Types of diagram adaptation

Cameron McCormack, Kim Marriott, and Bernd Meyer

Monash University, Clayton School of Information Technology
Clayton Victoria 3800 Australia
{clm,marriott,berndm}@csse.monash.edu.au

Abstract. Interactive media and the web require a new kind of diagram, one that can adapt its presentation to suit the viewing context by taking into account the viewing device and the needs of the viewer. As a first step we have tried to identify and understand the kind of adaptation that such diagrams could usefully exhibit. We examined nearly 200 diagrams from a wide variety of application areas and detailed how each type of diagram could be sensibly adapted to different viewing environments. Based on this we have identified the different kinds of adaptation that are appropriate for the diagrams in our corpus. Although the classification of adaptations is likely not complete it is the first attempt that we know of to provide a comprehensive list of adaptation techniques and we believe it will provide a good foundation for future research into adaptive diagrams.

1 Introduction

We believe that interactive media and the web require a new kind of diagram, one that can adapt its presentation to suit the viewing context by taking into account the viewing device and the needs of the viewer. There is considerable interest in intelligent, adaptive documents, that is documents whose appearance adapts to the viewing context and which support user interaction.[6] The desire to separate form from content, in part to allow adaptive layout, has been a major concern underlying the design of web document standards such as HTML, CSS and XSL. Grid-based templates have also been shown to be a promising document layout system.[5] But the focus has been almost exclusively on textual documents with virtually no exploration of what it means for diagrams to adapt their appearance to their context or how to support this.[1]

However, it is also important for diagrams to adapt and the importance of doing so is set to rise for at least three reasons. First, it is increasingly common for document content to be generated dynamically, for instance from a database. Second, the range of devices used to view web pages has increased enormously,

[1] Older raster-based image formats such as JPEG and GIF do not even re-scale satisfactorily. The new resolution independent graphics web format Scalable Vector Graphics (SVG)[3] is a considerable improvement. It supports zooming and uniform rescaling of diagrams, specification of text attributes through style sheets, and allows alternate versions of document elements for different media and languages. But support for more sophisticated adaptation is not provided.

and a diagram needs to look good when viewed using a notebook computer, PDA, or a wall-mounted video display. Third, material on the web is increasingly being used as the primary source for production quality printing. The requirements for print-media based layout are much more stringent than for on-line viewing.

We have identified seven main reasons for modifying or adapting the presentation of a diagram:

space The dimensions of the presentation space may be different from those of the original diagram, requiring the layout to be modified to use the available space more efficiently. This could be accomplished by collapsing whitespace, scaling the diagram, omitting elements of the diagram, or changing its orientation, among others.

medium A diagram should adapt to the presentation medium or viewing device. A diagram printed on a monochrome laser printer, for example, cannot utilise colour. A diagram viewed on a computer or PDA, however, can use colour and can also handle animation and interaction with the user.

user requirements The diagram may need to be modified in order to make its content more accessible to particular users. If the reader of the diagram cannot read the text because it is too small, the font size can be increased and the remainder of the diagram layout adapted accordingly. Diagrams dependent on colours may have to adapt by restyling content to accommodate a colour-blind reader. A completely blind user will need to have the document adapted to a non-visual form or perhaps to a tactile version. Adapting to language is another example of responding to user requirements, as this will require different text strings to be used and consequential layout changes.

context The placement of a diagram on a printed or web page can affect the presentation. Common information between diagrams may be omitted if several diagrams are being viewed together. The context could also include information from other dimensions, such as the list of previously viewed diagrams or the query used to generate a diagram's data.

cultural conventions Cultural conventions can influence how to present a diagram. For example, a diagram that implies an ordering of elements by position relies on the reading order the reader is accustomed to for the language that is being used. An ordering of elements in a diagram for speaker of English will be left to right, top to bottom, whereas an Arabic speaker may infer the ordering to be from right to left, top to bottom.

dynamic content Some adaptation may be required if the data in the diagram is dynamically generated and so may be unknown at design time, for example in an organisation chart that is dynamically generated from an employee database.

In addition to the reasons for adapting the presentation of a diagram, we need to consider the role of user interaction when the diagram is targetted to an interactive medium such as the web. Interaction with an adaptive diagram falls into two categories: control and experimentation. Control is where the user can influence the adaptations that affect the diagram. This allows the user, for example, to zoom around the diagram, expand and collapse group contents or increase the font size. Experimentation is used to help the user understand the information behind the diagram by playing around with it. This might be in an information preserving fashion, such as highlighting relations (e.g. highlighting the corresponding entry in the legend when moving the mouse over a data point in a chart), or it might be information mutating, such as changing the inputs to a diagram that demonstrates a process or computation.

