In *Proc. INFOVIS '96 (1996 IEEE Symp. on Information Visualization)* page 13-20*,* San Francisco, CA, October 28–29, 1996

Data Characterization For Automatically Visualizing Heterogeneous Information

Michelle X. Zhou **Steven K. Feiner**
Department of Computer Science
Columbia University
500 West 120th St., 450 CS Building
New York, NY 10027
{zhou,feiner}@cs.columbia.edu

Abstract

Automated graphical generation systems should be able to design effective presentations for heterogeneous (quantitative and qualitative) information in static or interactive environments. When building such a system, it is important to thoroughly understand the presentation-related characteristics of domain-specific information. We define a data-analysis taxonomy that can be used to characterize heterogeneous information. In addition to capturing the presentation-related properties of data, our characterization takes into account the user's information-seeking goals and visual-interpretation preferences. We use automatically-generated examples from two different application domains to demonstrate the coverage of the proposed taxonomy and its utility for selecting effective graphical techniques.

1. Introduction

Automated presentation systems promise to simplify an application programmer's design tasks by providing a set of design rules that choose among alternative presentation approaches [5, 8, 11]. To design appropriate visual displays, effectiveness criteria are necessary [8, 2]. To establish these criteria, we must understand which of the properties of the information to be visualized are related to presentation design and how they are related. This task is called *data characterization* [10]. A data characterization approach should be flexible enough to easily accommodate new data types and comprehensive enough to capture all presentation-related characteristics of a piece of information.

We present a data-characterization taxonomy that captures the properties of *heterogeneous* information, including both quantitative and qualitative information, in static or dynamic environments. We also discuss the data properties related to a wide range of graphical techniques (e.g., animation [7]). First, we give a brief overview of previous work in data characterization. Next, we present the dimensions used in our data-characterization method and discuss each dimension in detail. We illustrate our taxonomy with a set of examples that are automatically generated by a prototype knowledge-based visualization system that uses this method. Our system uses a top-down hierarchical planning approach to design visual presentations based on information characteristics and design constraints [13]. The examples are from two testbed application domains. One is a computer network management application that visualizes a network's structure and behavior. The other is a hospital information application that provides a care giver with a multimedia summary of information about a patient [4]. Following the discussion of the taxonomy, we provide a longer example that summarizes how our approach can be used to generate part of a presentation. Finally, we present our conclusions.

2. Previous Work

Mackinlay used data properties to guide the automatic design of visual presentations [8]. He described a data characterization method for quantitative data. Roth and Mattis [10] extended Mackinlay's work to address more complex quantitative data types in their system. They also analyzed data properties based on a user's information-seeking goals. These data-characterization approaches have been mostly concerned with presenting relational data using 2D graph displays in a relatively static environment. Thus, they do not take into account data properties related to figurative presentations (e.g., solid models) and dynamic displays (e.g., actively changing data values).

Complementing this work on quantitative data visualization, are two high-level conceptual taxonomies that also address qualitative data. Arens, Hovy, and Vossers [1] developed a vocabulary for characterizing multimedia information. In their work on visual task categorization, Wehrend and Lewis [12] introduced a rough partitioning of data into categories such as shape, location, and structure.

3. Data Characterization

We use six dimensions to describe data characteristics: *data type*, *data domain*, *data attributes*, *data relations*, *data role*, and *data sense*. *Data type* distinguishes a piece of information by whether or not it is divisible into smaller pieces. *Data domain* categorizes information in a semantic taxonomy; for example, whether or not it is a physical entity. *Data attributes* differentiate data based on properties such as an object's shape, and *data relations* specify connections among data. The last two dimensions characterize data based on user information-seeking goals (*data role*) and user visual-interpretation preferences (*data sense*).

We use an object-oriented approach: each piece of information is an *object*. Each object belongs to a data domain and has a data type, properties (data attributes) and connections to other objects (data relations). When an existing type of data is modified (e.g., by addition of a new attribute) or a new type of data or data relation is encountered, it can be added to the taxonomy. In the following subsections we describe each dimension in more detail.

