5.15**

The display of summary information, the yellow circle indicating the average of the values of the selected objects



### 5.4.3 Dimensionality

A fundamental advantage of the concurrent display of histograms in the Attribute Explorer is that it provides a satisfactory solution to the problem of **high** dimensionality: the existence of more attributes simply requires the use of a proportionate number of histograms. This consistent use of a histogram places equal perceptual and **presentational** weight on each attribute, an essential property that would be virtually impossible to achieve by other means except for parallel coordinate plots.

### 5.4.4 Boolean operations

The facility illustrated in Figure 5.11 allowed the user to identify (in green) those houses which satisfied concurrent limits on **Price AND Bedrooms AND** Garden size. **We** emphasize the **Boolean AND** operation because, **in** fact, a wide

variety of **Boolean** operations can be performed on the raw data. To render the Attribute Explorer more flexible for those who may wish to explore some other **Boolean** operations, the concept of the InfoCrystal (Spoerri, 1993) is relevant.

The basis of the InfoCrystal is simply illustrated for the case of three attributes A, B and C in Figure 5.16, which shows how the InfoCrystal is developed from the familiar Venn diagram. Thus, the 'topmost' diamond in the InfoCrystal identifies the operation 'A and (not B) and (not C)'. In the context of the Attribute Explorer, Tweedie (1997) has renamed the InfoCrystal as the Link Crystal to emphasize the fact that the **Boolean** operation links the limits on a number of histograms. The Link Crystal identifying the simple AND operation over all limits, and with sensitivity information additionally shown by the black and dark gray encoding of Figure 5.11, is shown in Figure 5.17. More complex queries can be constructed: the histograms and Link Crystal shown in Figure 5.18, for example, identify by yellow encoding those houses which fail the price limits but no others.

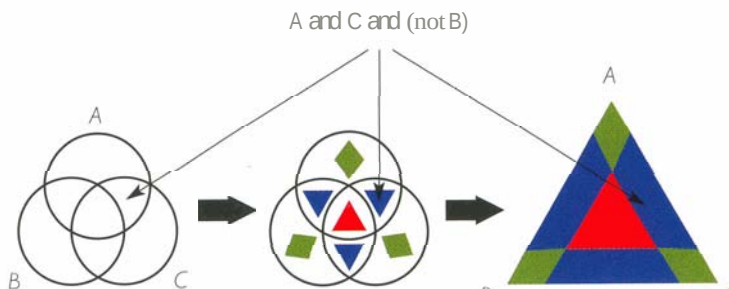


**FIGURE 5.16**  
The development leading from a Venn diagram to the Link Crystal



**FIGURE 5.17**  
The Link Crystal corresponding to the Attribute Explorer display of Figure 5.1

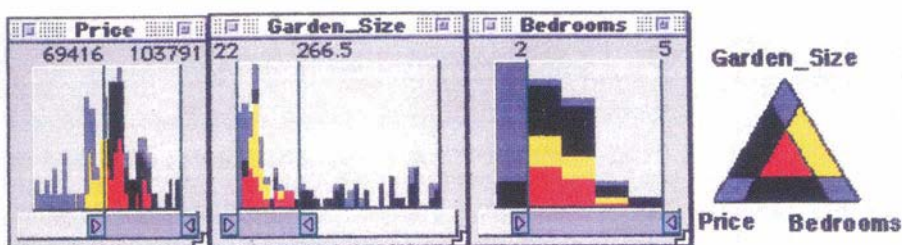


**FIGURE 5.18**  
The Link Crystal and Attribute Explorer display for houses failing only the Price limit

### 5.4.5 Fuzziness

The reader may be tempted to point out that mathematical techniques exist which can handle the concept of fuzziness, that others are capable of optimization, and that these should play a major part in performing tasks such as the search for an acceptable house. Such techniques do exist, and have been developed to a fine art over decades, so it is useful to examine why they are not always ideal for the task of problem formulation and solution.