We wish to provide a generic computational basis for this new kind of diagram. A necessary first step is to identify and understand the kind of adaptation that diagrams could usefully exhibit. To do so, we examined nearly 200 diagrams from a wide variety of application areas and detailed how each type of diagram could be sensibly adapted to different viewing environments. Based on this we have categorised the different adaptation types possible. This is the primary contribution of the current paper. Although our categorisation is probably not complete it is the first attempt that we know of to provide a comprehensive list of adaptation techniques and we believe it will provide a good foundation for future research into adaptive diagrams.

There is relatively little previous research into adaptive diagrams. In [9] we presented constraint-solving extensions to the Scalable Vector Graphics (SVG) web standard, motivating the extension by the need to adapt diagram layout.

2 Types of adaptation

Adapting a diagram involves changing the syntactic elements to achieve the objectives of the adaptation while ensuring that the information in the diagram that must remain the same does so. While there are many ways to represent the same information in a diagram, not all will be appropriate as candidates for adaptation. Previous work has been done on identifying compatible representations for information[4, 10], but performing automatic adaptation of a diagram is difficult, as it would rely not only on the possible appropriate representations of the information, but also must take into account design issues such as aesthetics and ease of understanding, which will differ between diagram types.

Adaptation must also be understandable and predictable by the author so that diagrams can be authored with a clear idea of how they will respond in different viewing environments. In order to determine what transformation an adaptation will perform on the diagram, we will look at the different syntactic elements and their semantics that we found in our collection of diagrams. The types of adaptations can then be framed in terms of these elements.

All diagrams comprise a few basic elements: shapes, images, lines and text. The information in a diagram comes from the ways these elements are used. First, the diagram elements can represent things themselves. Shapes and images can be symbolic, and represent some object in a non-literal fashion, or they can be visually similar to the thing they are representing.

But the majority of information in a diagram comes from the spatial relationships of the diagram elements, such as position. Precise positioning can indicate the distance between the physical objects the elements represent. Positioning can also be non-quantitative; placing one element to the left of another can imply an ordering. Containment can indicate a relationship such as hierarchy, composition or grouping. The proximity of two elements often indicates some sort of correspondence between them, for example placing text labels near another element would indicate that it is that element that is being labelled, and connecting two elements with an arrow creates some form of subordination relationship, such as ordering or hierarchy. Alignment, spacing, layering and size are other spatial relationships that may represent information in a diagram.

The styling of the visual elements can indicate some sort of property or relationship. Often, elements are styled similarly to denote a correspondence between them, as used in a bar chart legend. Shades of colours can be used to represent numerical data.

Finally, with dynamic media, diagrams can include animation. Animation is used to indicate change and movement. It can be discrete, showing a sequence of states, or continuous, showing flowing movement.

Adaptation can be viewed as a semantics preserving (at least of the relevant semantics) transformation of a diagram. As we have seen there are many such transformation since there is a wide choice in how information can be conveyed in a diagram. However not all transformations are useful or practical, for example performing a complete re-layout of a diagram in response to a minor change of font size is probably only going to annoy the diagram author. In order to identify those kind of transformations which are useful for adaptation and which can be usefully applied to real-world diagrams we have collected nearly 200 diagrams from a variety of current academic journals, newspapers, text books and from the web for analysis. Although there were many more than 200 diagrams in the sources that we studied, we collected only those that used a method of information representation that we did not already have. The diagrams were organised into broad types (such as trees, flow diagrams, maps and processes) and we attempted to perform what we considered reasonable adaptations on a few of each type of diagram. A reasonable adaptation, in this instance, is a modification to the diagram that achieves at least one of the adaptation objectives listed in the introduction while ensuring that it conveys the same information as the original (or degrades or discards information only where appropriate).

The following are the seven kinds of adaptation that we have identified from our collection of diagrams:

layout Layout involves repositioning and resizing diagram elements. This is usually done in response to the available space for rendering a diagram. The simplest form of layout adaptation is to compress or expand the elements in a diagram to change the amount of whitespace used. Such repositioning retains the spatial relationships of the elements in the diagram. This may not always be required, however, and in diagrams where the relative positions of elements does not encode any information, the elements could be moved anywhere in the diagram. Reorientation is a constrained form of repositioning where, for diagrams that are layed out linearly in general, it may be possible to change the orientation from horizontal to vertical, or vice versa.

If size is not encoding any information, diagram elements may be scaled to take up more or less space. This scaling could be constrained to preserve the aspect ratio or not, depending on the purpose of the elements in the diagram. Rotation of elements is also sometimes possible and can be used to conserve space in one dimension, as with text labels on a chart axis.