Data Types

There are two basic data types: *atomic* and *composite*.

Atomic. An *atomic* object is the most basic data unit. For example, a car and its weight might both be atomic objects in a traffic simulation. (Note that this is application-dependent. For example, a maintenance application may define a car as a composite object.)

Composite. A *composite* object is composed of a combination of atomic objects and composite objects. For example, a fleet of cars and the hierarchical budget for operating them might be composite objects. Composite objects can be further classified as *sets* and *structures,* based on the structural relations among their parts. These relationships are expressed in the data relations of the composite object and its components.

A *set* is a composite object in which each component is considered only in relation to the set without taking into account relationships among the components. For example, a fleet of cars may be treated as a set.

In contrast, in a *structure,* relationships among components are important. Thus, there are two types of relationships within a structure: relationships between the structure and its components (as in a set), and relationships among components. For example, in a family structure that has family members as its components, the member relation holds between the family and each of its members, while parent-child relations may be specified between certain pairs of members.

Knowledge of the different characteristics of composite objects can facilitate the construction of displays to represent them. Usually, the preferred visual representation for the structural relations among objects has structural properties based on those of the data. For example, a set might be presented using a table, but typically not a flow chart. In contrast, a structure might be presented by a network diagram.

Figure 1. 3D representation of a computer network

Data Domain

To construct an effective visual presentation, it is often important to take into account the data's domain. Roth and Mattis [10] characterize data with regard to quantitative domains such as time, space, temperature, and mass; for example, to allow temperature to be preferentially displayed along a vertical axis. We generalize this classification to accommodate a broader range of information. Each object belongs to one of the following top-level domains, which can themselves be further subdivided: *entity*, *concept*, and *measurement.*

Entity. An *entity* is an object that exists independently and has a unique identity. An entity may correspond to a physical object in the real world (e.g., a car) or may be depicted as a physical object (e.g., a dragon).

Concept. A *concept* is an abstract idea that either exists independently by itself or must be attached to other objects. Unlike entities, concepts are not physical objects. For example, the abstract notion of "age" is independent of any object, while a particular person's "age" is a concept that is attached to that person to specify how old the person is. The quantitative domains identified by Roth and Mattis would be among the subclasses of concept.

Measurement. A *measurement* is a numeric or non-numeric value with or without its unit of measure. A measurement cannot exist by itself, but must be related to a concept, such as weight. For example, we can use "120 lbs." to specify a person's weight *quantitatively*, or "good" to specify a car's performance *qualitatively.*

Data Attributes

Although, at first glance, information and its visual representations may appear to differ greatly from one application domain to another, there are basic connections between the nature of the information (its attributes) and the presentation style. Information properties can be abstracted for all types of information. For example, network nodes

Figure 2. Presentation of a patient's information for a nurse

and links can be represented by spheres and cylinders (Figure 1), and a patient can be represented by an approximation of a human body (Figure 2). In both cases, physical objects are used to represent the information. In the first case, the shapes of the actual nodes and links are completely abstracted, based on the information properties of the application domain. In the second case, enough correspondence to the shape of the actual patient's body is needed to allow references to body parts.

Each data type has its own attributes. Some attributes are shared by different types of data, while others are unique to a particular data type. First, we describe the data attributes that are common to all data types: *form*, *material*, *location, transience*, and *importance.*

Form. *Form* is related to shape and may be one of*: shaped*, *shapeless*, or *none.* A *shaped* form refers to a category of objects that have solid, physical shapes in the real world, such as a person or a car. A *shapeless* object might have a shape under certain circumstances, but its shape changes so often or is so abstract that it is hard to depict the object's precise shape. An example of a shapeless object is water. All conceptual objects use *none* as their form value. The main difference between *shapeless* and *none* is that a shapeless object cannot be mapped to a single shape, although conventional visual cues can be used to display such objects [6]. For example, water is often represented by wavy lines or a running faucet. In contrast, objects whose shape is none usually do not have a particular predetermined structural display. The visual representation for such a conceptual object could be one of a wide range of displays depending on the role this concept plays in the presentation context. For example, the concept "age" might be visualized as the wrinkles on a person's forehead when it is used as an attribute for describing a person's physical appearance. The same concept "age" could also be displayed as a textual table entry when it is considered as an attribute-value component of a person's demographic information.

