Describing Complex Charts in Natural Language: A Caption Generation System

Article *in* Computational Linguistics · September 1998

Source: DBLP

CITATIONS
88

READS
191

4 authors, including:

Johanna D Moore
The University of Edinburgh
252 PUBLICATIONS **7,996** CITATIONS
SEE PROFILE

Giuseppe Carenini
University of British Columbia - Vancouver
109 PUBLICATIONS **3,493** CITATIONS
SEE PROFILE

Describing Complex Charts in Natural Language: A Caption Generation System

Vibhu O. Mittal[*]
University of Pittsburgh

Johanna D. Moore[†‡]
University of Pittsburgh

Giuseppe Carenini[‡]
University of Pittsburgh

Steven Roth[§]
Carnegie Mellon University

Graphical presentations can be used to communicate information in relational data sets succinctly and effectively. However, novel graphical presentations that represent many attributes and relationships are often difficult to understand completely until explained. Automatically generated graphical presentations must therefore either be limited to generating simple, conventionalized graphical presentations, or risk incomprehensibility. A possible solution to this problem would be to extend automatic graphical presentation systems to generate explanatory captions in natural language, to enable users to understand the information expressed in the graphic. This paper presents a system to do so. It uses a text planner to determine the content and structure of the captions based on: (1) a representation of the structure of the graphical presentation and its mapping to the data it depicts, (2) a framework for identifying the perceptual complexity of graphical elements, and (3) the structure of the data expressed in the graphic. The output of the planner is further processed regarding issues such as ordering, aggregation, centering, generating referring expressions and lexical choice. We discuss the architecture of our system and its strengths and limitations. Our implementation is currently limited to 2-D charts and maps, but, except for lexical information, it is completely domain independent. We illustrate our discussion with figures and generated captions about housing sales in Pittsburgh.

∗ Current affiliation: Just Research, 4616 Henry Street, Pittsburgh, PA 15213; and Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA 15213; e-mail: mittal@jprc.com
† Intelligent Systems Program
‡ LRDC and Computer University of Pittsburgh Science Department}
§ Robotics Institute and Computer Science Department

1. Introduction

This paper describes a framework for generating natural language captions to accompany complex graphical presentations of diverse data sets. It describes an implemented system that integrates two robust systems: SAGE—an intelligent graphics presentation system (Roth et al., 1994), and a natural language generator, consisting of a text planner (Young and Moore, 1994; Young, 1997), a micro-planner implementing tactical decisions, and a sentence realizer (Elhad and Robin, 1992).

Graphical presentations can be an effective method for succinctly communicating information about multiple, diverse data attributes and their interrelationships. More than 80% of all business reports these days contain graphic presentations of data (Beattie and Jones, 1994; Schmid, 1983). When a display includes only a small number of data attributes or can make use of conventionalized graphical styles (e.g., spreadsheet graphics), it is easy for a viewer to understand how to interpret it. However, one of the main goals for automatic presentation systems is to allow users to see complex relationships between different attributes and perform problem-solving tasks (e.g., summarizing, finding correlations or groupings, and analyzing trends in data) that involve many data attributes at the same time. A number of research groups have developed systems that can automatically design sophisticated presentations to support a task -- presentations that are both novel and complex (e.g., (Casner 1991; Mackinlay 1986; Roth et al., 1994)). These graphics are often difficult to understand (Shah, 1995). Clearly, such graphics can only be fully effective for supporting analysis tasks if accompanied by explanations designed to enable users to understand how the graphics express the information they contain. Studies have shown that the presentation of captions with pictures can significantly improve both recall and comprehension, compared to either pictures or captions alone (Nugent, 1983; Large et al., 1995; Hegarty and Just, 1993). This suggests that the generation of captions for statistical graphics is an important application area in which natural language generation techniques can make a significant contribution.

In our system, the graphical displays are designed by an automatic presentation component, SAGE (Roth et al., 1994), and are often complex for several reasons. First, they typically display many data attributes at once. The mapping of many different data attributes to multiple graphical objects in a single display can be difficult to determine from the graphics alone. Second, integrating multiple data attributes in a display requires designing graphics that are unfamiliar to users accustomed to spreadsheet graphics that create simple displays of individual data attributes. While these integrated displays can be very useful once they are explained, it is often difficult to understand them completely without accompanying explanations. Finally, the nature of the data with which we are concerned is inherently abstract and does not have an

obvious or natural visual representation. Unlike depictions of real world objects or processes (e.g., radios (Feiner and McKeown 1991), coffee makers (Walster et al., 1993), network diagrams (Marks, 1991)) and visualizations of scientific data (e.g., weather, medical images), visualizations of abstract information lack an obvious physical analog.

As an example of the type of data we are concerned with, consider the graphic shown in Figure 1. This is a SAGE generated version of the famous graphic drawn by Minard in 1861[1] depicting Napoleon's march of 1812 (Roth et al., 1994). The graphic relates seven different variables: position (latitude and longitude), size, direction of movement, temperature, and dates and locations of battles. Unless one has seen this graphic (or a very similar one) before, it can be very difficult to understand. Indeed, Minard accompanied the original graphic with a paragraph of text, the first half of which is about how the graphic expresses the information the information it contains. [2]

Consider how the following human-generated caption for the graphic in Figure 1 explains the picture and the underlying data[3]

> *This map shows march segments and battles from Napoleon's 1812 campaign. The map shows the relation between the geographic locations, temperature and number of troops for each segment. Each line shows the start and end locations for the march segment. Its color shows the temperature, and the thickness shows the number of troops. The temperature was about 100 degrees for the initial segments in the west (the wide, dark red lines on the left), about 60 degrees in later segments in the east (the narrower, light red lines on the right) and about -40 degrees in the last segments, also in the west (the narrowest, dark blue lines on the left). The number of troops was 400,000 in the earliest segments, 100,000 in the later segments, and 10,000 in the last segments. The city and date of each battle is shown by the labels of a yellow diamond, which shows the battle's location.*

This caption can help users understand the various attributes and the underlying relations between them--conveyed so succinctly by the graphic.

---

[1] The original graphic can be found on page 41 in (Tufte, 1983).

[2] Minard's original caption, translated from French, reads:
Figurative map of the losses, expressed in men, of the army during the Russian campaign. 1812-1813)
Print by M. Minard. Retired general inspector of the "Ponts et Chaussees."
The number of men is represented by the width of the colored zones, where one millimeter corresponds to ten thousand men; moreover, they (number of men) are written across the zones. Red indicates men entering Russia, black indicates men exiting it. The information used to fill this map would have not been available without M. Thiers, de Segur, de Frezerisac, de Chambray, and the journal of Jacob, army pharmacist since Oct. 28th. To help the eye judge how the army shrank, I assumed that the regiments of Price Jeroms and Marchal Davons, that were detached in Minsk and Mobilov, and rejoined near Orscha and Wiltesk, always walked together with the army.

[3] Note that the original picture generated by SAGE was in color; the paper contains gray scale reproductions due to printing limitations.

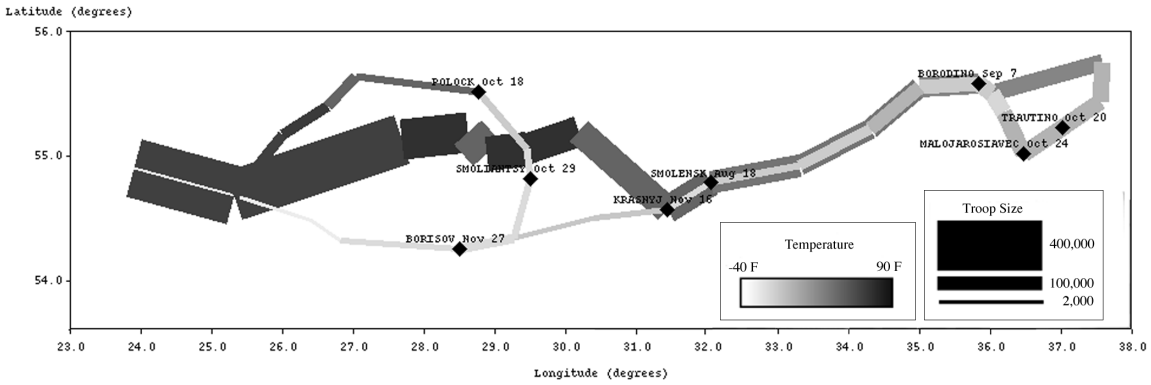


Figure 1
A SAGE generated version of the well known Minard graphic.

Although several projects have focused on the question of how such intelligent graphical presentations can be automatically generated (e.g., (Casner 1991, Mackinlay 1986, Roth and Hefley, 1993, Kerpedjiev, 1992)), they have not addressed the problem of generating the accompanying textual explanations. Without this ability, automatic graphical presentation systems will necessarily be limited to generating conventionalized graphics that do not use novel means to express complex relationships among data attributes, or risk generating displays that users will find difficult to fully comprehend and utilize.

In designing our framework for generating natural language captions we have adapted and integrated work in natural language generation (NLG) by a number of researchers--including ourselves--in different sub-areas: text planning, aggregation, centering, computing referring expressions, example generation and linearization. Given the applied nature of our work, in selecting specific NLG techniques we followed a parsimonious approach. For each sub-task we selected the simplest technique that was capable, in conjunction with the behavior of the other sub-tasks, of producing coherent text that could express the propositions we needed to convey.

The generation process starts with content selection. For this process, we use LONGBOW, a domain-independent discourse planner originally developed as part of a project aimed at generating tutorial (Young and Moore, 1994). Using plan operators that encode discourse strategies devised for the task of generating captions, the planner determines what information should be included in the captions (and consequently what should be left out), and how to organize the selected information. Operator constraints analyze the structure of the graphic presentation and the perceptual complexity of the graphical display to enable the planner to select and apply appropriate strategies. The output of the text planning stage is then further processed by a micro-planner, a sequence of modules implementing inter- and intra-clause ordering,

aggregation, and referring expression computation. The module performing intra-clause ordering is of special interest because it uses a novel technique based on centering theory. Although, we have devised such a technique specifically for generating captions, it is general and can be applied to any discourse structure. The other three micro-planning modules use standard NLG techniques. Ordering and aggregation are based on text genre (i.e., descriptions of information graphics) and domain specific (e.g., real estate sales or stock market data) heuristics. The referring expression module uses a well-known domain-independent algorithm that given an intended referent builds a description uniquely identifying it. The referential problems in our application did not require more sophisticated referring algorithms; there was also no interaction between computing the referring expressions and inter- and intra-clause ordering. Once micro-planning is complete the FUF/SURGE realization module generates the actual English. The modules of our NLG system are discussed in detail in section 5.

In addition to these NLG techniques, generating textual captions for information graphics requires the following knowledge sources:

- a representation of the *syntax* of graphical displays, that is, the structural, spatial and other relations among graphical objects and their properties. For example, the relationship between the end points of the lines in Figure 1 to positions on the map, the fact that an axis conveys the positional values for all the objects within a chart [4] the difference between alternative uses of color: (i) constant color, (ii) color used to distinguish between different attributes, and (iii) color used to encode data values.
- a representation of the *semantics* of graphical displays, i.e., the mapping from data objects and their attributes to graphical ones. For example, in Figure 1 the fact that the temperature during a march segment is mapped to the color of the corresponding graphical segment.
- a mechanism for determining which aspects of graphical displays must be explained based on their perceptual complexity or the complexity of the data attributes they express. This mechanism must take into account information about the underlying data and the perceptual complexity of the way in which data attributes and relations have been mapped to graphical entities.

We describe these knowledge sources and the discourse strategies in the following three sections.

[4] Except those that are positioned relative to other objects, as explained in Section 4.

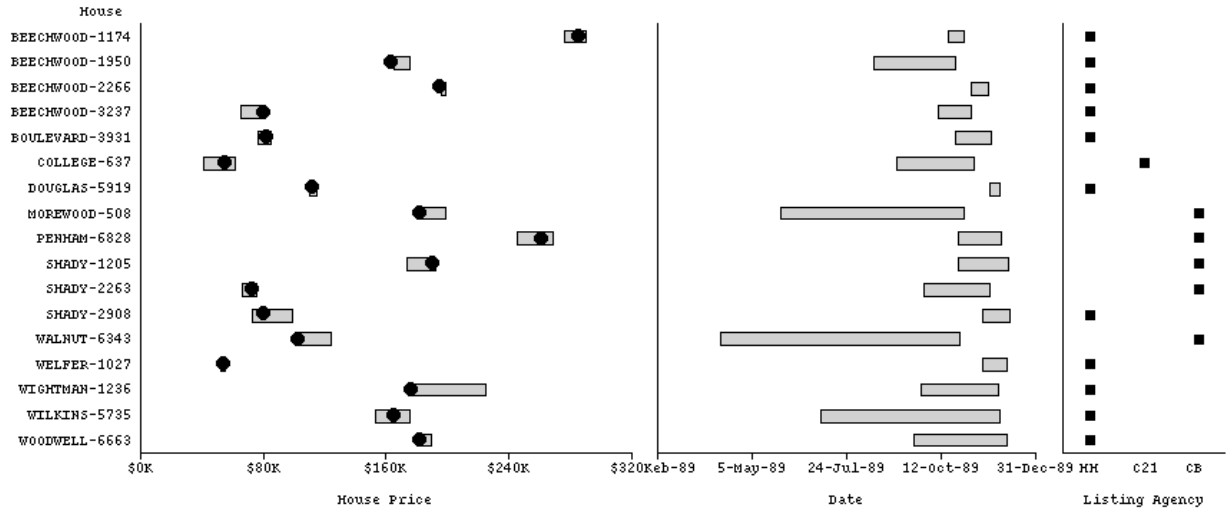


Figure 2
A sample graphic generated by SAGE.

2. SAGE : A System for Automatic Graphical Explanations

SAGE is a knowledge-based presentation system that designs graphical displays of combinations of diverse information (e.g., quantitative, relational, temporal, hierarchical, categorical, geographic). The inputs to SAGE include: (1) sets of data represented as tuples in a relational database, (2) a characterization of the properties of the data that are relevant to graphic design, and (3) an optional set of design specifications, expressing a user's preferences for visualizing the data set.