Finally, more extreme layout, for example by using general graph layout algorithms, may be appropriate for some diagram types.

focus Focussing is used to draw attention to more important parts of a diagram, and to reduce (or hide completely) peripheral information. Focussing takes two forms: zooming and importance reduction. Zooming results in displaying only a part of the overall diagram. The focus of the zoom could be a graph node, a group (or several groups), an axis range, a detail level or just a simple area. Zooming would often be instigated as a result of interaction. As opposed to zooming, which hides parts of the diagram outside of the focus of the zoom, importance reduction will keep the same view of the whole diagram but cause peripheral elements to be less pronounced. This could be done by scaling down the peripheral elements, as if viewing the diagram through a fish-eye lens, or restyling or repositioning them to have less visual impact.

detail A diagram can be adapted by changing the amount of detail shown. Whereas focussing will show only a certain part of a diagram, changing detail will still show the entire diagram but coarsen the information being shown. Levels of detail are often used in mapping applications, where very detailed cartographic information may be available. When viewing a map of an entire country, it is not useful to display all roads. Displaying only highways would be a reasonable level of detail to display, however. Using higher levels of detail will cause progressively more elements in a diagram to be shown. Diagrams can be adapted to the level of detail required based on, for example, the space available for displaying the diagram, or the user's preference.

Contents of groups and subordinate elements in hierarchies may be shown or hidden just as with levels of detail, depending upon user interaction or available space. This can be used to explore a diagram without cluttering it with all detail at one time.

In diagrams that show numerical data it is sometimes not completely necessary to show all datapoints to retain the intention of the diagram, thus another form of detail adaptation is to remove some of the datapoints. It may be that the diagram is showing a trend with a line chart. If there is insufficient space to display all of the datapoints, it might be possible to elide some of them. Labels on a chart axis and contour lines on a map are other examples where values can be omitted.

Similarly, diagrams that have repeated items in them often use some syntax, such as an ellipsis, to show that some of the items have been omitted.

The number of items that have been omitted could change with adaptation, depending on the amount of space available.

form The representation that is used to encode certain information in the diagram can be changed as a result of adaptation. Changing form could change the spatial relationships used to encode a given relation; for example, changing a hierarchy from using position and connector lines to using containment. Charts are another example of a diagram type that could easily have its form changed–stacked bar charts that add up to 100% can be changed into pie charts, or vice versa. A more extreme form could be to change a diagram to a completely textual description of the information therein.

factoring and indirection Factoring information in a diagram is the process of moving common information from multiple parts of the diagram into one place, and using some sort of correspondence to relate that information back to its original location. The obvious example of factoring is a legend. Rather than have labels next to all the elements of a diagram, the elements can be coloured and these colours associated with the labels in a legend. This way, duplicate text strings are eliminated, freeing up space in the diagram. Introducing indirection in to a diagram is a similar process, but does not eliminate duplicated information. By adding a new correspondence, diagram elements can be moved from one part of a diagram to another while still keeping their original spatial relationship required to evoke a particular semantic relationship. An example of this is moving text labels into a key, away from the elements that they label, and replacing them with numbers that are an index into the key. The proximity of the new numeric labels to the labelled elements keeps the correspondence relationship, even though the actual labels have been moved away.

text content The text content of a diagram can be adapted according to language or space needs. Alternate, equivalent text that is shorter than the original text can be used to conserve space. Text content can also be changed to adapt to the reader's language preference.

animation Animation can be introduced or removed from a diagram to adapt to the abilities of the medium on which the diagram is being rendered. If a diagram is conveying change, for example, this could be achieved by visualising the changes as an animation if the medium supported this. For a printed paper version of the diagram, however, the change could be represented by rendering snapshots of the system at various points while the change is occurring.

3 Example adapted diagrams

To illustrate the different reasons for adaptation and the adaptation types themselves, we present six representative diagrams from our collection and give adapted versions of these diagrams, explaining the first example's adaptations in detail.

3.1 Cell division

This first example, shown in Figure 1(a), depicts the four steps involved in the process of cell division. Arrows are used to indicate progression between the steps. Text is used for both describing what happens at each step and also to label the parts of the cell in the first step. For the first adaptation, let us assume that the diagram is to be presented in a space that will not fit this vertical orientation, but that there is enough space to present the steps horizontally. The diagram could thus be adapted to the available space to look like Figure 1(c).

There are some aspects of the original diagram that are important to keep in the adapted diagram, mostly for reasons of aesthetics and clarity. First, the lengths of the arrows are all equal and the space taken up by all four steps are equal. In the adapted version, these properties are kept, but because the arrows now run horizontally, in the same direction as the text, the arrows must be long enough to accommodate the widest label. To ensure that the labels describing the parts of the cell in the first step do not unnecessarily cause that step to take up more space, the layout of the labels is slightly different from those in the original diagram, where they are vertically left aligned.