Material. *Material* refers to the possibly complex set of appearance variables that determine how an object interacts with light. Material models can range from simple RGB color through sophisticated physical illumination models. Material can often be used effectively to encode other non-visual attributes of the data.

Location. While the *form* and *material* attributes determine the possible visual appearance for the data object, *location* affects the graphic design decision as to where and when to display the object. There are two types of locations: *spatial* and *temporal.* Spatial location is either represented by absolute or relative quantitative coordinates (e.g., numeric 2D or 3D coordinates) or by absolute or relative qualitative location (e.g., "in front of" or "near"). Similarly, temporal location can be represented by absolute or relative, quantitative or qualitative, values.

Transience and *importance* are used to describe the intrinsic properties of the information mentioned in [1]. These are especially useful for grouping or partitioning information when a large amount of information is to be conveyed.

Transience. Objects can be either *dynamic* or *static.* Dynamic information varies over time, while static information remains constant. This expresses the same notion as *live* or *dead* [1]. For example, in network management, the traffic load for a particular link varies over time, and animation can be used to capture the changes.

Importance. The fact that some information is more important than others allows us to select different graphical techniques to visually group or partition information. For example, important information may be highlighted to attract the user's attention. Importance can be characterized using several distinctions (e.g., *urgency* vs. *routine* [1], *abnormality* vs. *normality*). Figure 3 displays a hospital patient's lab results. The last item in the list is highlighted in red and italicized to indicate that it has an abnormal value relative to the rest of the normal lab results.

Other attributes are tied directly to composite objects

Figure 3. Lab results for a patient: Hct is abnormal

only: *ordering*, *scalability*, and *continuity*. We adopt the attribute categorizations discussed in [10] and [1], and extend them to address a wider range of information.

Ordering. There are two possible orderings among components of a composite object: *quantitative* and *qualitative*. Quantitative ordering includes both numeric ordering and non-numeric ordering, which correspond to the *quantitative* and *ordinal* notions mentioned in [8]. In our characterization, numeric ordering can be further distinguished as *quantitative individual* or *quantitative range*. For example, the dosages for a set of drugs form a quantitative individual set; in contrast, the price ranges for a set of drugs form a quantitative range set. Non-numeric ordering for orderable qualitative values is called *ordinal* (e.g., a range from "poor" to "excellent"). In contrast, qualitative ordering is not an inherent ordering. Instead, it refers to a logical ordering that is imposed on the objects by the domain application. To avoid introducing new terminology to distinguish qualitative orderings, we overload the terms used in identifying quantitative ordering. We call qualitative sets *nominal* if they are non-ordered (e.g., a list of names) and *ordinal* if they have a logical ordering among their components.

How quantitative ordering can automatically determine graphic design decisions has been carefully examined in both [8] and [10]. Qualitative ordering can also influence the spatial and temporal order of an information presentation. For example, Figure 2 displays a patient's demographics, which is characterized as a set object. To accomplish the communicative task being performed, the health care application itself imposes a logical ordering among the set's components in the original input data (e.g., surgeon precedes operation). The ordering information is used here to determine the relative spatial location of each component (e.g., surgeon is displayed above operation in the same column).