SAGE's output consists of one or more coordinated sets of 2-D information graphics that use a variety of graphical techniques to integrate multiple data attributes in a single display. SAGE integrates multiple attributes in three ways:

- by representing them as different properties of the same set of graphical objects. For example, both the left and the right edges of the bars in the left most chart in Figure 2 are used to map attributes (*asking-price* and *selling-price* respectively)

- by assembling multiple graphical objects into groups that function as units to express data. For example, the interval bar and the mark in the left most chart in Figure 2 are used to show different types of price-related attributes: *asking-price*, *selling-price* and the *agency-estimate*.

- by aligning multiple charts and tables together with respect to a common axis. For example, the three charts in Figure 2 are aligned on the Y-axis which indicates the house.

Creating a graphic that integrates data in this way is partly an encoding process in which the values of data attributes are converted into graphical values of properties of objects (e.g., color, shape, and spatial position of polygons). Interpreting the information in a graphic is a decoding process, where people must translate visual symbols back into data values. SAGE creates graphics that enable people to perform efficiently information-seeking tasks (e.g., searching for clusters of data values that are different from the rest and looking up other facts to understand what makes them different). However, in designing graphics, SAGE only considers how effectively attributes can be mapped to graphical properties to support a task. For example, a requirement to be able to search for particular values by name might result in the relevant attribute being arranged along an axis in lexicographic order; on the other hand, if it is important to find the maximum and minimum values in a set, SAGE might order these values in terms of magnitude. SAGE , like other automated presentation systems (Casner 1991, Mackinlay 1986,), does *not* take into account perceptual complexities associated with the resulting graphic. For instance, SAGE does not explicitly reason about the difficulties users may have in translating bi-color saturation scales to exact numerical values[5]

SAGE can also design complex presentations that have overlapping objects, or use cluster composition to define a novel combination or grouping of graphical objects in the presentation. This can make understanding some of the graphics that SAGE generates quite difficult. Fortunately, the picture representation used in SAGE contains a complete declarative representation of the content and structure of the graphic in a form that can be used for reasoning by other components. Thus, this representation can be used to reason about possible sources of user confusion arising from mappings that are either complex or ambiguous to the user.

SAGE 's representation serves three functions in explanation generation. First, it helps define what a viewer must understand about a graphic in order to obtain useful information from it. It does this by defining the elements of a graphic and the way they combine to express facts (i.e., how they map to data). Second, the representation describes the structure of both the graphical presentation and the data it presents, so that they can be explained coherently. Finally, the representation helps derive judgments of *complexity* for specifying graphical elements needing text explanation. To understand these three functions, we briefly review the representation.

[5] The Minard Graphic, shown in Figure 1, uses the bi-color saturation technique to map temperature values to the march segments shown in the map.

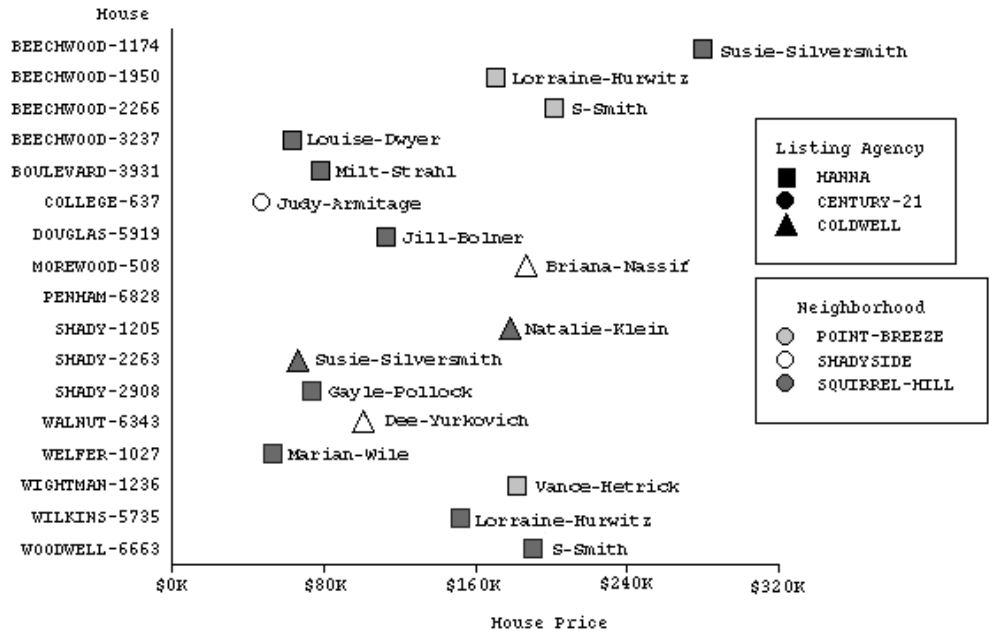


Figure 3
A graphic generated by SAGE to illustrate the use of encoders.

Graphemes are the basic building blocks for constructing pictures. Marks, text, lines, and bars are some of the different grapheme classes available in SAGE. Each grapheme type consists of a definition of the parameters that control the appearance of all graphemes of that type; different grapheme sub-types can be created by varying specific parameters. Individual graphemes can be generated by providing appropriate values for all the input parameters. For instance, individual marks can be generated by providing values for the parameters: *x* coordinate, *y* coordinate, *shape*, *size* and *color* to an instance of a mark class encoder; individual line segments can be generated by providing values for the coordinates *x*1, *y*1, *x*2 and *y*2, *thickness* and *color* to the line class encoder.

Symbol classes are used to organize graphemes into structures that express facts in the data set. A labeled mark, an interval bar and a bar with an attached label are some of the more familiar symbol classes available in SAGE . Each symbol class consists of a definition of the spatial relationship among a set of graphemes and the correspondence between the parameters of this set and attributes types in a data-set. A *labeled-mark*, for instance, would be defined as a combination of a *mark* and a *text label* and the spatial relationship between them. Consider the labeled-marks in the chart shown in Figure 3. The spatial position of the label is dependent on the position of the mark: it is offset slightly to the right and above the mark.

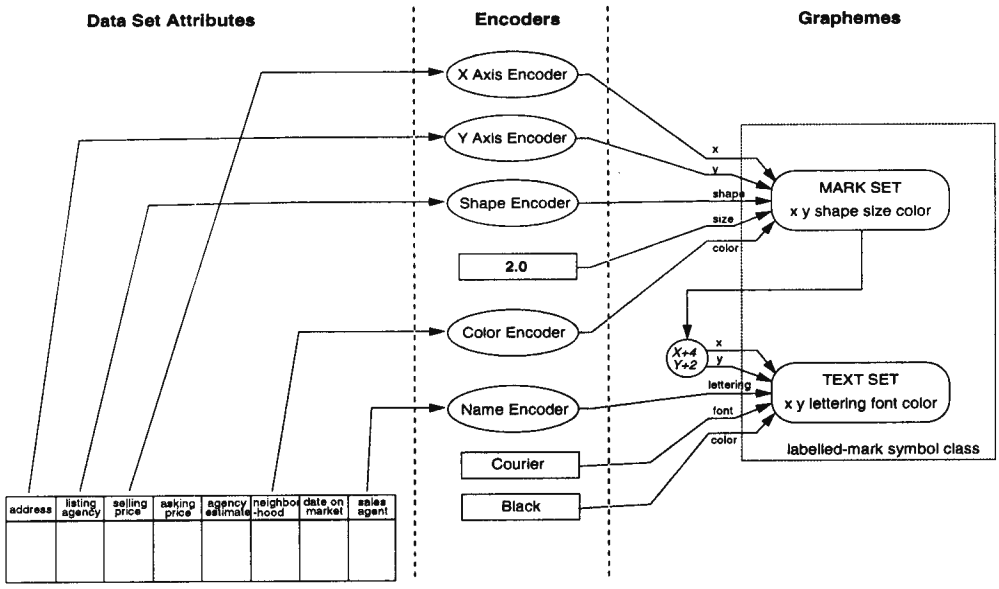


Figure 4
Encoders used in mapping attributes to a *labeled-mark* in the next chart shown in Figure 3.

Symbol classes in SAGE can be either pre-defined (some of the more common ones, such as a labeled-mark have already been defined), or created by the system based on rules about combining different graphemes into clusters.

Encoders are used to relate specific data values and graphical values to each other. Horizontal/Vertical axes, color keys, size keys and shape keys are some of the different encoders available in SAGE. Each encoder class consists of a definition of the relation between a family of data set attributes and a particular graphical type. SAGE can then use this information to map data values to graphical values in designing a picture, and provide a frame of reference (e.g., axes, keys, etc.) that can be used to visually interpret specific values in the picture. For instance, a color encoder could map data values "less than 5" to the graphical value "blue" and others to "red." A schematic of the encoders used in the chart shown in Figure 3 is shown in Figure 4.

In addition to this knowledge about graphemes, symbols and encoders, SAGE also uses knowledge of the characteristics of data relevant to graphic design (Roth and Mattis, 1990, Roth and Hefley 1993), including knowledge of data types and scales of measurement (e.g., quantitative, interval, ordinal, or nominal data sets), structural relationships among data (e.g., the relation between the end-points of ranges or between the two coordinates of a 2-D geographic location), and the functional dependencies among

attributes in database relations (e.g., one:one, one:many, many:many). As we will show later, the latter is an important factor in selecting a high-level discourse strategy for generating explanatory captions.

Finally, SAGE has a library of graphical techniques, knowledge of the appropriateness of the techniques for different data and tasks, and design knowledge for assembling these techniques into composites that can integrate information in a single display. SAGE uses this graphic design knowledge together with the data characterization knowledge to generate displays of information.

To summarize, the portion of SAGE's knowledge base that is most relevant for generating explanatory captions is its graphical syntax and semantics. The syntax includes a definition of the graphical constituents that convey information: spaces (e.g., charts, maps, tables), (graphemes) (e.g., labels, marks, bars) , their properties (e.g. color, shape), and encoders--the frames of reference that enable their properties to be interpreted/translated back to data values--(e.g., axes, graphical keys.). The syntax also defines the ways in which graphemes can be combined to form symbols--composites that integrate multiple data attributes (e.g., a label attached to a mark). The syntactic structure of a graphical display, like the linguistic structure of text, can provide guidance for creating structurally coherent explanations.

The representation of the *semantics* of graphics conveys the way data is mapped to the syntactic elements of displays. It also provides guidance for organizing explanatory captions by grouping graphical elements that express data attributes that form a coherent group. The data characterization provides knowledge of the structure of the data and therefore also influences the structure of the explanation.

3. Discourse Strategies for Generating Captions

Explanations about informational graphics can be classified into at least three categories based on the structural properties of the picture, the structure of the underlying data attributes, and their mapping to spaces and graphemes. These explanation strategies reflect the overall structure of the graphic presentation: whether the spaces are aligned along a common axis, and around the functionally independent attribute (FIA). An attribute is functionally independent if it uniquely determines the values of all other attributes. For example, in one of our current datasets about house sales, the house's street address has been specified as the (FIA); it uniquely determines the asking price, selling price, and the other attributes in the database. In contrast, the listing agency does not uniquely determine any of the other attributes in the house-sales relation.

In addition to the factors mentioned above--used to select the overarching discourse strategies--the system also makes use of additional information about the symbols and their mappings used in the display to select and organize information to be presented in the caption. For instance, the system uses graphical

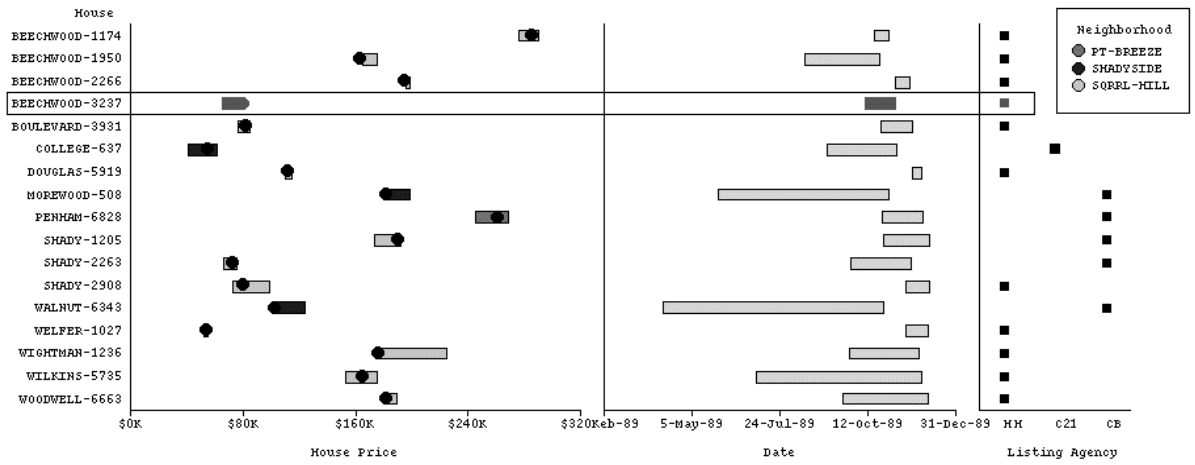
> These three charts show information about houses from data set PGH-23. Each chart has two axes. The Y-axis identifies the houses in the three charts. The data set contains 18 items. The X-axis in the first chart indicates house prices. The origin is at zero and there are five ticks on the axis, with the minimum value being $44,000. The difference between each tick is $110,000. The values mapped to the axis range from $55,000 to $399,000. The left edge of the bar shows the asking price of a house. Selling prices shown range from $55,00 to $387,000. Asking prices range from $61,000 to $399,000. The horizontal position of the square mark shows the agency estimate. These range from $55,000 to $387,000. For example…

Figure 5
A fragment of one possible verbose caption for the graphic in Figure 6.

information to determine the order in which information is presented. This reasoning can occur at various levels of the picture representation: at the space level (all objects in a space are described before objects in another space), at the grapheme cluster level (all objects in a cluster are described together) and at the encoder level (all objects that map the same attribute type are described together). SAGE 's representation of the graphical display thus provides additional information that can be considered when text explanations are generated.