The second adaptation of this diagram, Figure 1(b), demonstrates adapting to even less available rendering space and also the ability of a medium (in this case, a computer-based browser) to provide user interaction. size. The smaller rendering space is insufficient to show all four steps at once. Since this adaptation is targetting a computer-based browser, we can use zooming controlled by user interaction to allow a reader to navigate through the steps of the cell division process, thereby reducing the amount of diagram needing to be displayed at the one time. The figure shows views of the rendering zoomed to each step. Since nagivation between the steps requires some form of on-screen control, each view has hyperlinks at the bottom of the diagram to move to the next and previous step, as appropriate. The steps of the cell division could also be shown with animation.

Figure 1(d) shows an adaptation to a more complex user requirement: that the diagram is to be rendered as a uniform height tactile diagram for a blind user. There are a number of transformations to the diagram that need to be made for it to be suitable as a tactile diagram. One obvious transformation is that colours must be removed, and either replaced with texture or just by a non-textured raised area.[2] In our cell division example the only information encoded with colour is the identification of the plasma membrane and cell wall. Since these parts are labelled in the first step and remain in the same relative position in subsequent steps, the colour can be removed. Note though that since the lines that represent the plasma membrane and cell wall have no separation, if these two parts are to be mapped on to the one "colour" they will need some gap so that the reader can distinguish them. A second transformation required is that the English words must be rendered in Braille. It is not sufficient to use

[2] In the figure, black represents raised area in the tactile diagram while white represents unraised area.

the same text content with a Braille font, however, since each character does not map to a single Braille character, and Braille text often uses a system of abbreviations to aid comprehension.[3] It is closer to a translation of the diagram into a different language, from the point of view of adaptation, than a simple change in font. Another point is that complex icons are not as easily discerned in a tactile diagram, thus we have replaced the chromosome icon with a short textual identifier (⠉⠓⠗⠕⠲, “chro.”) that is also specified in the “Bacterial chromosome” label.

3.2 Bar chart

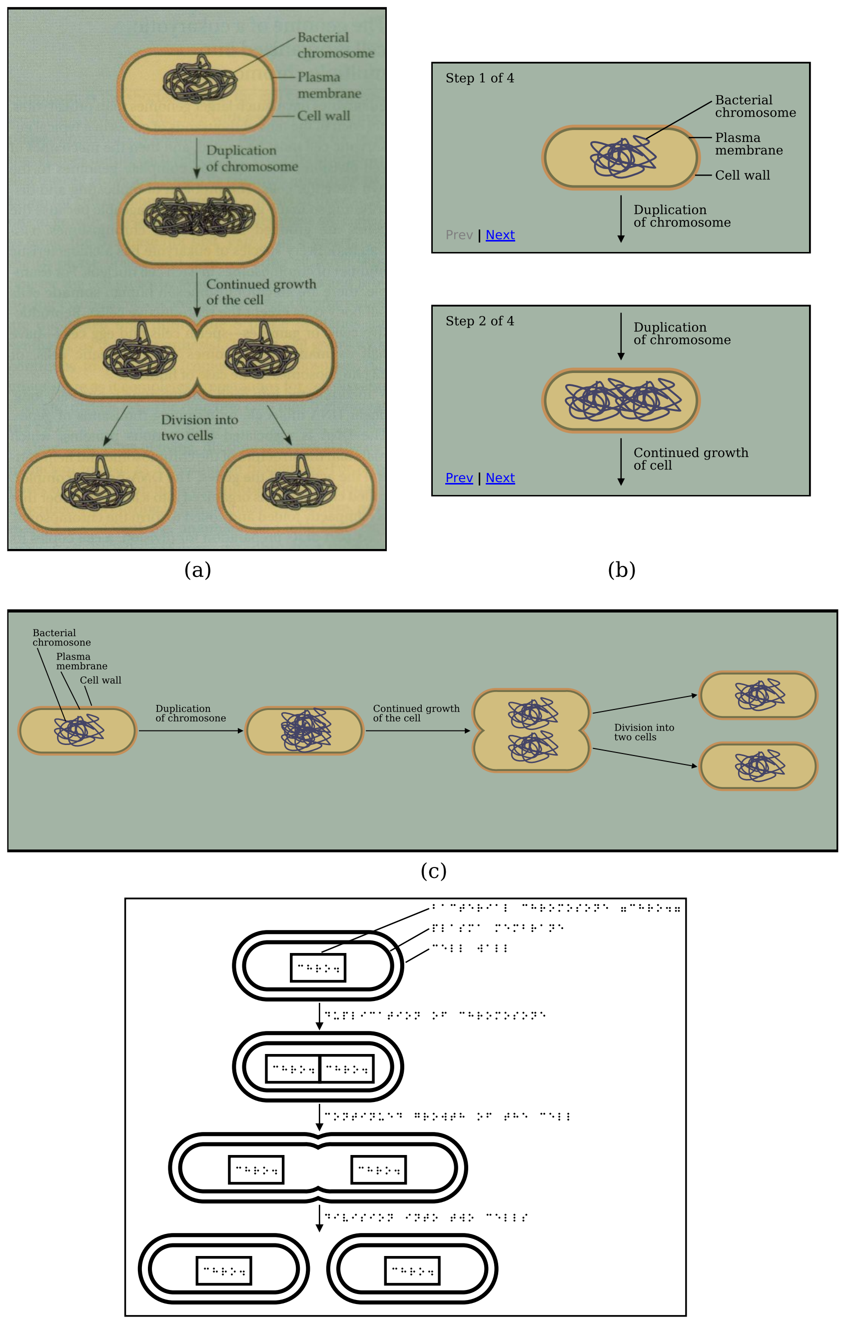
The second example, in Figure 2(a), is a typical grouped bar chart. The first adaptation, shown in Figure 2(b), illustrates how a bar chart could adapt to an increase in desired font size. The original diagram has quite a long title, and in the adapted version an abbreviated title is used–this is an example of using alternate text content to conserve space. Also, the number of vertical axis grid lines has been reduced due to the space that the axis labels now take up. Whitespace between the bar groups has also been compressed a little. Finally, the horizontal axis labels have been rotated to reduce the amount of horizontal space needed.