Scalability. Roth and Mattis [10] point out an important distinction between quantitative data sets. They went beyond the ordering attribute and use *coordinates* and *amounts* to differentiate two types of data: *coordinates* specify points in some domain (e.g., a start date for a project), while *amounts* signify the quantitative measurement used to describe a particular attribute of an object (e.g., the price in dollars of a car). See [10] for the usefulness of this distinction.

Continuity. As proposed by Arens, Hovy, and Vossers [1], a quantitative data set can be further distinguished by *discreteness* vs. *continuity*. For example, a set of coordinates can be either continuous (e.g., the set of longitude-latitude pairs on a map), or discrete (e.g., the set of dates in a year). On the other hand, a set of amounts can also be continuous or discrete. For example, the number of days in the months of the year is a discrete set of amounts, while the temperature in degrees over the course of a day is a set of continuous amounts.

Figure 4. Opened network node with its port structure

Data Relations

Characterization along the dimension of data relations captures various types of relationships among objects. We extend the scope of the data relations mentioned in [10] by including semantic relations among data. For example, if we want to display a set of network links and their capacities, we need to know the *functional dependency* relation (syntactic) between the links and the numerical values of their capacities, as well as the *attributive* relation (semantic) between the capacities and the links. First, we describe four basic relations: *functional dependency*, *constituency*, *attribute*, and *enumeration*. Then, we briefly discuss complex relations that describe relations among relations.

Functional Dependency. Functional dependency was first used by Mackinlay [8] to capture the one-to-one mapping relationship from one domain set to another. For example, every day has an average temperature. This type of relation is required by certain graphical techniques (e.g., a line graph) but not by others (e.g., a network diagram) [8, 10].

Constituency. Constituency corresponds to the *has-part* relation and can be further partitioned into *physical* constituency and *conceptual* constituency. Physical constituency either indicates the physical components of a physical object (e.g., a network node has several ports among its physical constituents) or refers to conceptual components that can be treated like physical ones (e.g., virtual links). Conceptual constituency, on the other hand, implies that a concept is conveyed by a set of sub-concepts (e.g., patient demographics is conveyed by age, gender, and other subconcepts). This distinction plays an important role in the process of generating a comprehensive visual display for the information.

Figures 4 and 2 illustrate different graphical techniques that can be applied to represent physical and conceptual constituency relations, respectively. Figure 4 shows that all

Figure 5. Opened physical link with its internal structure

the ports are a part of a node. The visualization rule used here is that when visualizing the physical constituency relation, it is necessary to show both the *part* and the *whole* together to provide sufficient context for the user. However, in Figure 2, the demographics information is simply represented by a table that has its components as its entries. Users with the necessary domain knowledge (i.e., nurses and doctors) can easily recognize that the table presents a patient's demographics. If the user does not know anything about the domain, then the user model will need to indicate this.

Attribute. In an attributive relation, some objects describe certain aspects (attributes) of other objects. For example, in a computer network, a *virtual path segment* is a virtual connection specified within a physical network link, and has attributes such as capacity and utilization. Furthermore, these relations can be effectively encoded by the corresponding image's visual attributes, such as its shape, size, or color. In Figure 5, the capacity of a virtual path segment is mapped to the diameter of the cylinder that is the virtual path segment's visual representation.

Enumeration. *Enumeration* includes both *isa* and *instance* relations [9]. Thus, an enumerative relation shows class inclusion (e.g., pitbull *isa* dog) or class membership (e.g., Spot is an *instance* of dog). Figure 2 uses a button to represent a patient's drips (i.e., intravenously administered drugs). At this level, only the button is shown without any details about the drips. In contrast, in Figure 6, the drips button is associated with a panel that displays details of the drips. There is an enumerative relationship between the drips concept and its components: each component is an *instance* of drips. Normally, to convey the information comprehensively, we need to show the generic concept of the classification first (e.g., the drips button), then *enumerate* the instances or specific subclasses of the concept in detail.