The underlying reason, stated at the start of this chapter, is that a user's problem is rarely formulated with precision, and that the first objective is to allow the user to gain knowledge of what is available. By contrast, an optimization program requires a user to specify in complete, precise and mathematical detail, what is being sought, as well as many other parameters such as the importance of getting the required number of bedrooms compared with that of keeping within a price limit. The program then returns a single solution without explanation; there is no indication, for example, of any trade-offs that were encountered. Overall, the user is left uninformed. If optimization techniques are to be employed, then very careful attention needs to be paid to the manner in which a user's knowledge and expertise can complement the power of the program: an example is provided by Colgan et al. (1995). With regard to fuzziness, the human brain is probably the best fuzzy computer around, especially when it is concurrently dealing with activities such as searching and exploring.

## 5.5 Very large databases

While there is no fundamental limit – except display area – to the number of objects that can be handled by the Attribute Explorer, it is useful to be aware of other visualization techniques that can also handle a large number of objects. One such technique is embodied in a visualization tool called VisDB (Keim et al., 1993), specifically designed to handle very large collections of data and to explore the limit in which a single pixel represents an object.

The principle underlying VisDB can be explained concisely by reference to an illustrative display (Figure 5.19). In the right-hand window are three vertical scales, each associated with an attribute. On each of these scales are two horizontal bars that can be positioned interactively by a user to indicate preferred lower and upper limits to the attribute's value. The number of objects thereby identified is indicated (1758 for Attribute 1) and the scale between the limits is coloured yellow. Since the satisfaction of all attribute limits by an object is of interest, and because many objects may fail to satisfy one or two limits, there is a facility for assigning a numerical 'weight' to each attribute to indicate the importance of satisfying the limits.

Together, the attribute limits and weights allow all objects to be placed in an order reflecting the extent to which each object satisfies the user's requirements. The question as to how the result can most usefully be displayed has been answered by ordering pixels (or sets of pixels) in a spiral pattern (Figure 5.20), with the 'best' objects at the **center**, and with colour coding to indicate the degree to which the objects satisfy requirements. Yellow denotes that all limits are satisfied, while other colours indicated on the left of the right-hand

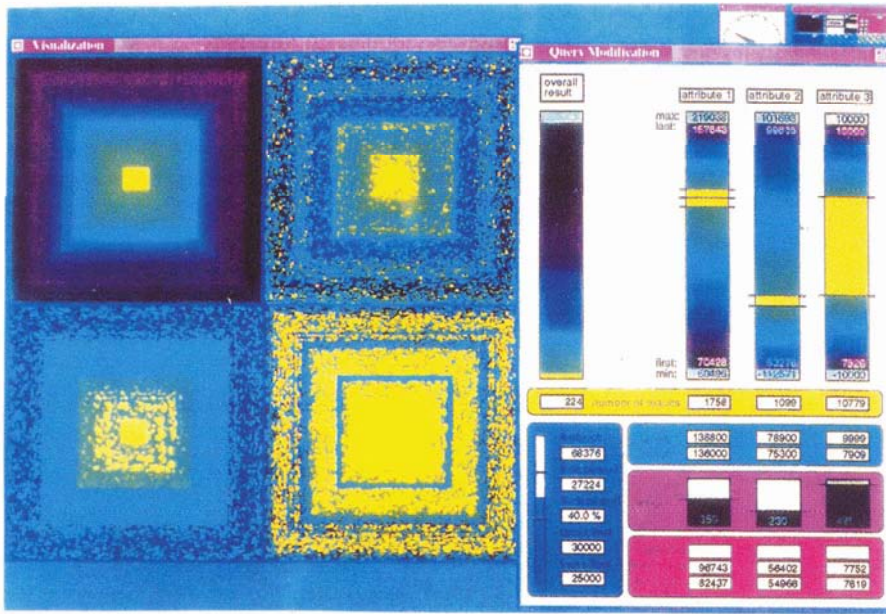


**FIGURE 5.19**  
The VisDB  
visualization  
tool

Source: Keim

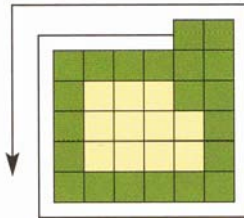


**FIGURE 5.20**  
The spiral  
arrangement of  
pixels adopted  
in the VisDB  
visualization  
tool

window show how the remainder are ordered in the spiral display. The result of forming these spiral displays is shown in the left-hand window in Figure 5.19. The top-left display is associated with the overall satisfaction of all limits according to the assigned weights, while the other three, based on the same ordering, show the extent to which the limits on each attribute are satisfied.