The second adaptation, Figure 2(c), shows how the bar chart could be presented on a PDA screen. The bars in a group have been made to overlap to conserve space, and importance reduction is used to allow the currently focussed bar group to be wider than those that are not focussed. The legend has also been removed, the functionality replaced by interaction that shows the data series name (and also the data point value) when tapping the stylus on a bar. This example also shows adaptation to a different language.

3.3 Decision tree

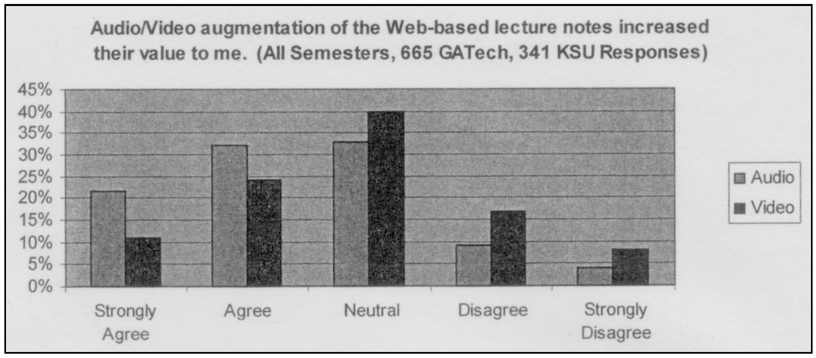
Figure 3(a) shows the third example diagram, a decision tree. The tree has a number of branches which can be followed to diagnose a medical condition. A form of detail reduction (collapsing a subtree) is shown in the first adapted version of the diagram in Figure 3(b), as a result of user interaction. The second adapted version, in Figure 3(c), shows a more comprehensive form of user interaction. Here, the diagram has been adapted to the ability of the medium, a PDA, to allow the reader to navigate through the decision tree. This is a method of focus control. Clicking on a node at each level will cause the diagram to be zoomed to that node, showing the new child nodes as options in the centre of the display.

[3] Note that the figure does not use Grade 2 Braille, the standard system of contractions, due to the authors’ unfamiliarity with the system.

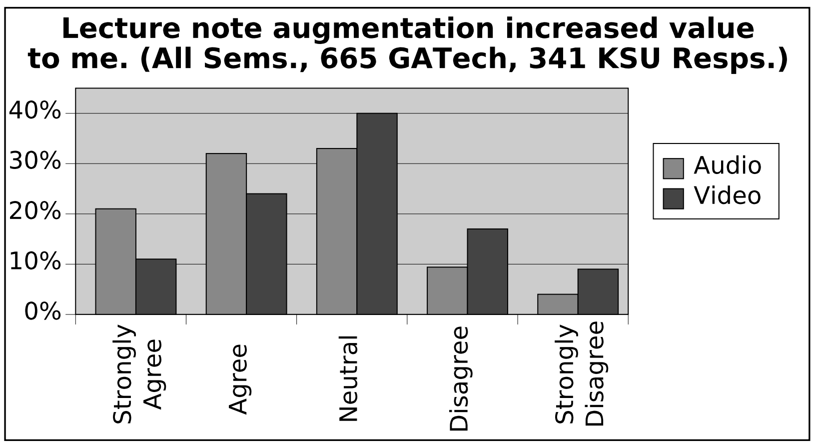


(d)

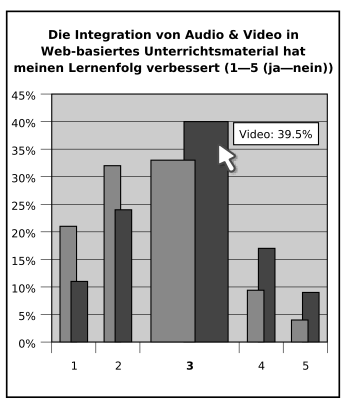
Fig. 1. Cell division diagram: (a) original diagram[2], (b) interactive step-by-step version, (c) horizontal version, (d) tactile version.