All the relations discussed above usually do not exist independently from each other. More often, one or more types of relations coexist in one piece of information. Figure 5 was designed to convey all network virtual path segments within a physical link between two nodes, such that the segments reflect their capacities. In the design process, the system recognizes that there are two types of relations: *constituent* relations between virtual path segments and the link, and *attributive* relation between segments and their capacities. It then decomposes the task into two subtasks that design presentations guided by these relations.

On the other hand, when the request is to display several relations together, it is necessary to take into account the interconnections between those relations in order to integrate all related information in one presentation. Using the example about complex relations provided in [10], suppose we want to show a set of activities with their start dates, end dates, and durations. In our characterization method, we know the attributive relations between the start dates, end dates, durations, and the activities. Moreover, we also know that another two attributive relations exist between start date and duration, and between end date and duration. A single composite relation can be recursively defined, since duration is an attribute of activity, while start date and end date are attributes of duration. Based on this knowledge, an effective display can be composed in which a bar encodes the duration, and the start and end points of the bar indicate the start date and end date, respectively. Understanding complex relations enables us to perceive both explicit and implicit connections among data, and guides us to select the best possible graphical techniques to convey the information.

Attributes of Data Relations

We use a straightforward combination of *dimensionality* [1], along with *coverage*, *cardinality*, and *uniqueness* [10], to describe the attributes of a relation. In a dynamic application domain, such as network management, the properties of data relations can change over time. For example, normally there are three types of traffic in a particular kind of network link (its cardinality value is fixed-multival-

Figure 6. Drips button is pressed to show its components

ued); however, at some times there might be only one or two types of traffic in certain of these links. Such observations lead to more design requirements: either we must be able to predict future changes in advance to make the design generic enough to cope with different expected situations, or we need to redesign the visual presentation on-the-fly when the current one no longer effectively conveys the new information.

Data Role

To generate effective presentations, not only do we need to know the data's *intrinsic properties* (e.g., data type or data attributes), but we also need to identify the information-presentation tasks. Casner [3] discusses how user tasks can affect the presentation design, while Roth and Mattis [10] also analyze data properties based on the user's or application's information-seeking goals. We use the *data role* dimension to characterize the functional role each piece of information plays in a visual presentation context. In other words, data role specifies particular information-seeking tasks. It could be given by a user during interaction, or could be expressed by domain experts during knowledge acquisition.

We adopt the taxonomy of visualization goals proposed by Wehrend and Lewis [12], which is a superset of the display functions listed in [10]. These visual tasks include: *categorization* (categorizing information), *clustering* (grouping information), *identification* (revealing information identity), *distinguishing* (displaying the difference among information), *comparison* (comparing one type of information to another), *association* (associating one type of information with another), *ranking* (comparing all information in a particular order), *correlation* (correlating one type of information to another), and *distribution* (partitioning information). For example, Figure 5 displays the virtual path segments within a physical link, and those segments are *distinguished* by their capacities.

Data Sense

To construct an effective visual presentation, the user or application domain visual preference should also be taken into account. For example, one way to distinguish the elements of a small set of objects is to use a different color for each. However, such a presentation would not be effective for a color-blind user. Presentation preferences might also be a function of a specific application domain, perhaps because users in that domain have been trained to use a particular presentation style. For example, to convey the details of a patient's intravenous drips, each drip's name and dosage are displayed in a table (Figure 6), instead of being plotted in a bar graph, because our domain experts indicated that our hospital caregiver audience would prefer the table to a bar graph.

This notion of visual preference leads to our last characterization dimension, *data sense*. The word "sense" can be used to refer to one of a set of meanings for a word or phrase, as in a dictionary. For example, the word "person" can mean a "human being" in one sense, and a "human body" in another. We have adopted this meaning of "sense" to coin the term "data sense," which signifies a preferred way to present the data visually.