Such a display raises interesting questions as to desirable interaction mechanisms that can enhance the VisDB principle for particular applications. Nevertheless, it is important to note that, like the Attribute Explorer, VisDB offers no convenient mechanism for the display of images associated with attributes. Such a display (though for just a few objects), as well as other desirable features, is a characteristic of the Neighbourhood Explorer, to be discussed next.

## 5.6 The Neighbourhood Explorer

There are many circumstances in which only a relatively small set of data is involved. In house-hunting, for example, it would be quite unusual for a person to consider seriously houses in a very wide price range, such as £50,000 to £200,000. Usually, as a search approaches a conclusion, one is principally

involved in the comparison of a relatively small number of houses, each of which is a relatively close neighbour – in the sense of attributes – of the others. In such a situation a simple rearrangement of the Attribute Explorer, as shown in Figure 5.21, positions houses relative to each other with regard to attributes such as Price, number of Rooms and distance from vineyard. Such attributes are associated with radial scales, with the currently examined house at the **center**: it also allows images to be associated with each house. Within a given sample set of houses it is the ordering of items which determines their relationships, rather than their attribute values, though a continuous linear scale might be adopted to emphasize, for example, relative prices. Exploration is achieved by dragging the image of a house which is of interest along its axis to the **center**, whereupon all of the axes are automatically realigned around this new origin.

**FIGURE 5.2**

Appearance  
of the  
Neighbourhood  
Explorer

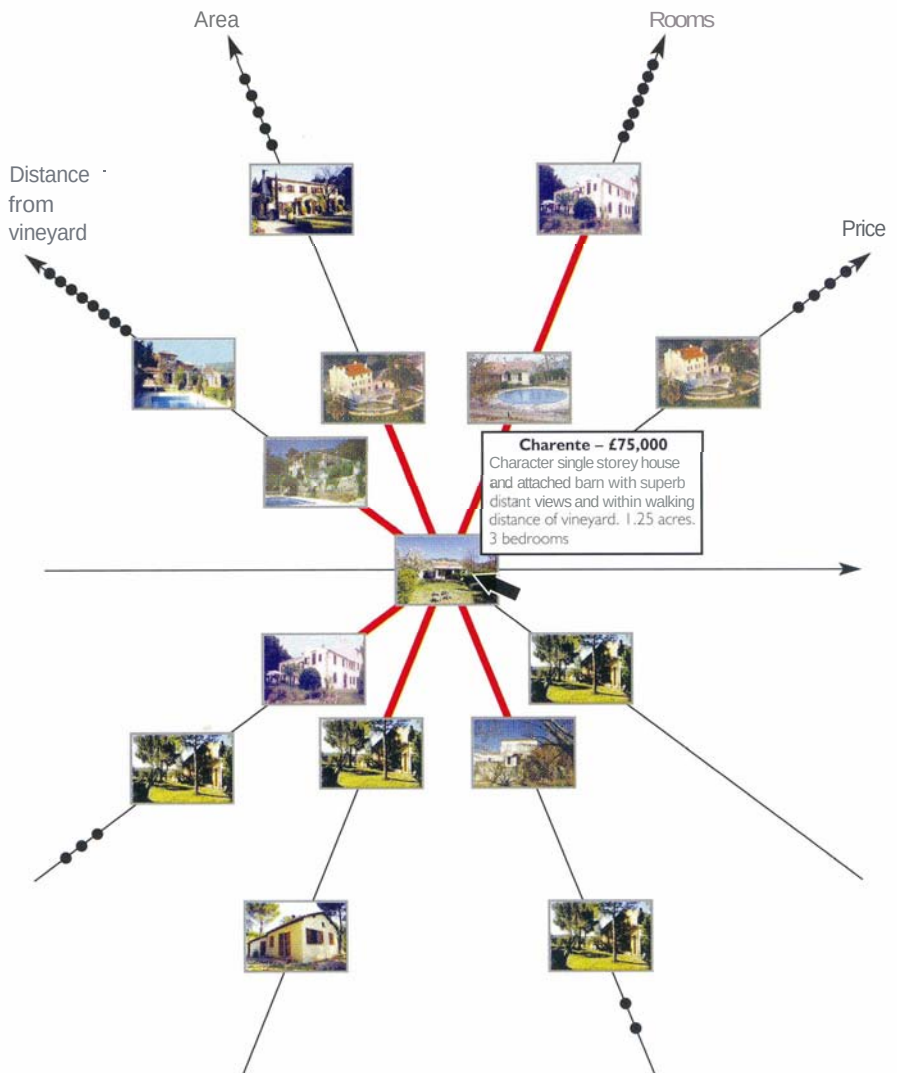
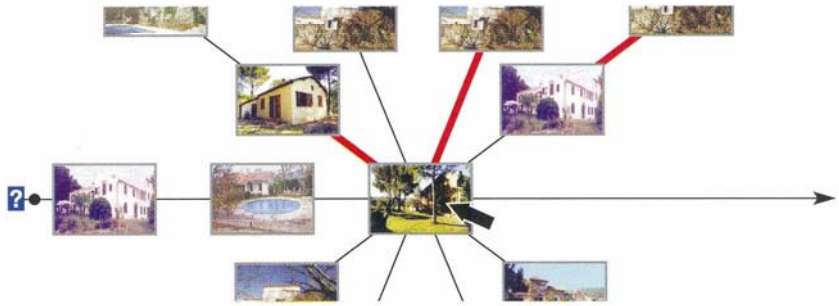