The process of generating natural language explanations can be divided into three conceptual stages: (i) select a discourse strategy to provide the overall organization of the explanation based on the structural properties of the graphical presentation, the relations expressed in the dataset and the data to grapheme mappings, (ii) within each space of the presentation, use the complexity metric to determine the amount of detail to be included in the explanation, and (iii) reason about the tactical decisions in sentence planning.

In our current application, content selection mainly consists of determining the complex or ambiguous aspects of a graphic presentation. In general, knowledge based systems cannot afford to generate a paraphrase of the entire knowledge base. As illustrated by Figure 5, an explanation that includes all the facts in the underlying picture representation or data set for even a simple graphic in SAGE would be extremely verbose. Most of the facts expressed in such a caption would be both obvious and unnecessary for the average user. Studies have shown approximately three-fourths of the time spent by users in interpreting a graphic is used in understanding the data to grapheme mappings (Shah, 1995; Cleveland and



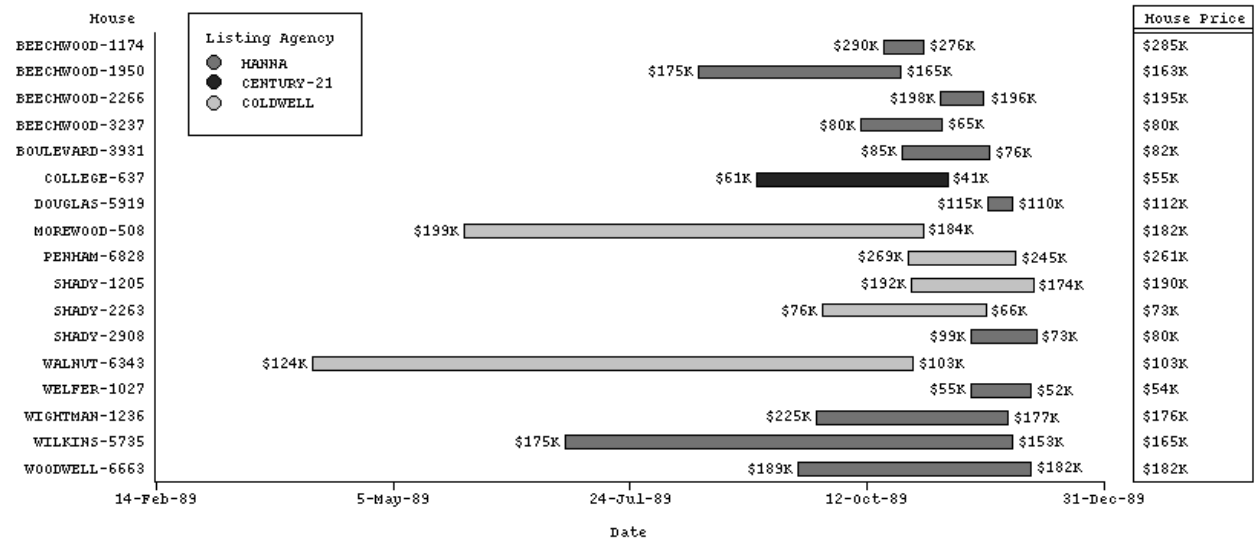
These three charts show information about houses from data set PGH-23. The Y-axis identifies the houses in the three charts. In the first chart, house prices are shown by the X axis. The house's selling price is shown by the left edge of the bar, whereas the asking price is shown by the right edge. The horizontal position of the mark shows the agency estimate. For example, as shown in the highlighted tuple, the asking price of 3237 Beechwood is $82K, its selling price is $75K, and the agency estimate is $81K. In the second chart, the house's date on the market is shown by the left edge of a bar, whereas date sole is shown by the right edge. Color indicates the neighborhood. The third chart shows the listing agency.

Figure 6
Graphic with the caption generated using strategy 1.

McGill, 1987). Therefore, our initial goal was to generate captions describing only those mappings that might be either complex or ambiguous for the average user. The system can currently analyze a picture representation for five different types of complexities and ambiguities; these are discussed in greater detail in the following section (Section 4). This section discusses the three strategies used by the system to structure the content during text planning. The sentence planning phase is discussed in Section 5, where the individual components implementing the tactical decisions in the micro-planner are described in detail.

3.1 Strategy 1: Graphic organized around the functionally independent attribute

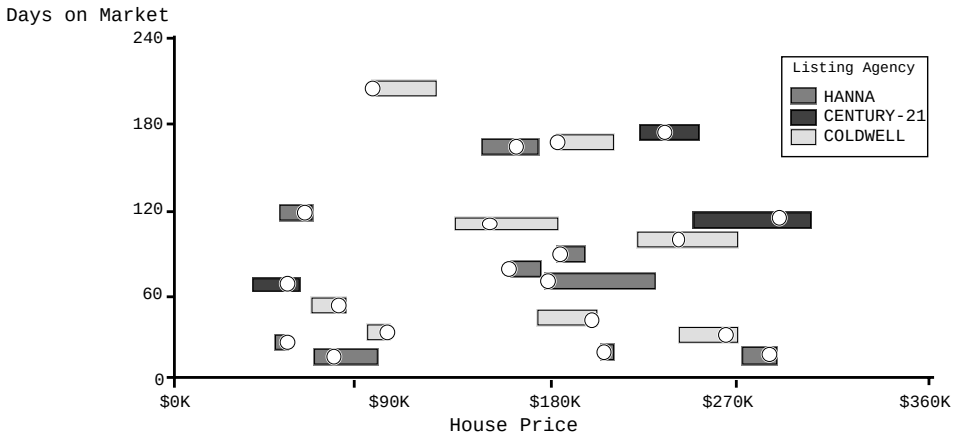
As mentioned earlier, the three strategies used by the caption generator depend upon both the structure of the graphic presentation and the relations in the data set presented in the graphic. The first strategy can be applied when the data set contains a functionally independent attribute (FIA) that is used as an organizing device or "anchor" for the entire graphic. This occurs either when the graphic has only one space and the FIA is mapped to one of the axes, or when there are multiple spaces and the FIA is mapped to the axis of alignment.



This chart and table show information about house sales from data set PGH-23. The Y-axis identifies the houses in the two spaces. In the chart, dates are shown along the X-axis. The house's date on the market is shown by the left edge of a bar, whereas the date sold is shown by the right edge. Color indicates the listing agency. The label to the left of a bar indicates the asking price, whereas the label to the right indicates the selling price. The table shows the agency estimate.

Figure 7
Caption for an alternative presentation of the dataset used in Figure 6.

In such cases, the strategy attempts to reinforce the organizing role of the functionally independent attribute. The explanation strategy identifies the anchor and the independent attribute first. Then, it describes each space in the picture relative to the anchor. Domain attributes mapped in the graphic are also mentioned in the context of the FIA and the type of relationship defined between them (one:one or one:many). Two SAGE generated graphics and the associated explanations that illustrate this organizing principle are shown in Figures 6 and 7. These two figures illustrate the importance of a caption generator in this application. Both figures present the same data set about house sales. However, the presentations generated by SAGE are different, make use of different mappings, and give rise to different perceptual complexities. Consequently, the content of the captions generated is also different. However, in both the captions, the overall discourse strategy is the same: to emphasize the aligning Y-axis, the functionally independent attribute--the *house-address*, and structure the description of the other attributes in terms of the FIA.

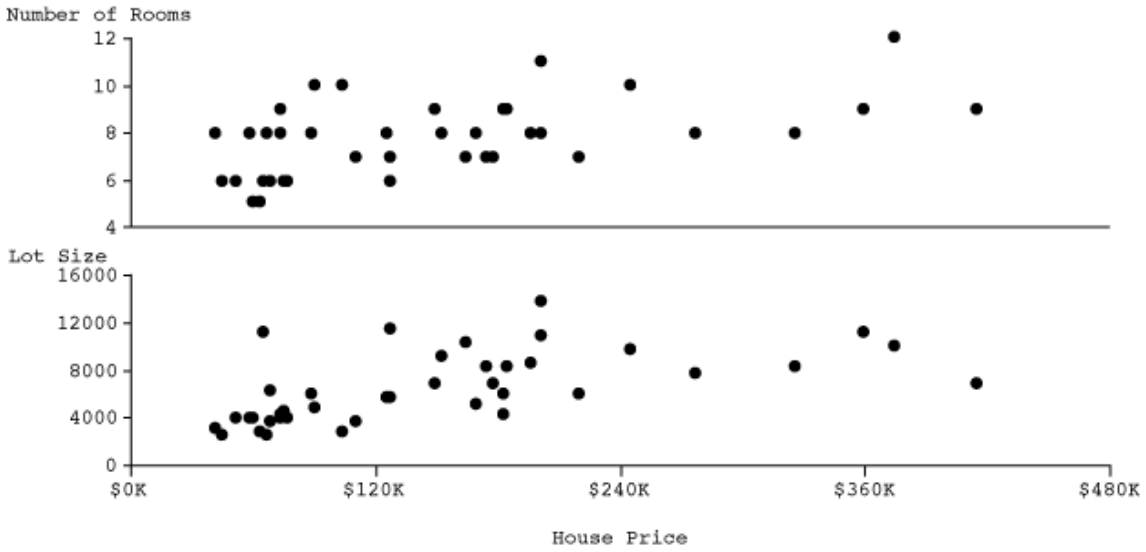


This chart and table show information about house sales from data set PGH-23. It emphasizes the relationship between house prices and the number of days on the market. The X-axis shows the house prices, whereas the Y-axis shows the house's number of days on the market. The house's listing agency is indicated by color. The selling price is shown by the left edge of the bar, whereas the asking price is shown by the right edge. The position of the mark shows the agency estimate.

Figure 8
Graphic with caption generated using strategy 2.

3.2 Strategy 2: Single space organized around dependent attributes

However, in cases where the graphic is organized around dependent attributes, the explanation cannot be structured around any of them. This is because the attribute may be defined in either one:many or many:many relationships in the dataset and cannot therefore be used as an identifier. This is the case in Figures 8 and 9. In these two figures, the attributes that are mapped to the axes of the charts are dependent attributes such as *days-on-market, number-of-rooms* and *lot-size*. Neither of these can be used to refer to other attributes unambiguously. Thus, the discourse strategy cannot be the same as in the case where an FIA is mapped along one of the axes. Instead the explanation emphasizes the *relation* between the dependent attribute(s) that serve as organizer(s). There are two strategies depending on whether or not the figure consists of multiple spaces. If there is only a single space in the graphic, the explanation emphasizes the relation between the attributes encoded against the two axes. A SAGE generated graphic and the associated explanation that illustrates this organizing principle is shown in Figure 8. The caption generated for the figure illustrates how the strategy emphasizes the relationship between the attributes mapped along the axes. Figure 8 shows the relationship between the variation in house prices and the number of days a house is on the market in the data set.



These charts show information about house sales from data set PGH-23. In the two charts, the X-axis shows the selling prices. The top chart emphasizes the relationship between the number of rooms and the selling price. The bottom chart emphasizes the relationship between the lot size and the selling price.

Figure 9
Graphic with caption generated using strategy 3.

3.3 Strategy 3: Multiple spaces aligned along an axis with dependent attributes

The second strategy discussed above is only applicable if there is a single space in the presentation. However, SAGE is capable of designing presentations with multiple spaces that are aligned along dependent attributes in the data-set. In such cases, the explanation generator cannot describe all the concepts in the presentation using strategy #2. This is because if one of the spaces in the presentation happens to have the FIA mapped to its non-aligned axis, a description such as ("this space shows the (one:one) relationship between the <*FIA*> and <*attribute-2*>") would not be natural. In such cases, it is more natural to use strategy #1 to describe the mappings in that space. Therefore, strategy #3 allows the system to organize the caption *for each space* accordingly, depending upon whether the FIA is mapped along its non-aligned axis. Figure 9 shows such a graphic and the corresponding caption. The two charts in 9 are aligned along the X-axis, which is used to encode (house-price). In generating the captions for the two charts, the system describes each one independently, using either strategy #1 or #2, as appropriate. It describes the top one first (following the structure of the graphic) and then the bottom one. Each of them, in this case, is described using strategy #2 because they both have dependent attributes mapped along the axes.

4. Graphical Complexity: The Need for Clarification

In the previous section, we discussed three strategies used to organize the information to be presented. As mentioned earlier, it is important to select information about mappings based on either complexity or ambiguity if the caption is to be both succinct and informative. We have identified the following five types of graphical complexities that can make it difficult for a user to understand complex data to grapheme mappings.[6]

4.1 Encoder Complexity

Understanding the encoders used in designing a picture is necessary for users to be able to read data values shown in the picture. Encoders allow the user to map between graphical values and attribute values. Two examples of encoders are the axes (which allow users to map between positional values in the picture and data values along the axes), and graphical keys (these can illustrate mappings between variables such as size and shape and attribute values). Complexities can arise either (i) when an encoder is complex, or (ii) when an encoder mapping uses a scale that is complex.

Consider for instance, Figure 10. Among the encoders used in this picture are the X and Y axes which map positional information to house prices and house addresses respectively. In the chart shown here, the X axis does not have a zero origin (presumably in order to make the differences between the data items clearer by having more screen real estate to display a smaller range of data values). Because of this translation of the origin, it is no longer possible to conclude in this chart that a bar twice as long as another bar encodes a value twice as large (for instance, bars representing houses WALNUT-6343 and VERMONT-637 in 10). Both axis translation and truncation--to compress empty regions in quantitative data--can lead to false inferences. Similar decoding problems can occur with other encoding techniques as well, as when a quantitative attribute is mapped to the area of a circle or non-linear scales are used along axes.

A more complex example of encoding technique complexity can be seen in Figure 1. Saturation and color are combined in a single encoding technique to express temperature. Dark red indicates 100 degrees and dark blue indicates -40 degrees. As the color gets paler (less saturated) it indicates a less extreme temperature. For example, pale red (pink) indicates 65 degrees, while pale blue indicates -5 degrees. White indicates a transition point.[7] Thus both the frame of reference (the color-saturation key) and the technique are potentially

[6] It is clear that explicitly reasoning about individual users and maintaining a record of user capabilities would result in better individualized descriptions than by using a default user model as is done here. Our system actually makes use of simple user models but we will not discuss this issue here.