Like data role, data sense can be specified by an end user to indicate her individual preference, or captured in the process of knowledge acquisition. Alternatively, data sense might also be inferred, based on other data characteristics. Data sense could be extended to include a wide variety of low-level visual preferences (e.g., color or shape). Thus far, however, we have limited data sense only to the set of five high-level presentation style preferences described below: *label*, *list*, *plot*, *symbol*, and *portrait*.

Label. One way to display information is through a textual label (e.g., a network node may be represented by the generic string "node" or the specific string corresponding to the node's name). A label can be enriched by adding graphics, such as underscoring or a button shape, to convey additional information. For example, in Figure 2, the drips are represented by a labeled button, implying that additional information can be obtained by "pressing" the button. The data sense can be derived from other data properties. For example, using the terminology of [12], suppose the data role for drips is to *identify* the existence of the drips without identifying its components. In addition, we may know from the user model that an accurate physical representation of the drips is not needed, so that the data sense can be *label*. Furthermore, the data type of drips is composite and there is an enumeration data relation between drips and its components. Our system has a rule that states that if the user should be informed that an object is composite and given the ability to explore it, then a labeled button can be created that allows the object's components to be accessed.

List. In contrast to the label sense, the *list* sense states that a composite object should be displayed in a tabular form in which all its components are listed textually as table entries. A typical list representation is the lab result display or the demographics display of Figure 3. Here, the *list* preference for representing the lab results or the patient introduction information comes from our medical experts. As in the *label* sense, both physical and conceptual objects can be represented in the *list* sense. Furthermore, in the *list* sense, all components of a composite object are either listed as an attribute-value pair (e.g., the "age" in Figure 3) or as a textual string (e.g., an entry in an indentation chart).

Plot. Both *plot* sense and *list* sense indicate that there is a composite object and that all its components need to be represented. The major difference between *plot* sense and *list*