**FIGURE 5.23**

Use of an axis of the Neighbourhood Explorer to record potentially interesting houses



## 5.7 Musical visualization and composition

A memorable quotation from Bertrand Russell(1922) is that:

*A good notation has a subtlety and suggestiveness that makes it seem, at times, like a live teacher.*

There are many fields which benefit from notation: mathematics, chemistry, choreography and others easily come to mind. So, of course, does musical notation. Conventional musical notation is reasonably familiar to most, but that familiarity has to be extremely intimate if it is to be used as a tool for composition. For most people, the sight or **creation** of a line of musical notation does not immediately lead to the visualization of how the corresponding tune will sound. To bring the power of composition closer to the lay person, experimental musical instruments have been devised whose output is not only aural – as one would expect – but also visual, and where changes made by direct manipulation within the externalization of the relevant data is either immediately or quickly heard.

The form of one such instrument is sketched in Figure 5.24; the metaphor on which it is based is the planetary system. In the **center** is the **sun**, and a collection of available planets is located at lower right. Each planet is associated with a given note, which appears as a label on the planet. If a planet is dragged within the circular area it comes under the 'gravitational influence' of the sun and, upon release, moves around it in an elliptical orbit. The user can also position an 'ear' somewhere in the planetary system. As a planet approaches and then recedes from the vicinity of the ear the 'sound' of that planet can be heard at a volume determined by how close the planet is to the ear. The effect of having a number of planets concurrently in orbit is reminiscent of a simple organ. Those with a knowledge of music can easily suggest collections of planets alternative to the pentatonic scale **shown** in Figure 5.24 and other effects such as chords or arpeggios. Though a simple interface, it proved in practice to be remarkably successful with users having no conventional musical **skills**, and the results of compositions by such people were pleasant to listen to. Basically, the success of the instrument is a result of its instantaneous response to manual changes in planet positions, and the simultaneity of the visual and aural presentation and hence the ease of exploration. An aural awareness of an unsuitable sound can immediately be correlated visually with the planet responsible, and the necessary corrective action to take quite easily deduced.