[7] Not only is the encoding technique complex, but the user must understand the conventions used--blue to the cooler side of the scale, red to the warmer.

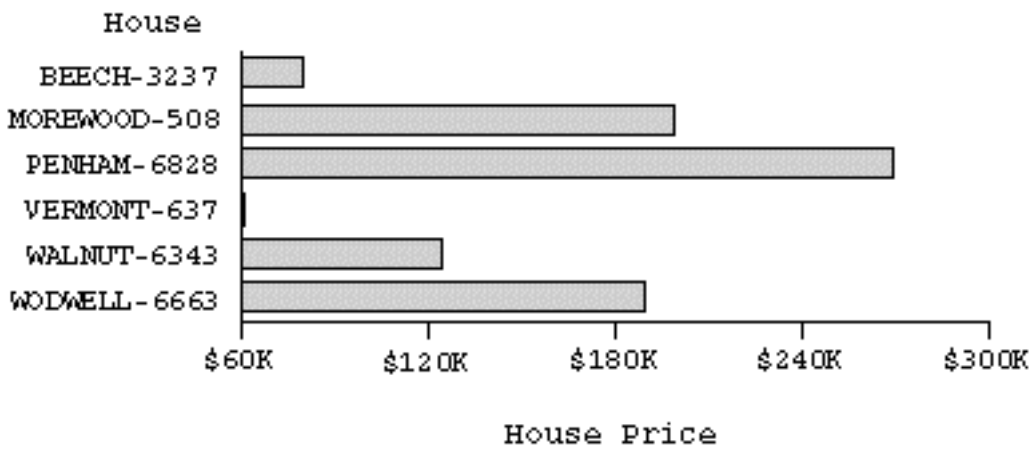


Figure 10

Comprehension can be hindered by encoding technique complexities (e.g., a truncated X-axis).

complex here. Figure 1 also illustrates *range complexity*: the user must determine what the transition point is (whether it is the center of the scale, or some special value, such as 32 degrees F). The graphic is not explicit about whether the two ranges on both sides of this special transition point are balanced.

4.2 Grapheme Complexity

Although the encoder (e.g., positional encoding on an axis) and the mapping (e.g., the scale used along the axis) may both be simple, a grapheme that uses that encoder and mapping may still be difficult for users to interpret. This may occur for a variety of reasons ranging from too many mappings to problems in identifying the mappings. Complexities of this type can arise from:

- **multiple grapheme properties:** In some cases, the presentations can include graphemes that have a large number of geometric properties used in mapping data attributes. Consider, for instance, Figure 11. While the encoders in the figure are relatively straightforward, the fact that four different mappings are used here--*x* position, *y* position, shape and color--can hinder comprehension.
- **unclear geometric properties:** Circular marks and horizontal bars are usually familiar to most readers and SAGE chooses them whenever possible. However, in some cases the system may have to use graphemes that are not as common. In such cases, the reader has to not only understand the encoder and the mapping technique, but also understand *which* property of the grapheme is being used in each encoding. Consider, for instance, if a triangular mark is used in a plot chart: in order to interpret its positional property, it is essential to know which of its three vertices (or the center) is used in the mapping.)

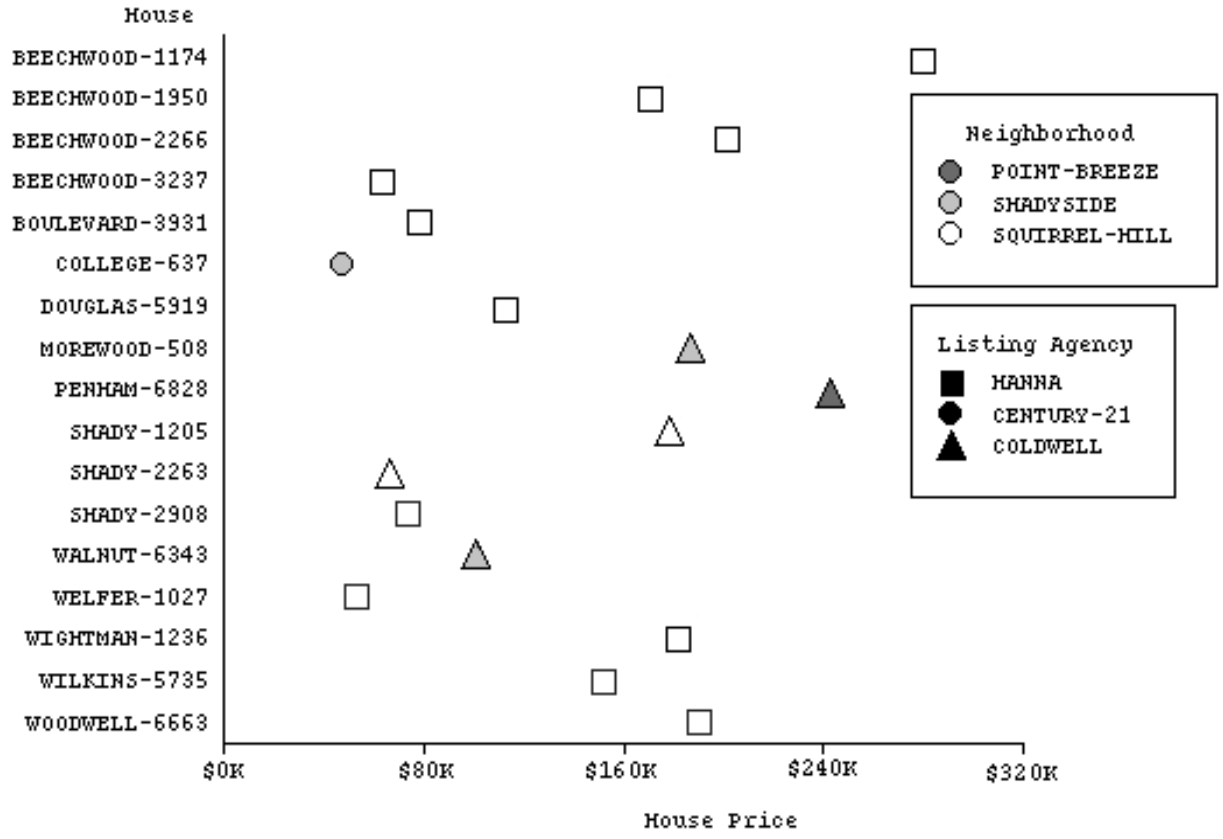


FIGURE 11
Comprehension difficulties can result from complex graphemes with multiple properties being used in the encoding.

- **semantic properties:** The third type of grapheme complexity occurs in graphemes that have sub-components. For instance if an icon of a truck were to be used as a grapheme, and different sub-components were used in the mappings (e.g., speed of the truck to the wheel size, cargo type to tank color), the reader must understand not only the various data to grapheme mappings, but also the relationship between the various sub-components.)

4.3 Ambiguous Mapping Complexity

A user's ability to identify the mapping of even simple techniques can be hindered when dissimilar graphemes (or dissimilar properties of a grapheme) are used to map to similar attribute types. Consider for instance, the charts in Figures 12 and 13. The left and right edges of the bar in 12 refer to the *selling-price*

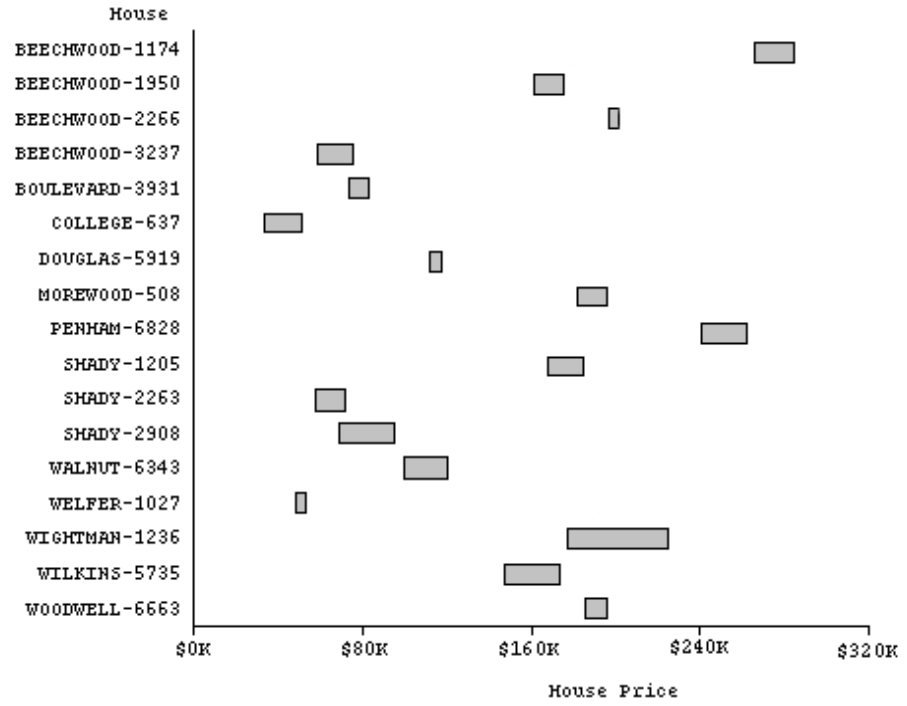


Figure 12
Complexities can arise from ambiguous mappings (a).

and *asking-price* of a house in the domain. However, the X axis represents *prices* in general, and there is no way to distinguish between the two from the figure itself. Similarly, in Figure 13, the two text labels refer to two different prices, but the two attributes cannot be distinguished from one another solely from the figure.[8]

4.4 Composition Complexity

When multiple graphemes occur in a space, they can be confusing at first until their relationship to each other are clarified. Compositions can result in clusters of two types:

- **Cooperative Graphemes:** For example, consider the chart shown in Figure 14. The mark and label graphemes form an aggregate that must be considered together. In this case, since the label conveying the real estate agency is slightly offset from the position on the X and Y axes, it cannot be interpreted as being related to a particular house and a date of sale on its own. Grapheme composition results in multiple graphemes being displayed as a spatially grouped conceptual unit--these need to be understood as such and interpreted accordingly.

[8] In the housing domain, it may be assumed that *asking-price* is either greater or equal to the *selling-price*, but in fact, this is not always the case. Buyers sometimes get into bidding wars that cause the *selling-price* to become greater than the *asking-price*.

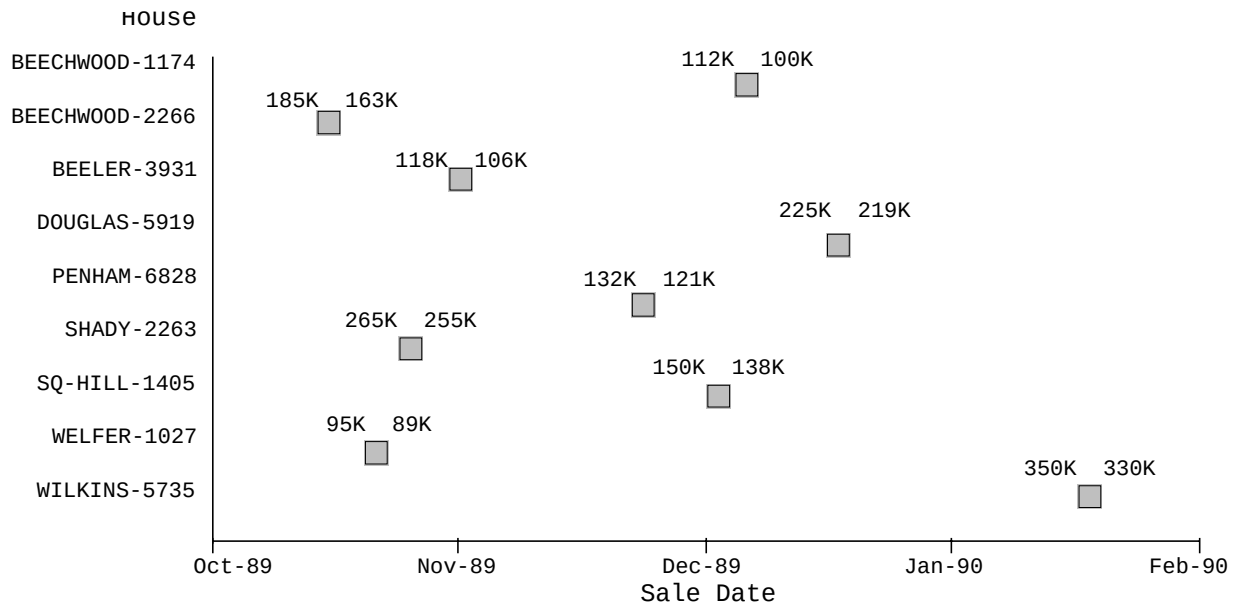


Figure 13
Complexities can arise from ambiguous mappings

- **Interfering Graphemes:** Unfortunately, grapheme composition does not always result in a cluster where the graphemes are distinct and non-occluding. Consider, for instance, the chart shown in 8. The mark indicating the agency estimate of the selling price often overlaps with the interval bar showing the actual asking and selling prices. In some cases, the asking and selling prices are so close that the mark indicating the agency estimate actually occludes the interval bar. Clusters such as this can hinder interpretation and it is important that such mappings be clarified.

4.5 Alignment Complexity

As illustrated in Figures 6, 7, and 9, alignment of multiple charts and/or tables can be a useful technique for supporting comparisons, rapid lookups for many attributes of the same object, and for maintaining consistent scales. Whenever an alignment occurs, all but one of the charts become separated from the aligning axis labels and the relation between the aligned axis and the rest of the charts may not be clear.