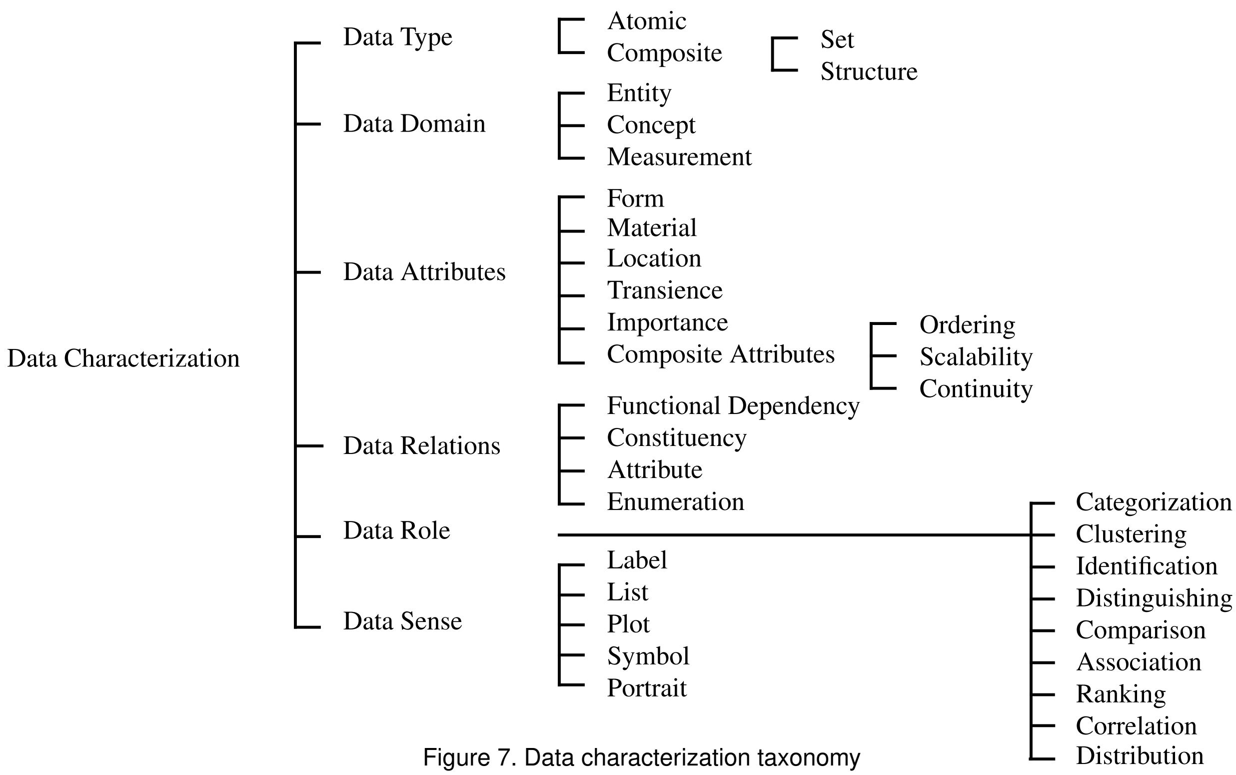

Figure 7. Data characterization taxonomy

sense lies in the primitives used in the representation. In *list* sense, information is represented textually, while graphical primitives such as markers and lines are employed to encode different kinds of information in *plot* sense. Quantitative information can usually be effectively expressed in *plot* sense. The research of [8] and [10] is mostly concerned with how to generate appropriate visual presentations automatically in *plot* sense.

Symbol. Representing a piece of information in *symbol* sense involves the use of a concrete shape object to symbolize the information. For example, a thermometer icon with its scale can be used to symbolize the temperature concept. Both physical and conceptual objects can be represented in symbol sense. We may use a much simplified (abstracted) version of a real image to symbolize a physical object (e.g., a network node may be symbolized as a sphere) or to represent a concept by implying the special connection between the real object and the concept (e.g., the thermometer and the temperature).

Portrait. Compared to symbol sense, visual representations require much more detail and precision when the information needs to be displayed as a realistic *portrait*. Such precise and detailed visual representation is necessary for people to carry out certain tasks, such as the design of a product's physical appearance.

Note that the same data may be characterized differently depending upon information needs. For example, Figure 2 shows the drips concept represented in *label* sense, while Figure 6 displays the same concept in *list* sense.

4. Example

An example from our medical application will demonstrate how a piece of information can be characterized for visualization using our data-characterization taxonomy. Figure 6 displays patient information provided to a nurse. The patient information used to create this figure is an *ordinal* (data attributes) *set composite* (data type) *entity* (data domain) specified to be represented *symbolically* (data sense). It contains the components: demographics, several "lines" going into the patient's body, and their contents, which include drips. Here we focus only on the data-characterization method applied to the top-level patient information, and three of its components: demographics, the "line" labeled "Swan-Ganz with Cordis," and drips.

Demographics is a *set-composite concept* (data domain), the "line" is an *atomic entity,* and drips is a *set-composite entity.* Demographics has a data attribute *form* of *none* because it is a *concept*. The "line" has a relative *location* (data attribute) specified in terms of the patient's body, and drips has a relative *location* specified in terms of its "line." The patient information's *ordering* attribute is *ordinal*, which specifies a logical presentation order among its components. Thus, demographics should be first in the presentation, temporally and/or spatially.

There are two types of data relations in this example:

constituent relations between the patient information and its components, and between demographics and its components, and an *enumeration* relation between the drips and its components. The data role of demographics is to *identify* this patient, while the data role of drips is to *identify* the value of each of the drips that are its components. Both demographics and drips have *list* as their data sense and the "line" has *symbol* as its data sense.

A symbolic human figure is used because *symbol* sense is specified at the top level and a spatial representation is needed to allow other components to be arranged relative to the patient's body. The "line" is also represented symbolically. Demographics is presented as described in the discussion of Figure 2 and is mapped to a spatial location that precedes the "line" and drips, based on a top-down reading order. To satisfy the *list* sense, the drips are presented as a labeled button. This button is shown in Figure 6 after the user has pressed it, so that it has popped up a panel that contains the enumerated drips. The drips are placed relative to their "line" and the "line" is placed relative to the body because of their *location* attributes.