(a)

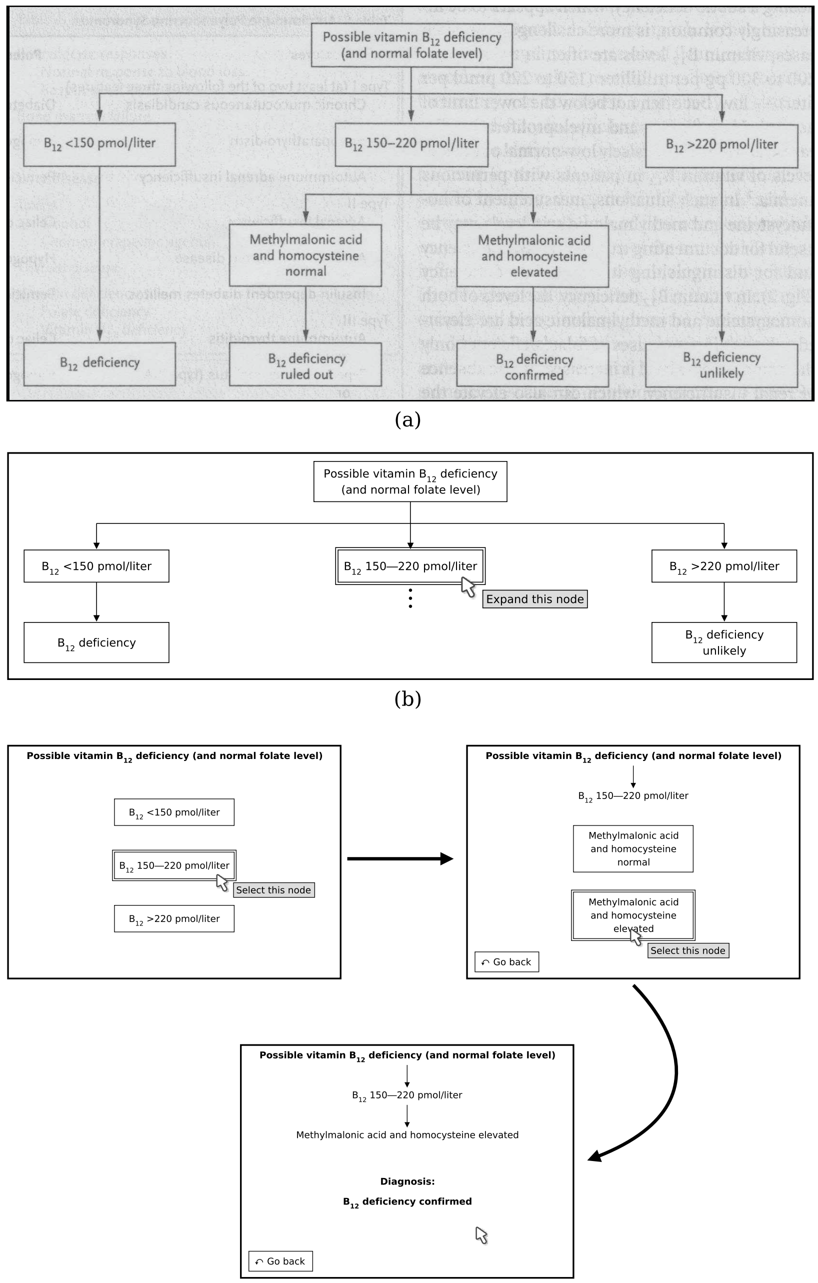


(b)



(c)

Fig. 2. Bar chart: (a) original diagram[1], (b) adapting to a large font size user requirement, (c) adapting to a different language, small screen size and adding interaction.



(c)

Fig. 3. Decision tree: (a) original diagram[7], (b) collapsed group, (c) interactive exploration.

3.4 Pie chart

An expanded pie chart, Figure 4(a), is the fourth example diagram. This chart is interesting because of its exploded view of the "Other" pie segment of the main pie at the left. This leads us to recognise that this subordinate information can be omitted, and Figure 4(b) shows the pie chart adapted to a smaller viewing area by lowering the level of detail and omitting the second pie. Figure 4(c) demonstrates an extreme kind of adaptation, where the form of the diagram has been changed completely. The data is now represented in a table, which may be a suitable adaptation for a very small rendering area or for the input to a speech synthesis engine for a blind reader.