The complexity assessment module in the system is capable of identifying the graphemes in the display that are complex for any of the five reasons described in this section. It annotates the picture representation generated by SAGE to indicate the graphemes and their types of complexity. The result of the complexity assessment for the Minard graphic—Figure 1—is shown in Figure 15. As discussed earlier, for instance, the mapping between the attribute *temperature* and the color of the line is complex for two

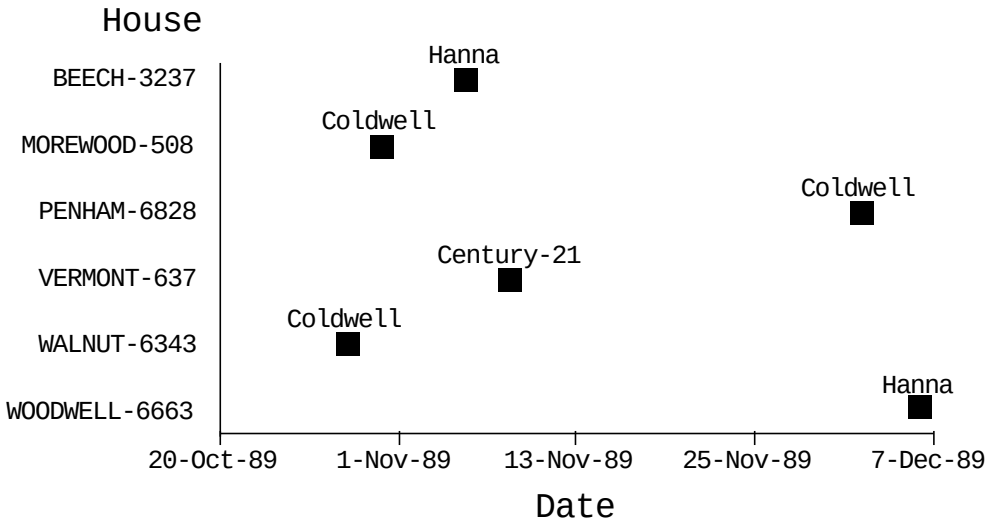


Figure 14
Presentations can have clusters of *cooperative* graphemes

reasons: (i) encoding complexity, because of the use of color and saturation, and (ii) range complexity, because of the unequal distributions of warm and cold temperatures. Figure 16 gives the complexity assignment for the graphic shown in Figure 6. In this case, the mapping between the attribute *asking price* and the bar is complex for three reasons: (i) grapheme complexity, since the interval bar is a complex grapheme (ii) ambiguous mapping, since from the graphic, it is not possible to determine whether the attribute is mapped to the left edge or the right edge of the bar, and (iii) composition complexity, since the bar and the mark can overlap and occlude each other (as indicated by the "*i*" for interfering). The annotated picture representation can then be used as one of the knowledge sources in the NLG system to select and structure information appropriately in generating the captions.

5. Generating Explanatory Captions

A high level overview of the system divided into functional modules is shown in Figure 17. A brief description of each module is given below. Detailed descriptions follow later in the section.

Data attribute	Graphical element	Complexity type
temperature	grapheme, property	encoding complexity range complexity
troop size	grapheme, property	grapheme complexity composition complexity (i)
start-position	grapheme, property	grapheme complexity composition complexity (i)
stop-position	grapheme, property	grapheme complexity composition complexity (i)
battle city	grapheme, property	composition complexity (c) ambiguous mapping
battle date	grapheme, property	composition complexity (c) ambiguous mapping
battle location	grapheme	composition complexity (c)

Figure 15
Result of the complexity assignment module for the "Minard Graphic" in Figure 1 ("I" and "c" are used to indicate interfering and cooperating graphemes respectively).

Data attribute	Graphical element	Complexity type
house	grapheme, property	alignment complexity
asking price	grapheme, property	grapheme complexity ambiguous mapping composition complexity (i)
selling price	grapheme, property	grapheme complexity ambiguous mapping composition complexity (i)
agency estimate	grapheme, property	composition complexity (i)
date on market	grapheme, property	grapheme complexity (c) ambiguous mapping
date sold	grapheme, property	grapheme complexity (c) ambiguous mapping
listing agency	grapheme	

Figure 16
Result of the complexity assignment module for Figure 6.

Text Planning Module: The text planner takes as input the goal to generate a caption, the picture representation generated by SAGE (annotated by the complexity module), and generates a partially ordered text plan. The leaves of the text plan represent speech acts about propositions that need to be conveyed.

Ordering Module: The ordering module takes a partially ordered text plan and imposes a total order on the speech acts. This may be based on (i) domain specific knowledge about orderings (for instance, knowledge about temporal order of events), or in the absence of this, (ii) knowledge about graphics (e.g., the left edge of a bar is discussed before the right edge of a bar).

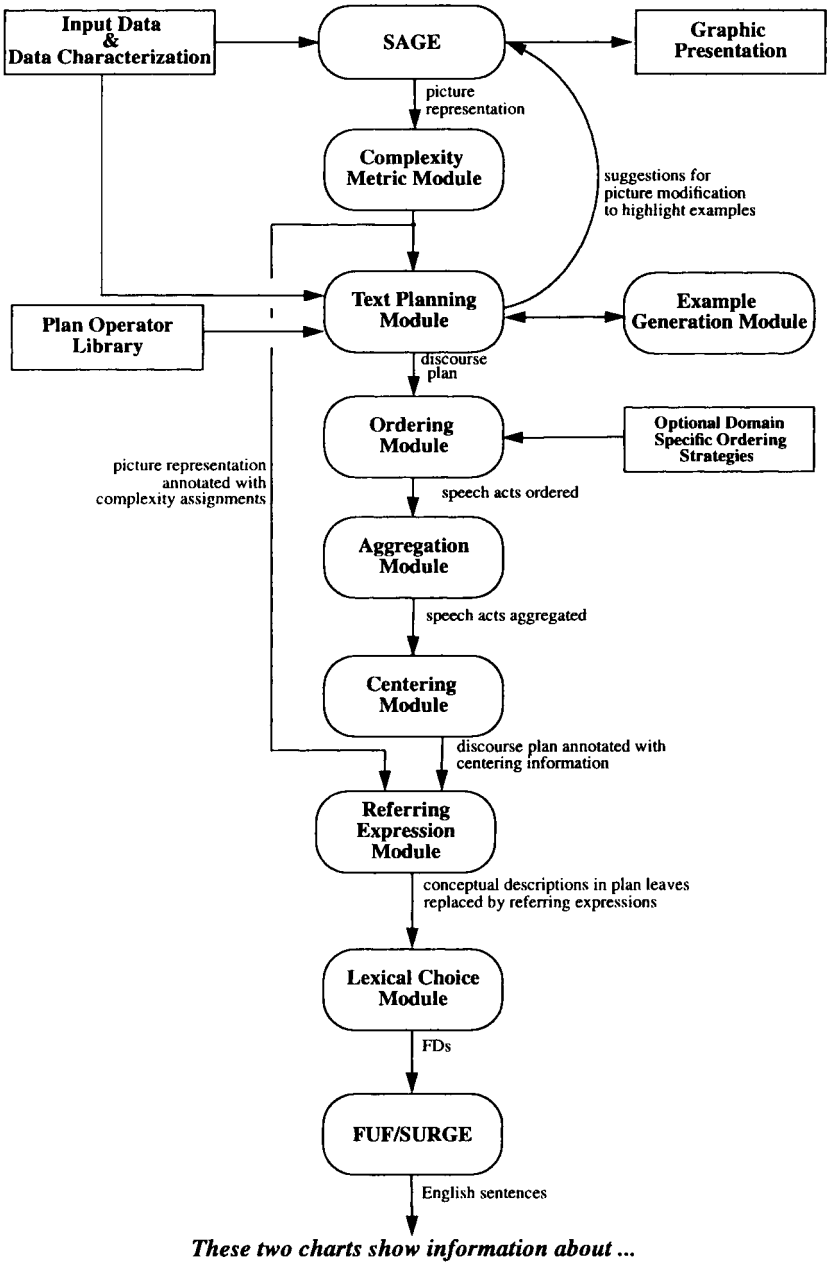


Figure 17
System architecture: A functional-block diagram

Aggregation Module: The output of the ordering module is passed to an aggregation module that can combine multiple propositions into fewer, more complex ones. For instance, the module may combine some propositions regarding a grapheme into one complex proposition for more natural output.

Centering Module: Once clauses are ordered and aggregated, coherence of the generated text can be further improved by selecting appropriate orderings between arguments of each clause. For this task, we have developed a selection strategy based on the centering model.

Referring Expression Module: The referring expression module analyzes the picture representation and uses the discourse plan to determine appropriate referring expressions for the concepts in the speech acts.

Lexical Choice and Realization Modules: This lexical choice module picks lexical items and transforms the speech acts to functional descriptors (FDs) to be processed by FUF/SURGE (Elhadad and Robin, 1992; Elhadad, 1992), the realization module used to generate the English text.

5.1 Text Planning Module

The planner constructs text plans from its library of discourse action descriptions. The representation of communicative action is separated into two types of operators: action operators and decomposition operators. Action operators capture the conditions (preconditions and constraints) under which an action can be executed, and the effects the action achieves if executed under the appropriate conditions. Preconditions specify conditions that the agent should plan to achieve (e.g., the hearer knows a certain term), while constraints specify conditions that the agent should not attempt to plan to change (e.g., facts and rules about the domain). Effects describe the changes that a discourse action is intended to have on the hearer's mental state. If an action is composite, there must be at least one decomposition operator indicating how to break the action down into more primitive steps. Each decomposition operator provides a partial specification for a sub-plan that can achieve the action's effects, provided the preconditions are true at the time the steps in the decomposition are executed.

As an example of how action and decomposition operators are used to encode discourse actions, consider the two operators in Figure 18. These two operators describe the discourse action `describe-space-mappings`, whose only effect is achieving the state in which the reader knows all the data to grapheme mappings shown. The first operator is an action operator and it indicates that `describe-space-mappings` can be used to achieve the state where the reader knows about the mappings.

```
(define    (action describe-space-mappings)
           :description "describe all mappings in a space"
           :parameters (?space)
           :primitive nil
           :effect ((know-all-mappings ?space)))
```

```
(define    (decomposition describe-space-mappings)
           :description "Describe mappings: +C+I+V"
           :constraints ((space-p ?space)
                         (single-space? ?space)
                         (get-interfacing-graphemes ?space ?int-graphs)
                         (get-cooperating-graphemes ?space ?coop-graphs)
                         (get-vanilla-graphemes ?space ?vanilla graphs)
                         (get anchor ?space ?anchor-axis ?anchor-domain))
           :preconditions ((recognize-space ?space)
                           (know-mapping ?space ?anchor-axis ?anchor-domain))
           :steps ((begin    (start ?space))
                   (end      (finish ?space)))
           :rewrites (((know-all-mappings ?space)
                       ((forall ?ig in ?int-graphs
                                (know-all-interfering-mappings ?ig ?space))
                        (forall ?cg in ?coop-graphs
                                (know-all-cooperating-mappings ?cg ?space))
                        (forall ?vg in ?vanilla-graphs
                                (know-all-vanilla-mappings ?vg ?space))))))
```

Figure 18
Sample Plan Operators.

The second operator in Figure 18 is one of the decomposition operators for the `describe-space-mappings` action. The decomposition of a non-primitive action can be expressed either in terms of sub-actions (`:steps` slot), or in terms of sub-goals of one action's effect `:rewrite` slot), or in terms of both. For instance, the `:rewrite` slot of the decomposition in Figure 18 specifies that one way to achieve `describe-space-mappings` 's effect of having the hearer to know all the mappings in one space is to achieve the three sub-goals of having the hearer to know all the interfering, cooperating and vanilla[9] mappings in that space. This example also illustrates how the graphical complexity metrics are used for content selection by the text planner: just as this operator can be used to describe spaces in which all three types of graphemes are present, there are other operators which deal specifically with encoder complexities, compositional complexities, etc.

As illustrated by the second operator in Figure 18, decomposition operators may also have constraints, which indicate the conditions under which the decomposition may be applied. Such constraints often specify the type of information needed for particular communicative strategies, and satisfying them causes the planner to find content to be included in explanations. For example, the constraints of the second

[9] Vanilla graphemes are those that are neither interfering nor cooperating.

operator not only check that a single space is being described, but also find the graphemes of the three types used in the explanation, and the *anchor-mapping* in this space. When the planner attempts to use a decomposition operator, it must try to satisfy all of its constraints. If a constraint contains no unbound variables, it is simply checked against the knowledge source to which it refers. However, if the constraint contains free variables (e.g., `?int-graphs` in the second operator), the system must search its knowledge bases for acceptable bindings for these variables. In this way, satisfying constraints directs the planner to select appropriate content to include in explanations. In the case of the operator shown in Figure 18, the two preconditions that must be satisfied are (i) that the reader must be able to *recognize* the space (i.e., know which space is being discussed, and the data set being visualized), and (ii) know what the anchor mapping in the space is (if any). Anchor mappings refer to the mapping between a functionally independent attribute (FIA)--usually the *key* in the database schema--and the axis it is mapped to. Thus, action and decomposition operators specify how information can be combined in a discourse to achieve effects on the hearer's mental state.

5.1.1 Generating Discourse Plans Planning begins when a set of communicative goals are posted to the text planner. The system generates a plan by iterating through a loop that refines the current plan (either decompositionally or causally), checking the plan after each refinement to ensure that it has not introduced any errors. Decompositional refinement selects a composite action and creates a sub-plan for that action by adding instances of the steps listed in the decomposition operator to the current plan. Causal refinement selects an unsatisfied precondition of a step in the plan and adds a causal link to establish the needed condition. This is done either by finding a step already in the plan that achieves the appropriate effect, or by using an action operator to create a new step that achieves the needed condition as one of its effects. For a complete definition of the algorithm, its computational properties, and its utility for discourse planning, see (Young, Pollack and Moore, 1994; Young and Moore, 1994).

In the remainder of the section, we present the modules that follow the text planning process and implement tactical decisions. To clarify the discussion, we describe how each module contributes to the generation of clauses (3) to (5) in the sample caption shown in Figure 19.

Sample caption

(1) This chart presents information about house sales from data-set TS-2480.
(2) The y-axis shows the houses.

(3) The house's selling price is shown by the left edge of the bar
(4) whereas the asking price is shown by the right edge.
(5) The horizontal position of the mark shows the agency estimate.)

Figure 19
A representative caption used to illustrate our discussions.