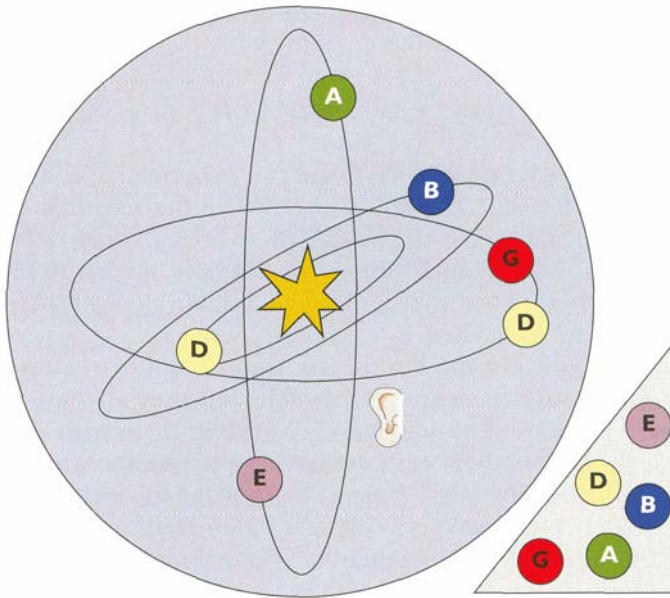


**FIGURE 5.24**  
A musical  
instrument  
based on a  
'planetary'  
metaphor

## 5.8 The Model Maker

An important task in engineering design is to find a mathematical relation (model) between the performances and parameters of an artefact, for the simple reason that the performance of that artefact for *any* chosen parameter values can then be predicted inexpensively before the costly process of manufacture. Suppose, for example, that someone has manufactured four samples of a gearbox, each with a different axle diameter, and has measured, for four different axle diameters ( $X$ ), the torque ( $Y$ ) that it can transmit before damage occurs (Figure 5.25(a)). A simple mathematical relation somehow 'fitted' to the data (Figure 5.25(b)) can then be used (Figure 5.25(c)) with some confidence to calculate the effect of other diameters, and obviate the need for additional – and usually expensive – measurements. While the value of such a model is clear, its derivation can be difficult. It is here that information visualization can help.

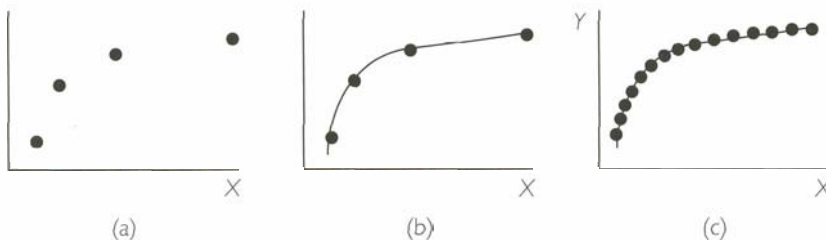


**FIGURE 5.25**  
The derivation  
and use of a  
mathematical  
model

Typically, the type of relation fitted to available data is a polynomial of the form:

$$\text{Performance (P)} = a + bx_1 + cx_2 + dx_1x_2 + ex_1^2 + fx_2^2 + \dots \quad (5.1)$$

where, in this illustrative example, there are two parameters ( $x_1$  and  $x_2$ ) and one performance (P) of interest.<sup>4</sup> The fitting task is then the selection of appropriate values of the coefficients a, b, c and so on. There are two components of the fitting task. One is to choose which terms (for example,  $dx_1x_2$ ) to include in the model. The next is to find the appropriate numerical values of their coefficients (for example, d for the term  $dx_1x_2$ ).

The person carrying out the 'fitting' has two conflicting requirements. They want to keep the relation as simple as possible, but they also want the relation to be sufficiently accurate. The solution is to present them with a display which allows them to choose whether or not to include a particular term in the polynomial, but which additionally offers support by showing the estimated accuracy of the resulting model. The freedom to choose which term to next include in the model results in there being a large number of possibilities, so an essential property of any supportive visualization tool is to display the currently proposed model in the context of all possible models 'one step away'.