5. Conclusions

To automatically generate effective visual presentations for a wide variety of information in an interactive environment, we need a thorough understanding of the information's characteristics. To achieve this goal, we are developing a data-characterization taxonomy that aims to provide comprehensive coverage of heterogeneous information. It uses an object-oriented approach to allow new data properties to be added to the existing taxonomy. As shown in Figure 7, our taxonomy uses six dimensions. Data types distinguish information by whether it is atomic or composite; data domains categorize information in a semantic taxonomy (entity, concept, and measurement); data attributes differentiate data based on its properties (e.g., form and location); and data relations indicate various types of connections among data.

Complementing these intrinsic properties possessed by the data itself are the data properties derived from the user or application tasks (data role), and the user or application visual preference (data sense). Separate from the data characterization taxonomy, but important to its use, is information derived from the user model and presentation dialogue history: how much the user knows about what is being presented and whether the information has been mentioned before in the presentation.

Our research goal is to develop systems that can automatically design effective visual presentations for heterogeneous information in static and dynamic environments. By providing a data characterization taxonomy that builds on previous work, we aim to establish a better understanding of the connections between data properties and visual designs.

Acknowledgments

The research described in this overview is supported in part by DARPA Contract DAAL01-94-K-0119, the Columbia University Center for Advanced Technology in High Performance Computing and Communications in Healthcare (funded by the New York State Science and Technology Foundation), ONR Contract N00014-94-1-0564, and a gift from Digital Image Design.

References

[1] Y. Arens, E. Hovy, and M. Vossers. The knowledge underlying multimedia presentations. In M. Maybury, editor, *Intelligent Multimedia Interfaces*, pages 280–306. The MIT Press, 1993.

[2] J. Bertin. *Semiology of Graphics*. University of Wisconsin Press, 1983. (translated by W.J. Berg).

[3] S. Casner. A task-analytic approach to the automated design of graphic presentations. *ACM Transactions on Graphics*, 10(2):111–151, 1991.

[4] M. Dalal, S. Feiner, K. McKeown, D. Jordan, and B. Allen. MAGIC: An experimental system for generating multimedia briefings about post-bypass patient status. Submitted, 1996.

[5] S. Feiner. APEX: An experiment in the automated creation of pictorial explanations. *IEEE Computer Graphics and Applications*, 5(11):29–37, 1985.

[6] E. Goldsmith. *Research Into Illustration: An Approach and A Review*. Cambridge University Press, 1984.

[7] P. R. Keller and M. M. Keller. *Visual Cues: Practical Data Visualization*. IEEE Computer Society Press and IEEE Press, 1993.

[8] J. Mackinlay. Automating the design of graphical presentations of relational information. *ACM Transactions on Graphics*, 5(2):110–141, 1986.

[9] E. Rich and K. Knight. *Artificial Intelligence*. McGraw-Hill, Inc., second edition, 1991.

[10] S. F. Roth and J. Mattis. Data characterization for intelligent graphics presentation. In *Proceedings of the 1990 Conference on Human Factors in Computing Systems*, pages 193–200, New Orleans, LA, 1990. ACM/SIGCHI.

[11] S. F. Roth and J. Mattis. Automating the presentation of information. In *Proceedings of the IEEE Conference on AI Applications*, pages 90–97, 1991.

[12] R. Wehrend and C. Lewis. A problem-oriented classification of visualization techniques. In *Proceedings of the First IEEE Conference on Visualization: Visualization 90*, pages 139–143. IEEE, Los Alamitos, CA, October 1990.

[13] M. Zhou and S. Feiner. Top-down hierarchical planning for automated design of coherent visual discourse. Submitted, 1996.