5.2 Ordering Module

The steps in a completed text plan are partially ordered, and thus further processing must be performed in order to generate a caption. The order of execution of steps in the plan may either be explicitly specified by the operator writer or may have constraints imposed on them by causal links. For instance, in the plan operator shown in Figure 18, all the steps corresponding to the goal `recognize-space` will be ordered before the steps corresponding to the goal `know-all-mappings` because `recognize-space` is a precondition. However, most steps in the plan are not explicitly ordered and do not have causal links between them dictating the ordering. The ordering module takes as input the discourse plan, with links specifying the ordering relations between sub-trees, and orders the leaf nodes--the speech acts--based on a set of heuristics. In our application, for instance, unless otherwise indicated, the system will describe the "left edge of the bar" before the "right edge."[10]

The ordering module sorts first based on the *space* ordering. This is based on the assumption that in the absence of any other discourse strategy (such as the need to emphasize or compare properties of a concept across multiple spaces), the reader will browse the spaces from left to right. After the plan steps have been sorted on a space-by-space basis, the module sorts plan steps based on their *graphical mappings*, using the following ordering heuristics:

position > *color* > *shape* > *size* > *text* > *others*

Finally, within each resulting subset, the module orders steps by *grapheme type* using the following ordering:

line\ set > *bar\ set* > *mark\ set* > *text\ set* > *others*

The strategy of ordering first by *graphical\ mapping* and then by *grapheme\ type* is based on our analysis of hand generated captions. We found that most captions tended to be structured along the mappings rather than along the graphemes.

Let us now examine how the system's ordering rules determine the ordering among clauses 3-5 of the sample caption shown in Figure 19. First, clauses 3-5 are grouped together because they are all mappings to *position.* Second, clauses 3-4 precede clause 5 because *bar\ set* must precede *mark\ set*. Finally, clause 3

[10] This is conventional for languages which are written from left to right, and may be different in other languages that are written from right to left.

precedes clause 4, because of the conventional preference for left to right ordering between edges of floating bars.

So far, we have examined the ordering strategy that the system will follow by default. However, the ordering module can also take an optional input, a functional specification, which can be used to determine plan step orderings that do not conform with the default ordering. Using this optional specification, the system can take advantage of domain knowledge, such as temporal sequencing, which can play an important role in discourse sequence. For instance, in general it may be preferable to state the mappings of the left and right edges of a bar in that order. However, if the left edge of a bar indicates a house's *selling-price* and the right edge indicates the house's *asking-price*, the usual temporal ordering between the events would suggest that one discuss the asking-price *before* the selling-price, which in turn would lead to mentioning the right edge before the left edge, contrary to the default ordering.

5.3 Aggregation Module

Once the speech acts are ordered they are passed to the aggregation module. In the general case, aggregation in natural language is a very difficult problem (Dalianis, 1996; Shaw 1995; Huang and Fiedler, 1996). Fortunately, our generation task requires a type of aggregation that is relatively straightforward. Our aggregation strategy only conjoins pairs of contiguous propositions about the same grapheme type in the same space. The module checks for grapheme (types) rather than specific graphemes to cover circumstances where, for instance, a chart may have a number of grey and black bars (which are different graphemes of the same type). This enables the system to generate text of the form "*The grey bars indicate the selling price of the house, whereas the black bars indicate the asking price.*"

When two propositions are combinable, namely they are about the same grapheme type in the same space, the system checks to see if the two properties being discussed are *contrastive* in some way. For instance, whether the two properties under consideration are the opposite edges of a bar, or are the X and Y axes, etc. If so, the system picks a contrastive cue phrase (e.g., "whereas") to merge the clauses resulting from the two propositions, otherwise the system picks the cue phrase "and."

Let us now briefly examine how aggregation affected clauses 3-5 of the sample caption in Figure 19. Clauses 3-4 were conjoined because they are about the same grapheme type, a horizontal bar, in the same space. Moreover, the module placed a "whereas" cue phrase between the two clauses, because the opposite edges of a bar are considered contrastive properties.

5.4 Centering Module

Once clauses are ordered and aggregated, coherence of the generated text can be further improved by selecting appropriate orderings between arguments of each clause. For this task, we have developed a selection strategy based on the centering model. Focus (e.g., (Sidner, 1979, Grosz, 1997)) and centering (e.g., (Grosz, Joshi and Weinstein 1995)) models are attempts at explaining linguistic and attentional factors that contribute to local coherence among utterances. Although focus and centering models were originally developed as foundations for understanding systems, they have frequently been proposed as effective knowledge sources for NLG systems. In particular, for (generating referring expressions) (including pronominalization) (see (Dale, 1992; Appelt, 1985; Maybury, 1991)), for (deciding when to combine clauses) (subordination and aggregation) (see (Derr and McKeown, 1984), and finally for choosing appropriate inter/intra clause orderings, namely, ordering between clauses and between their arguments (see (Maybury, 1991; Hovy and McCoy 1889; McKeown, 1985)).

Details on centering theory and its relation to discourse structure can be found in (Grosz, Joshi and Weinstein 1995, Walker 1993, Walker, Iida, and Cote, 1994, Grosz and Sidner, 1993, Gordon, Grosz, and Gilliom 1993); for lack of space in this paper, we only provide a minimal introduction to the basic terminology of centering theory.

Centers are semantic objects (not words, phrases, or syntactic forms) that link an utterance to other utterances in the same discourse segment. Centering theory provides definitions for three different centers, and for four possible center transitions between two adjacent utterances. It also states two fundamental constraints on center movement and realization.

Basic Center Definitions[11]

- *Cf(U):* The set of forward-looking centers, which contains all the entities that can link the current utterance to the following one. It is not constrained by features of previous utterances. Elements of *Cf(U* are ordered; the major determinant of the ranking on the *Cf(U)* is grammatical role with *subject* > *object* > *others*[12].
- *Cp(U)*: Highest ranking element of *Cf(U*)

[11] The following functions Cf(U), Cp(U) and Cb(U) apply to a particular utterance *U* and a particular discourse segment *DS.* As in(Grosz, Joshi and Weinstein 1995), we assume *DS* fixed and we drop it as an argument. Whenever the utterance argument *U* is not critical we drop it too.

[12] Other factors influence the ranking of *Cf(U* elements. The effects of word ordering, clausal subordination and lexical semantics are currently under investigation by other researchers.)

- *Cb(U):* The backward-looking center (unique) is the highest ranking $Cf(U_i\text{-}1)$ realized in the current utterance U_L. *Cb(U)* is a discourse construct, therefore the same utterance in different discourse segments may have a different *Cb*.

Center Transitions: The four possible center transitions across pairs of utterances are shown in Table 1.

	$Cb(U_i)=Cb(U_{i-1})$	$Cb(U_i) \neq Cb(U_{I-1})$
$Cb(U_i)=(Cp(U_i)$	Continue	Smooth-Shift
$Cb(U_i) \neq Cp(U_i)$	Retain	Rough-Shift

Table 1
Center transitions

The central tenet in centering theory is that discourse coherence of a text span increases (and a reader's cognitive load decreases) proportionately to the extent that discourse within the span follows two fundamental centering constraints (Grosz, Joshi and Weinstein 1995). These are:

Constraint on realization: If any element in the set of forward looking centers of an utterance (U_i) is realized by a pronoun in the following utterance ($U_i + 1$), then the backward looking center of the following utterance ($U_I + 1$) must also be realized by a pronoun.

Constraint on movement:(i.e. centering transitions) Sequences of CONTINUATIONS are preferred over sequences of RETAININGs; and sequences of RETAININGs are preferred over sequences of SHIFTINGs (and consequently, smooth shifts are preferred over rough shifts).

Grosz and her colleagues suggest that a competent generation system should apply the constraint on movement by planning ahead in an attempt to minimize the number of SHIFTs in a locally coherent discourse segment (Grosz, Joshi and Weinstein 1995).

Our centering-based strategy implements this suggestion by selecting intra-clause orderings that enforce centering transitions consistent with a given discourse structure. The strategy is general and can be applied to any discourse structure, but to be effectively applied to the generation of captions some assumptions not supported in terms of centering theory must be made. The problem is that the NPs generated in the captions are often possessive and have complex syntactic structures (e.g., “the selling price of the house”, “the mark's horizontal position”) and centering theory is not yet clear on the determination of centers in complex syntactic structures such as possessives and subordinate clauses

(Grosz and Sidner 1993). To accommodate for this problem we made two assumptions. First, given possessives of the form "property of grapheme/entity", either the grapheme or the entity is the center, not their properties. Secondly, even when only a property (e.g selling-price, right edge) is mentioned, the corresponding entity or grapheme is the center.

Our centering strategy processes the ordered speech acts sequentially and assumes that text spans describing the mappings from properties of a grapheme to properties of an entity are locally coherent discourse segments. The strategy enforces the constraint on movement within each of these discourse segments by preferring a CONTINUATION or a SMOOTH-SHIFTS transition to a RETAIN or a ROUGH-SHIFT transition respectively. This is done by keeping the highest ranking forward-looking center of the first clause of the segment (which is either an entity or a grapheme), as the *Cp(U_(i))* of all the following clauses in the same segment. In this way, in all such clauses the *Cb(U_(i))* and the *Cp(U_(i))* will be the same and, according to Table 1, this corresponds to forcing either CONTINUATIONs or SMOOTH-SHIFTS.

Furthermore, the strategy applies an additional constraint on movement: between segments dealing with different graphemes, the strategy explicitly marks the segment boundaries by ROUGH-SHIFT, SMOOTH-SHIFTS and RETAIN over CONTINUATION. This case is not mentioned in (Grosz, Joshi and Weinstein 1995). However, since the system maintains local coherence in a segment by minimizing ROUGH-SHIFTS and RETAIN s, it seems intuitive to prefer ROUGH-SHIFTs and RETAIN s to emphasize the change at segment boundaries (i.e. the boundaries between such segments should be maximally incoherent). Thus, in the caption generation application, when a text span describing the mapping for a grapheme (a discourse segment) is followed by a description of a mapping for a *different* grapheme (another discourse segment), the centering strategy will try to force either a ROUGH-SHIFT or RETAIN to mark the segment boundary. This is done by moving the *Cb(U_(i))* of the clause following the boundary out of the clause front position. That is, if the grapheme is the *Cb(U_(i))*, the domain entity is placed in front of the clause, and vice versa in the other case.

For example, consider the effect of the centering strategy on clauses 3-5 of the sample caption shown in Figure 19. Since clauses 3-4 are about mappings from properties of the same grapheme--a horizontal bar--they are assumed to belong to the same discourse segment. Therefore, the system keeps the *Cp* of clause (4) equal to the *Cp* of clause (3) by placing the possessive "the house's asking price" in front of the clause. In contrast, since clauses (4) and (5) are about mappings from properties of different graphemes, a RETAIN centering transition was enforced (vs. a CONTINUATION) by moving the possessive corresponding to the *Cb*, "the house's agency estimate", out of the front position. Once

intra-clause orderings are determined by the centering strategy, the annotated speech acts are passed to the referring expression module.

5.5 Referring Expression Module

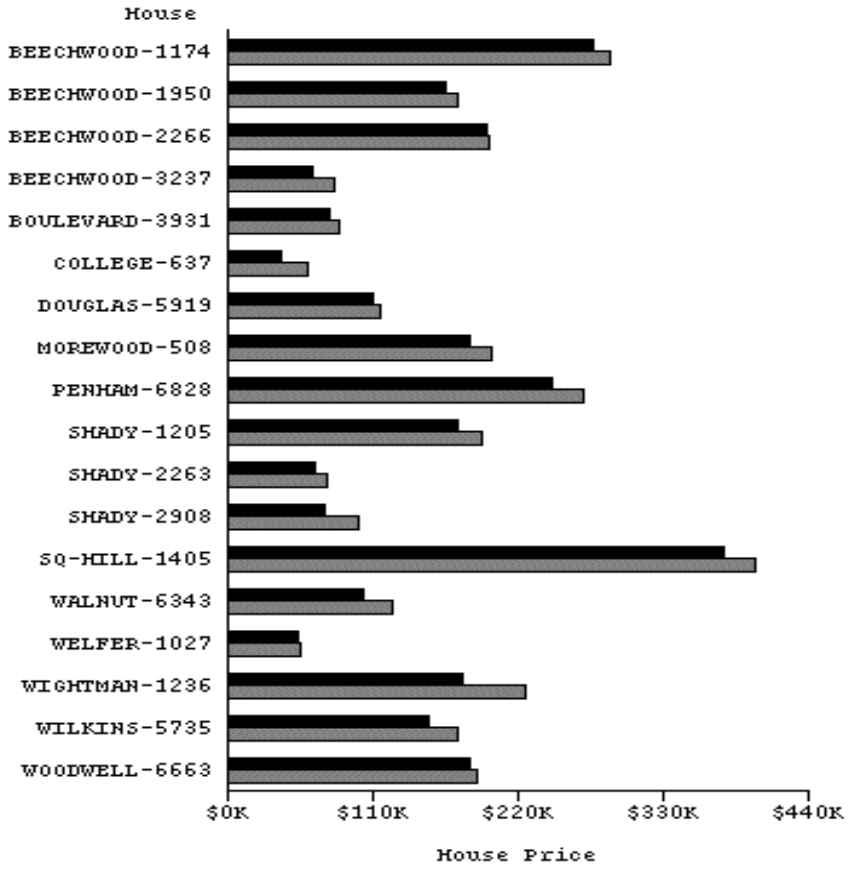
The referring expression module is largely based on the algorithm for *incremental interpretation* described in (Dale and Reiter, 1995). The incremental interpretation algorithm can generate appropriate referring expressions by incrementally constructing a set of attributes that uniquely identify the desired referent. These identifying attributes are selected based on a domain specific default ordering. In our case, the only referential problem is identifying the graphemes, and often the type of the grapheme (e.g., "bar") is sufficient to do so.[13] However, sometimes, a graphic may contain multiple graphemes of the same type. In such cases, the system must utilize additional perceptual properties (e.g., color, saturation, size, shape) to build an appropriate referring expression. For example, the referring expressions for the bars in the caption for the chart shown in Figure 20 use color as an additional identifying attribute.