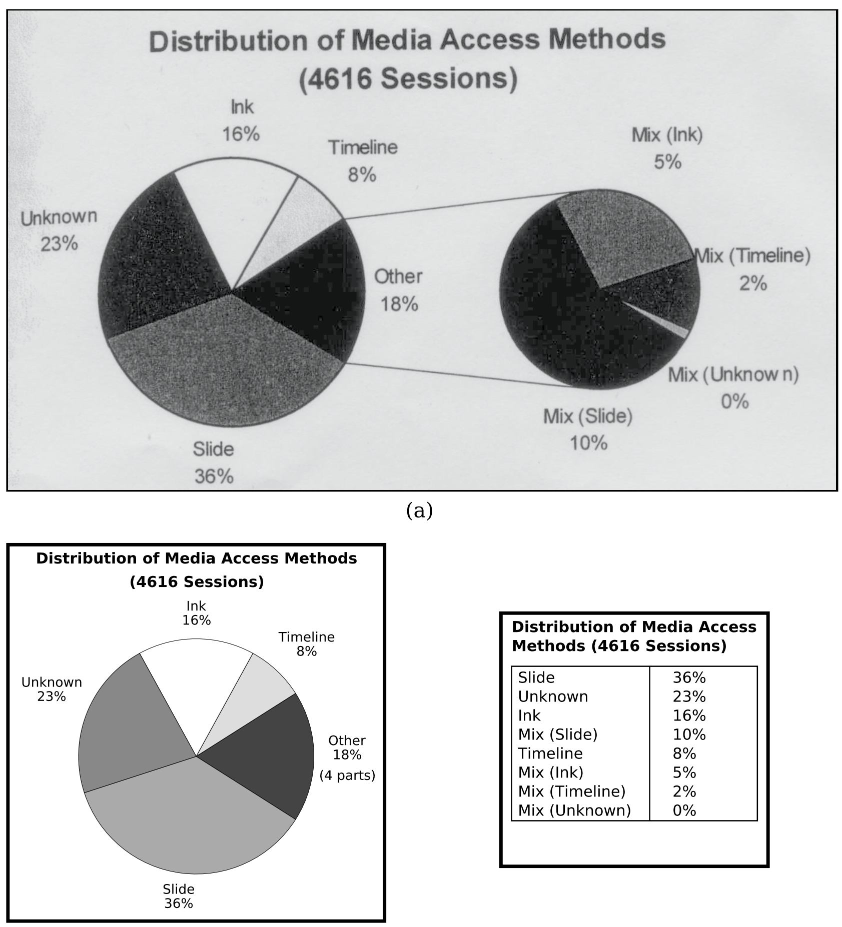


Fig. 4. Pie chart: (a) original diagram[1], (b) level of detail reduced, (c) form changed to a table.

3.5 Exploded part diagram

The fifth example diagram, an exploded part diagram, is given in Figure 5. This kind of diagram is common for illustrating how mechanical parts fit together. Possible adaptations for this diagram include removing the part labels to conserve space, displaying them only with user interaction, or allowing the user to shift the parts along the axis so that they can experiment to understand better how the parts fit together.

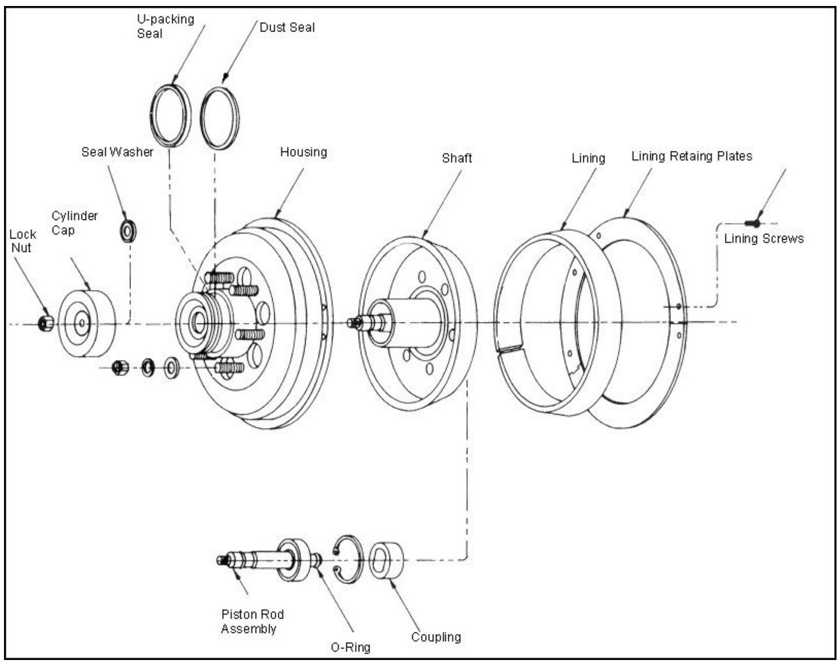
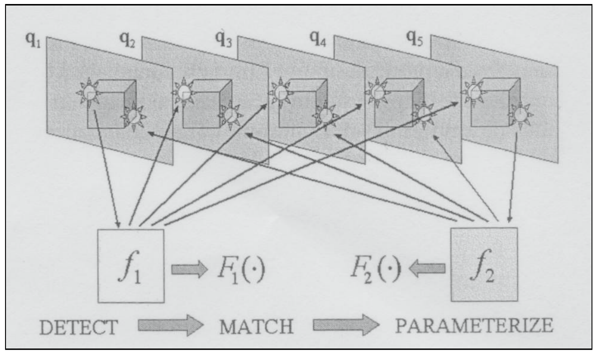