An interface called the Model Maker (Tweedie et al., 1998) is shown in Figure 5.26. Each square represents a single term in the polynomial being fitted to measured data, and has one of the three possible appearances shown in Figure 5.27. A black inscribed circle denotes a term already included in the model, whereas white circles are associated with terms that are not yet included. The size of those circles provides a measure of how useful the term is (black) or how useful it can be if included in the model (white). A white background indicates that the term can be added or removed on its own, whereas a gray background indicates that, for heuristic mathematical reasons, the term's inclusion requires other terms to be included at the same time. The modeler

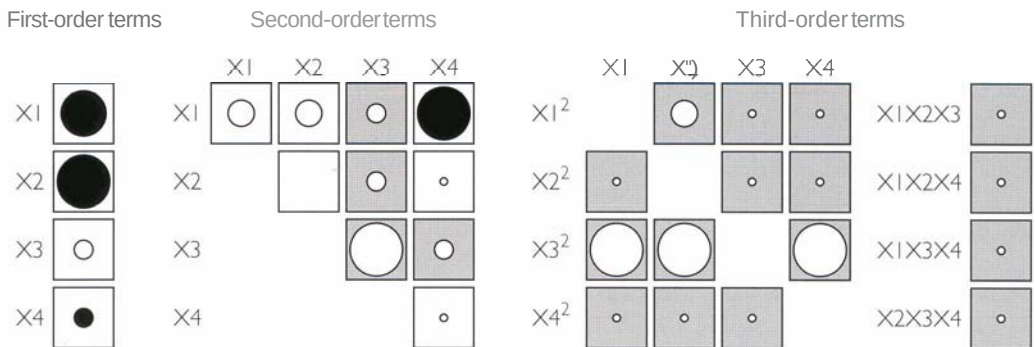


**FIGURE 5.26**

The interface to the Model Maker

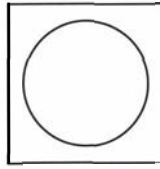
<sup>4</sup> The need to consider four or many more performances and a similar number of parameters would not be regarded as unusual in engineering design.



A **black** circle indicates a term already **included** in the model

The term is **moderately** valuable (**medium** size circle)

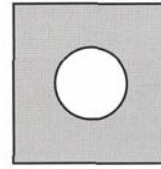
The term is statistically **legal** (**white** background)



A **white** circle indicates a term **not** yet included in the model

Circle size indicates that it would be **very** valuable (**large** circle)

The term is statistically **legal** (**white** background)



A **white** circle indicates a term **not** yet included in the model

Circle size indicates that it would be **moderately** valuable (**medium** circle)

The term cannot be included on its own (**gray** background)

**FIGURE 5.27**  
Representation of the three possible states of a term in a mathematical model

might start by looking at the single ('first-order') terms, such as  $bx_1$  and  $cx_2$ , each represented by a square, and by searching for the largest interior circle: this is, by definition (Figure 5.27), the one whose inclusion is likely to be most beneficial to a good 'fit'. Its inclusion in the model (with a coefficient calculated by a background algorithm) is achieved by a single mouse-click on the selected square, whereupon the circle will turn black to confirm its inclusion. Concurrently, a value of a 'goodness of fit' of the current model is updated. The **modeler** then continues the process, including more terms and possibly higher order ones, until a satisfactory balance is achieved between goodness of fit and the model's simplicity. The basic fitting actions just described are complemented by many other facilities in the actual Model Maker. An example of a fitted polynomial is provided as Equation 9.1 in Chapter 9.

The principal motivation for the development of the Model Maker is the generally low level of statistical knowledge possessed by design engineers, coupled with the typical absence of domain knowledge possessed by statisticians. The interfacce described is a visualization tool like all others described in this chapter in that it allows dynamic exploration among a large number of possible outcomes, but differs in the sense that considerable computation proceeds in the background to compute the sensitivity information (the sizes of the interior circles), the corresponding coefficients (that is,  $a$ ,  $b$ ,  $c$ ,  $d$ , etc.) and the goodness of fit.