Since our system generates multi-sentential captions, the referring expression module takes into account what is in focus at a given point in the discourse in order to generate concise and natural expressions. The referring expression module considers in focus all of the forward looking centers (i.e. *Cf*) computed by the centering module, and simply removes identifying attributes if they are in the *Cf* at that point in the discourse. This strategy results in the more concise rephrasing:

(3) The house's selling price is shown by the left edge of the bar (4) whereas the asking price is shown by the right edge. The horizontal position of the mark shows the agency estimate.

There are other forms of referring expression reduction due to discourse context that require a more sophisticated treatment. Hand written captions often radically simplify descriptions to express facts such as: "the third chart shows the neighborhood." However, the system generated caption would express the underlying proposition, based on the data to grapheme mappings, as*: "the position*

[13] For instance, we do not have to worry about issues such as implicatures conveyed by lexical choices or the use of non basic-level classes, since the set of objects and the available ways of referring to them in our context is so limited.



This chart presents informatin about house sales from data-set TS-1742. The Y-axis indicates the houses. The dark gray bar shows the house's selling price whereas the black bar shows the asking price.

Figure 20
The referring expression module uses color in this case to distinguish between the two types of bars and the attributes mapped to them.

of the mark in the third chart shows the neighborhood." The sequence of reductions shown below could achieve the more natural effect by repeatedly reasoning about the picture and the information being conveyed by each statement

the position of the mark in the third chart shows the neighborhood (1)

⇒ the mark in the third chart shows the neighborhood (2)

⇒ the third chart shows the neighborhood (3)

The system would need to realize that *position* was the only attribute of the mark being used for a mapping, and position is always clear in a graph and need not explicitly be mentioned; thus resulting in statement (2). However, since the mark is the only grapheme used in the graph, the system could leave off mentioning the mark as well, thus resulting in statement (3). There are two ways of dealing with this issue, (a) the system could apply iterative refinements of the referring expressions generated by the planner, as done in the *local brevity algorithm* (Reiter, 1990). However, this single case would have substantially increased the computational cost of generating referring expressions in all cases, without significantly improving any of the other (perfectly appropriate) referring expressions generated by the module, (b) the system could recognize this specific situation at a higher level and process the speech acts appropriately to avoid this situation completely. Thus, rather than considering this situation as a problem of generating an appropriate expression for the *concept position of the mark in the third chart,* we have chosen to push this problem up to the planner level during content selection. (Consequently, there are operators that look specifically for situations such as this--single grapheme in a space, mapping a single property--that are selected by the planner in such situations.) While this does tend to muddy the distinction between the "high-level" planner and the "lower-level" tactical processing--because the planner is now forced to deal with this one situation regarding referring expressions that should arguably be dealt with more properly by the referring-expression module--it does enable the system to generate appropriate texts with a simpler, more efficient approach in this application.

It should be noted that there is one additional type of referring expression that our system is capable of generating. This happens in situations when the graphic being explained is considered complex enough to require an example. In such cases, the system attempts to highlight the grapheme corresponding to the tuple being used in the example. There are a number of ways in which the relevant grapheme can be highlighted: with a graphical annotation, such as an arrow, a circle surrounding the grapheme, a change in color, etc.--with a corresponding number of ways in which the caption can then refer to the grapheme. This is similar to the approaches used for generating cross-modal references discussed in the context of the COMET (McKeown et al., l992) and WIP (André and Rist, 1994) projects. This will be illustrated in the next subsection which discusses the generation of examples.

5.6 Example Generation Module

If the text planner encounters particularly complex data to grapheme mappings, it can attempt to present an example to clarify the problematic mappings. Our current implementation is designed to trigger the example generation process in the case of interfering grapheme clusters where occlusion can hinder interpretation. Plan operators (described in Section 5.1) contain constraints that check for the appropriate conditions and establish goals for the generation of an example in the caption. In response, the example generation module selects a grapheme shown in the picture, finds the data values associated with the individual grapheme, and constructs an example that can be use by the text planner. Additionally, the example generator also posts a request to SAGE to highlight the relevant instance in the picture. If the highlighting request succeeds, the example generator annotates the example with this information and the resulting caption mentions the highlighted grapheme. Currently, this is the only case in which the caption generation mechanism can influence the graphic design. A caption fragment which includes an example is shown below:

> For example, as shown in the highlighted tuple, 3237 Beechwood Boulevard's asking price is 79900 dollars and its selling price is 65000 dollars. Its agency estimate is 79781.625 dollars. Its neighborhood is Squirrel Hill.

There are a number of issues relevant to the generation of captions that integrate examples and text (Mittal and Paris, 1992; Mittal and Paris, I993). We will not discuss them here in detail because the context in which our current system generates explanations is very restricted (as compared to the general case of expository text in which examples are traditionally used when novel or abstract concepts are being introduced). The main difference between generating examples for purely textual descriptions and our current application is in the selection of values used for illustration: one of constraints in our current situation is the ability of the reader to identify the grapheme in question. Rather than use a strategy that finds and uses either extreme, limiting values or more prototypical values, the current application requires the selection of a grapheme that is easy to identify and interpret values mapped to it. To enable this, the system must be able to reason about individual graphemes as well as the picture as a whole: which graphemes are not crowded by other graphemes, are not too small, thin or otherwise unconventional to make interpretation difficult, have data values mapped to them that can be discussed in the caption[14] , etc.

[14] This situation often occurs in maps, when certain tuples are better for examples because they're close to landmarks that can be used to identify them.), etc.

6. System Implementation and Evaluation: A Discussion

In general, it is essential to empirically evaluate theories and systems that purportedly implement them. Not only do evaluations help others understand the strengths and limitations of various hypotheses and systems, but they also facilitate comparisons between competing claims in many cases. However, NLG evaluations are considered difficult (Hovy and Meeter, 1990). NLG systems can be evaluated at many different levels, some of them being orthogonal to each other. Our case is no exception. There are at least three different, and equally important questions that one could investigate further:

- **validity of the complexity metric:** perhaps the most critical aspect, since without a valid complexity metric, the system would not be able to generate reasonable captions irrespective of how well any/all of the other components performed. The only way to corroborate the complexity metrics we discussed here would be through rigorous user experiments; fortunately, a recent dissertation on graph comprehension (Shah, 1995) looked at some of the factors in our complexity metrics and found that many of the factors used in our complexity metric were indeed correlated with the increased times required to interpret graphs and charts.

- **validity of the discourse strategies:** the paper discussed three discourse strategies for structuring information presented in the captions. There are at least two ways to evaluate a set of strategies used (1) by performing a corpus analysis on a different set of charts and captions than those used to initially infer the strategies in an effort to see how well they fit the test set: this is the usual approach in machine learning where the learning and test sets are kept separate for precisely this reason. However, this would require significant resources to find and code charts and their captions for both the data displayed and the discourse strategies used. This would help determine whether the set of discourse strategies we had come up with was both consistent and complete. (2) Another way to evaluate the discourse strategies would be by conducting user comprehension tests with various charts and captions generated using different strategies at random: while this would be less efficient at testing the set of strategies for completeness, it would allow us to validate that a particular strategy (from our set of three) was best suited for particular types of charts.

- **utility of the captions generated:** or the value-added test: are the captions and the graphics together better than the graphics alone for some purpose? If so, the value of generating the

captions would be confirmed. We conducted an informal, subjective evaluation of the system over a period of two years. Whenever users interacted with SAGE and were unable to understand a graphic, we suggested that they generate a caption. Later on, we requested feedback on their experience: whether the captions were useful or not, and if they would have liked to see something different. We can categorically state that the captions clearly help in understanding the graphic being presented. The need for natural language explanations seems to arise every time a novel, complex graphic is generated--something that happens quite frequently with SAGE.

A large part of the work we have discussed in this paper is system independent and applicable to any automatic graphic design system. Perhaps the most surprising aspect about our current implementation is how far one can get with such a simple architecture. We made certain simplifying decisions initially in order to get a prototype implemented. Surprisingly few of these simplifying assumptions were problematic down the line. An example of this is our pipelined architecture. Most NLG researchers agree that the various modules in a NLG system need to be strongly interconnected with bi-directional communication and control and use shared data structures. We started off by using a pipelined architecture and were surprised to find that the simplifications seemed to be problematic in only one situation (which we were able to get around by planning appropriately). There are several advantages of a pipelined approach as in our case: not only is it easy to design, implement and test each module independently, it also becomes easy to extend the functionality of any individual module without significantly affecting the others. While such a simplified architecture will certainly not suffice for *all* generation tasks, this is a strong argument for trying this minimal approach to see where it falls short and why.

Over the last two years, this system has been used to generate captions for several hundred figures in different domains (housing-sales, Napoleon's march of 1812, logistics transportation, scheduling, etc.). Porting the system from one domain to another usually requires only specifying the lexicon for the new domain (e.g., “battle,” “troops,” etc.). The fact that the captions generated in each of these--quite different--domains are deemed useful and natural by users is testimony to the effectiveness of the caption generation mechanism currently in place.

It should be noted that there are two shortcomings in the system that will be addressed in future work: (1) the caption generation system, as described here, cannot in general, modify the graphics designed by SAGE if so required by the caption. There are several cases where this capability would be extremely useful, but the caption generation system described here was designed to work after

SAGE had designed and rendered the graphic. There is one specialized case where coordination currently occurs, which is when the caption generator presents an example. In that case, the caption generator can request that the graphemes corresponding to the tuple values used in the example be highlighted in the picture; (2) the system does *not*, as yet, analyze the data set for interesting patterns or clusters of data points. To do this, the system will need a clustering analysis module that can be used by the caption generator. As a result, the system cannot generate captions of the sort *"this chart shows that sales were flat throughout 1995, but rose sharply in 1996."*

7. Related Work

Most previous efforts in generating intelligent multi-media presentations have focused on coordinating natural language and graphical depictions of real world devices (e.g., military radios (Feiner and McKeown, 1991) and coffee makers (Wahlster et al., 1993)) for generating instructions about their repair or proper use. These projects tackled important problems such as apportioning content to media and generating cross references between them. Research has also focused on issues regarding the generation of coordinated presentations in applications where the graphics are familiar, or possess an obvious mapping between the dataset and a graphical image (e.g., weather maps (Kerpedjiev, 1992) and network diagrams (Marks and Reiter, 1990)).

Our work differs from these projects in two ways. The first difference concerns the type of data that our system deals with. Unlike the presentations generated by the systems mentioned above, presentations generated by SAGE are usually based on abstract or relational information (e.g., census reports, logistics data, hospital administration data, real estate sales data), lacking any obvious graphical depiction. Second, although our long term goal is to generate coordinated multi-media explanations using informational graphics and natural language, our focus in this paper was on generating effective natural language explanations about the graphical presentations. In order to do this, the system had to explicitly reason about the perceptual complexity of the presentation. Generating such captions is an important component of constructing multi-media explanations involving integrative graphical displays.

The POSTGRAPHE system (Fasciano, 1996; Fasciano and Lapalme, 1996) is the closest related research effort. As in our work, POSTGRAPHE generates statistical graphics and accompanying captions. However, the issues considered in our work differ from those in POSTGRAPHE in several ways and both the text and the graphics generated by POSTGRAPHE emphasize aspects orthogonal to the ones considered in our project. For instance, POSTGRAPHE can take as input a list of aspects that

should be conveyed by the presentation. (These goals are represented in the system as a pre-defined set of templates, such as, “show the evolution of <attribute-name-1> with respect to <attribute-name-2>”.) This information is then used by POSTGRAPHE to not only generate an appropriate type of diagram (e.g., a line chart), but also to generate a caption that explicitly captures the specific aspects of interest, such as: “The profits were at their highest in 1975 and lowest in 1974, with about half their 1975 value.” This is in contrast to our system, which does not reason about trends or relationships between different data points shown in the graphic. Instead, our work has focused on describing complex data to grapheme mappings and deriving metrics for perceptual complexity. This is due, in part, to the nature of the graphical presentations that the two systems can design. (\sc SAGE), for instance, is capable of designing novel graphical presentations for very complex datasets, using techniques such as multiple grapheme composition and space alignment to facilitate cross-attribute comparisons. The range of graphical capabilities in POSTGRAPHE is more limited. Combined with the fact that the graphics are generated in response to an explicit user goal, user comprehension problems in POSTGRAPHE are less likely than in our system. Perhaps in light of this, POSTGRAPHE does not need to explicitly analyze its graphic presentations for potential ambiguities or perceptual complexities, and the captions accompanying the graphic do not take these factors into account.

However, our current implementation, described in the paper, should not be confused with our long term research agenda; it was designed as a framework to evaluate more sophisticated capabilities. These include some of the capabilities that POSTGRAPHE has, particularly those dealing with the generation of information about trends and patterns. We plan to extend the approach used by POSTGRAPHE to take into account both the writer's goals and domain- and data-specific aspects. To this end, we are developing a language to express presentation intentions, taking into account both our experiences as well as the language used in POSTGRAPHE. Furthermore, whereas the sequence of presentation goals to be achieved are part of the input to POSTGRAPHE, our new framework generates these dynamically by integrating a data analysis module with a discourse planner. The data analysis module is being designed to identify all possible relevant aspects of the data based on the domain specification and an analysis task. The planner can use a variety of strategies to select and organize these aspects into complex arguments that can be realized as presentations combining both text and graphics (see (Kerpedjiev et al., 1997) for further details on our new framework).

8. Conclusions and Future Work

Captions that explain novel or creative graphics can be crucial in understanding how data and various relations are expressed in them. This paper presents a framework for generating explanatory captions for information graphics. The system generates captions based on: (1) a representation of the structure of the graphical presentation and its mapping to the data it depicts, (2) a framework for identifying the perceptual complexity of graphical elements, and (3) the structure of the data expressed in the graphic.