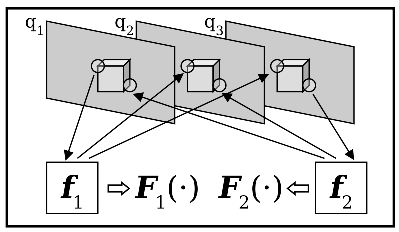
Fig. 5. Exploded part view diagram.[8]

3.6 Image processing feature recognition diagram

The final example demonstrates adaptation by removing repeated items from a diagram. Figure 6(a) shows an illustration of an image processing framework. This diagram has five image frames, but for the purpose of the diagram it does not matter how many frames are actually present, as long as there are a few. Figure 6(b) shows an adapted version of the diagram where this repetition has been reduced to save space.



(a)



(b)

Fig. 6. Diagram illustrating image recognition feature detection: (a) original diagram[11], (b) with repetition reduced.

4 Conclusion

We have examined nearly 200 diagrams from a wide variety of application areas and detailed how each type of diagram could be sensibly adapted to different viewing environments. Based on this we have developed a collection of different kinds of adaptation which range from slight modification to the syntax and layout, such as changing the size of the text and appropriately relaying out the rest of the diagram, to performing complete change in representation from a visual form to an audio description.

For diagrams to provide the sort of adaptivity that we have identified we require web standards that allow the specification of such adaptive behavior. In principle this is possible using scripting but it would be better to have high level support in languages such as SVG. However the main unsolved problem is how to specify the desired adaptive behaviour. We do not believe specification of the adaptive behavior can be completely hidden from the author. This may be possible for very restricted kinds of diagrams such as bar-charts and limited kinds of adaptation, but in general we need to develop new authoring tools and metaphors to enable the easy construction of adaptive diagrams. Document authors need to be confident that they control and can predict how the diagram will adapt to different contexts.

The kinds of adaptations identified above provide considerable guidance in developing these, allowing us to create, for instance, a set of tools that generate a new version of a diagram adapted to narrow viewports or for tactile viewing. But this will not happen invisibly, as the author will need to control application of the tools and will be able to fine-tune the generated diagram. Work on an authoring tool for adaptive diagrams is currently underway.

References

1. Jason A. Brotherton and Gregory D. Abowd. Lessons learned from eClass: Assessing automated capture and access in the classroom. *ACM Trans. Comput.-Hum. Interact.*, 11(2):121–155, 2004.
2. Neil A. Campbell. *Biology*, page 205. Benjamin/Cummings, California, fourth edition, 1996.
3. Jon Ferraiolo, Jun Fujisawa, and Dean Jackson. Scalable Vector Graphics (SVG) 1.1 specification. `http://www.w3.org/TR/SVG11/`, January 2003.
4. Corin A. Gurr. On the isomorphism, or lack of it, of representations. In Kim Marriott and Bernd Meyer, editors, *Visual language theory*, pages 288–301, New York. Springer Verlag.
5. Charles Jacobs, Wilmot Li, Evan Schrier, David Bargeron, and David Salesin. Adaptive grid-based document layout. In *ACM transactions on graphics (TOG)*, volume 22, July 2003.
6. Charles Jacobs, Wilmot Li, Evan Schrier, David Bargeron, and David Salesin. Adaptive document layout. *Commun. ACM*, 47(8):60–66, 2004.
7. Peter W. Marks and Lawrence R. Zukerberg. Case 30-2004 – a 37-year-old woman with paresthesias of the arms and legs. *N. Engl. J. Med.*, 351(13):1333–1341, 2004.
8. Kit Masters. Instructions to install 8500SK (Sealkit). `http://kit-masters.com/instruction_kysor8500sk.html`, March 2004.
9. Cameron McCormack, Kim Marriott, and Bernd Meyer. Adaptive layout using one-way constraints in SVG. `http://www.svgopen.org/2004/papers/ConstraintSVG/`, September 2004.
10. Atsushi Shimoima. Derviative meaning in graphical representations. In *IEEE symposium on visual languages*, pages 212–219, Tokyo, Japan, 1999.
11. Robert Sim and Gregory Dudek. Learning generative models of scene features. *International Journal of Computer Vision*, 60(1):45–61, October 2004.