One of the strengths of our approach is that the system is able to generate surprisingly effective and comprehensible descriptions in the absence of a detailed semantic model for the domain. The captions shown in this document were generated using only the data characterization used by SAGE for designing the visual presentation and an extremely basic lexical representation. Thus, the caption generation mechanism can be quickly and easily transferred to another domain (the only thing required is a lexicon for the new terms). However, this is also a limitation, because under certain circumstances, the system generates seemingly odd descriptions. This occurs in cases where the underlying database representation happens to contain attribute specifications that differ from the way they would normally be described in discourse. For instance, if the database schema happened to relate house attributes such as house address, number of rooms and sale price to the (owner) of the house, rather than the house itself, the system would generate statements such as “John's sale price is…”

A secondary limitation of our implementation is that it does not generate general graphical annotations. While the system can (and does) highlight specific graphemes in the presentation if so required by the planner (currently done to single out the tuple being used in an example), the system does not coordinate the generation of graphical keys and the captions. This is because our speech act language does not permit bi-directional communication between the text planner and SAGE . The ability to specify arbitrary graphical annotations in the speech act language would make the current simple specification quite complex. As we extend the planning framework to generate both the text and the graphics, this will be remedied as well.

There are two ways to facilitate an effective use of a graphic: (1) explaining how the graphic expresses its data, and (2) conveying what aspects of the data are relevant to the current user's analysis task. In the work described in this paper, we have addressed the first issue. We are currently working on the second one.

Appendix A: The Speech Act Specification

```
<speech-act> ::=    ((PROCESS <symbol>)
                     (CIRCUM   <circum-expr>)
                     (AGENT    <entity-expr>)
                     (AFFECTED <entity-expr>))
                    |
                    ((PROCESS ASCRIPTIVE)
                     (CIRCUM     <circum-expr>)
                     (IDENTIFIED <entity-expr>)
                     (IDENTIFIER <entity-expr>))

<circum-expr> ::=   :NONE |
                    (<symbol>+) |
                    (ONLY-ONE   <symbol>) |
                    (<position> <symbol>) |
                    (EXAMPLE    <symbol>)

<entity-expr> ::=   :NONE|
                    (<symbol>+)|
                    (ONLY-ONE <symbol>) |
                    (:SET (<symbol>+))
                    (:PROP <entity-expr> <entity-expr>) |
                    (RELATIONSHIP (:SET (<symbol>+))
                                  (:SET (<symbol>+))) |
                    (DATA-SET (NAME (<symbol>))
                              (RELATION (<symbol>))) |
                    (<number> <symbol>)

<position> ::= LEFT | RIGHT | MIDDLE | TOP | BOTTOM
```

Figure 21
The BNF for specifying speech-acts in our system.

```
STEP: (SPEECH-ACT
        ((PROCESS PRESENT)
         (CIRCUM :NONE) (AGENT (CHART))
         (AFFECTED
          (DATA-SET (NAME (:SET (TS-2480)))
           (RELATION (:SET (HOUSE-SALES-INFO))))
          (ANCHOR DOM-2516))))

STEP: (SPEECH-ACT
        ((PROCESS SHOW)
         (CIRCUM (:ONLY-ONE CHART)) (AGENT (Y-AXIS))
         (AFFECTED
          (ANCHOR (Y-AXIS) DOM-2589))))

STEP: (SPEECH-ACT
        ((PROCESS SHOW)
         (CIRCUM (:ONLY-ONE CHART))
         (GRAPHEME HORIZONTAL-INTERVAL-BAR-SET)
         (AGENT
          (:PROP (X1) (HORIZONTAL-INTERVAL-BAR-SET)))
         (AFFECTED
          (:PROP ((:SET (DOM-2512))) (DOM-2589 DOM-2516)))
         (ANCHOR DOM-2516)))

STEP: (SPEECH-ACT
        ((PROCESS SHOW)
         (CIRCUM (:ONLY-ONE CHART))
         (GRAPHEME HORIZONTAL-INTERVAL-BAR-SET)
         (AGENT
          (:PROP (X2) (HORIZONTAL-INTERVAL-BAR-SET)))
         (AFFECTED
          (:PROP ((:SET (DOM-2517))) (DOM-2589 DOM-2516)))
         (ANCHOR DOM-2516)))

STEP: (SPEECH-ACT
        ((PROCESS SHOW)
         (CIRCUM (:ONLY-ONE CHART))
         (GRAPHEME NIL)
         (AGENT (MARK-SET))
         (AFFECTED
          (:PROP ((:SET (DOM-2590))) (DOM-2589 DOM-2516)))
         (ANCHOR DOM-2516)))
```

Caption-Generated: *This chart presents information about house sales from data-set TS-2480. The y-axis shows the houses. The left edge of the bar shows the hosue's selling price whereas the right edge shows the asking price. The mark shows the agency estimate.*

Figure 22
Plan steps and the corresponding caption generated are shown here. (Terms such as DOM- 2516 are pointers to domain concepts and attributes in the KB).

Acknowledgements

This work was conducted under the auspices of DARPA agrant DAA-1593K0005. The results and opinions reported here are that of the authors and do not necessarily represent the views of DARPA or the US government.

Joe Mattis helped formulate and implemented the SAGE graphical complexity metric. Other members of SAGE project who participated in the design of this system at various points include Jake Kolojejchick, Mark Derthick, Stefan Kerpediej and Nancy Green. Help in preparing some of the figures for this paper came from Donald J. Mulder, Jr. The first author would like to acknowledge helpwith some questions about FUF/SURGE from Charles Callaway.

We are grateful to four anonymous reviewers who helped substantially improve the paper. It goes without saying all remaining errors are solely our fault.

References

André, Elisabeth and Thomas Rist. 1994. Referring to world objects with text and pictures. In *Proceedings of COLING-94*, pages 530--534.

Appelt, Douglas. 1985. *Planning English Sentences* . Studies in Natural Language Processing. Cambridge University Press.

Beattie, Vivien and Michael John Jones. 1994. Information design and manipulation: Financial graphs in corporate annual reports. *Information Design Journal*, 7 (3):211--226.

Casner, Steven. 1991. A task-analytic approach to the automated design of graphic presentations. ACM Transactions on Graphics, 10(2):111--151, April.

Cleveland, William S. and Robert McGill. 1987. Graphical perception: The visual decoding of quantitative information on graphical displays of data. *Journal of the Royal Statistical Society*, 150 (Series A, Part 3):192--229.

Dale, Robert. 1992. *Generating Referring Expressions* . ACL-MIT Series in Natural Language Processing. The MIT Press.

Dale, Robert and Ehud Reiter. 1995. Computational Interpretations of the Gricean Maxims in the Generation of Referring Expressions. *Cognitive Science*, 19:233--263.

Dalianis, Hercules. 1996. *Concise Natural Language Generation from Formal Specifications*, Ph.D. thesis, Royal Institute of Technology/Stockholm University, June.

Derr, Marcia A. and Kathleen R. McKeown. 1984. Using focus to generate complex and simple sentences. In *Proceedings of the Tenth International Conference on Computational Linguistics, COLING*, pages 319--326.

Elhadad, Michael. 1992. *Using Argumentation to Control Lexical Choice: A Functional Unification Implementation.* Ph.D. thesis, Columbia University, New York, NY. (The SURGE grammar for FUF is available via anonymous FTP at: ftp://www.cs.columbia.edu/fuf).

Elhadad, Michael and Jacques Robin. 1992. Controlling content realization with functional unification grammars. In R. Dale, E. Hovy, D. Rosner, and O. Stock, editors,

Proceedings of the Sixth International Workshop on Natural Language Generation . Springer Verlag.

Fasciano, Massimo. 1996. *Génération intégrée de textes et de graphiques statistiques* . Ph.D. thesis, University of Montreal, Montreal, Canada.

Fasciano, Massimo and Guy Lapalme. 1996. PostGraphe: a system for the generation of statistical graphics and text. In Donia Scott, editor, *Proceedings of the Eighth International Workshop on Natural Language Generation*, Brighton, U.K.

Feiner, Steven K. and Kathleen R. McKeown. 1991. Automating the generation of coordinated multi-media explanations. *Computer*, 24(10):33--42, October.

Gordon, Peter C., Barbara J. Grosz, and Laura A. Gilliom. 1993. Pronouns, Names, and the Centering of Discourse . *Cognitive Science*, 17(3):311--349.

Grosz, Barbara J. 1977. The representation and use of focus in dialogue understanding. Technical Report 151, SRI International, Menlo Park, CA.

Grosz, Barbara J., Aravind K. Joshi, and Scott Weinstein. 1995. Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics*, 21(2):203--226.

Grosz, Barbara J. and Candace Sidner. 1993. Lost intuitions and forgotten intentions. *In Workshop on Centering Theory in Naturally - Occurring Discourse* .

Hegarty, Mary and Marcel A. Just. 1993. Constructing mental models of machines from text and diagrams. *Journal of Memory & Language*, 32(6):717--742, December.

Hovy, E. H. and K. F. McCoy. 1989. Focusing your RST : A step toward generating coherent multisentential text. In *Proceedings of the Conference of the Cognitive Science Society*, Ann Arbor, MI.

Hovy, Eduard and Marie Meteer, editors. 1990. *Proceedings of the AAAI Workshop on Evaluating Natural Language Generators.* AAAI, Boston, MA.

Huang, Xiaorong and Armin Fiedler. 1996. Paraphrasing and aggregating argumentative text using text structure. In *Proceedings of the 8th International Workshop on Natural Language Generation.*

Kerpedjiev, Stephan. 1992. Automatic generation of multimodal weather reports from datasets. In *Proceedings of the 3rd International Conference on Applied Natural Language Processing*, pages 48--54, Trento, Italy.

Kerpedjiev, Stephan, Giuseppe Carenini, Steven F. Roth, and Johanna D. Moore. 1997. Integrating Planning and Task-based Design for Multimedia Presentation. In *Proceedings of the Second International Conference on Intelligent User Interfaces (IUI '97)*,pages 145--152, Orlando, FL, January.

Large, Andrew, Jamshid Beheshti, Alain Breuleux, and Andre Renaud. 1995. Multimedia and comprehension: The relationship among text, animation, and captions. *Journal of the American Society for Information Science*, 46(5): 340--347, June.

Mackinlay, Jock D. 1986. Automating the design of graphical presentations of relational information. *ACM Transactions on Graphics,* 5(2):110--141, April.

Marks, Joe W. 1991. *Automating the Design of Network Diagrams.* Ph.D. thesis, Harvard University, Cambridge, MA.

Marks, Joe W. and Ehud Reiter. 1990. Avoiding unwanted conversational implicatures in text and graphics. *In Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI-90)*, pages 450--456, Boston, MA. AAAI, AAAI/MIT Press.

Maybury, Mark T. 1991. Topical, temporal, and spatial constraints on linguistic realization. *Computational Intelligence,* 7-4:266--275.

McKeown, Kathleen, Steve Feiner, Jacques Robin, Dorée Seligmann, and Michael Tanenblatt. 1992. *Generating Cross-References for Multimedia Explanation. In Proceedings of Tenth National Conference on Artificial Intelligence,* San Jose, California, July 12-17.

McKeown, Kathleen R. 1985. *Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text.* Cambridge University Press, Cambridge.

Mittal, Vibhu O. and Cécile L. Paris. 1992. Generating Object Descriptions which integrate both Text and Examples . In *Proceedings of the Ninth Canadian Artificial Intelligence Conference (AI/GI/VI 92),*pages 1--8. Canadian Society for the Computational Studies of Intelligence (CSCSI), Morgan Kaufmann Publishers.

Mittal, Vibhu O. and Cécile L. Paris. 1993. Automatic Documentation Generation: The Interaction between Text and Examples . In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence,* pages 1158--1163, Chambery, France. IJCAI.

Nugent, Gwen C. 1983. Deaf students' learning from captioned instruction: The relationship between the visual and caption display. *Journal of Special Education,* 17(2):227--234.

Reiter, Ehud. 1990. The computational complexity of avoiding conversational implicatures. In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics,* pages 97--104. ACL.

Roth, Steven F. and William E. Hefley. 1993. Intelligent Multimedia Presentation Systems: Research and Principles . In *Intelligent Multimedia Interfaces.* AAAI/MIT Press, Menlo Park, CA, pages 13--59.

Roth, Steven F., John Kolojejchick, Joe Mattis, and Jade Goldstein. 1994. Interactive graphic design using automatic presentation knowledge. *In Proceedings of CHI'94: Human Factors in Computing Systems,* Boston, MA. ACM/SIGCHI.

Roth, Steven F. and Joe Mattis. 1990. Data characterization for intelligent graphics presentation. In *Proceedings of the 1990 Conference on Human Factors in Computing Systems,* pages 193--200, New Orleans, LA. ACM/SIGCHI.

Schmid, Calvin F. 1983. *Statistical Graphics: Design Principles and Practices.* John Wiley and Sons. Shah 1995] Shah-PhD

Shah, Priti. 1995. *Cognitive Processes in Graph Comprehension.* Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA 15213.

Shaw, James. 1995. Conciseness through aggregation in text generation. *In Proceedings of the Thirty-third Annual Meeting of the Association of Computational Linguistics*, pages 329--331. ACL. (Student Session).

Sidner, Candace L. 1979. *Towards a Computation Theory of Definite Anaphora Comprehension in English Discourse.* Ph.D. thesis, MIT, Cambridge, MA.

Tufte, Edward R. 1983. *The Visual Display of Quantitative Information* . Graphics Press, Cheshire, Conn.

Wahlster, Wolfgang, Elisabeth Andre, W. Finkler, H. J. Profitlich, and Thomas Rist. 1993. Plan-based integration of natural-language and graphics generation. *Artificial Intelligence,* 63(12):387--427, October.

Walker, Marilyn, Masayo Iida, and Sharin Cote. 1994. Japanese discourse and the process of centering. *Computational Linguistics,* 20(2):193--232.

Walker, Marilyn A. 1993. Initial contexts and shifting centers. In *Proceedings of the Workshop on Centering Theory in Naturally-Occurring Discourse,* Columbus, OH. ACL.

Young, R. Michael. 1997. The LongBow Discourse Planner. Available at URL http://pogo.isp.pitt.edu/young/longbow/longbow.html.

Young, R. Michael and Johanna D. Moore. 1994. DPOCL: A Principled Approach to Discourse Planning . In *Proceedings of the Seventh International Workshop on Natural Language Generation,* pages 13--20, Kennebunkport, ME, July.

Young, R. Michael, Martha E. Pollack, and Johanna D. Moore. 1994. Decomposition and Causality in Partial-Order Planning . In *Proceedings of the Second International Conference on AI and Planning Systems,* pages 188--193, Chicago, July.