Data Characterization Schema for Intelligent Support in Visual Data Analysis

Gennady Andrienko and Natalia Andrienko

GMD - German National Research Center for Information Technology
Schloss Birlinghoven, Sankt-Augustin, D-53754 Germany
gennady.andrienko@gmd.de
http://allanon.gmd.de/and/

Abstract. The project CommonGIS[1] aims at building a system allowing users to view and analyze geographically referenced thematic data. The system is oriented to the general public, i.e. people without special training and expertise in map design. Therefore the system is required to understand data semantics that, hence, must be formally represented. The project involves development of a data characterization schema that defines what knowledge about the data and in what form should be provided to the system for enabling intelligent and user-friendly support in visual data analysis. In this paper we propose a schema developed on the basis of the approach adopted in the system Descartes for automated thematic mapping. The approach involves creation of a domain model containing relevant notions and establishing of a correspondence between data components and the notions. Presence of the domain model is the main difference of the described schema from the ones proposed earlier for the purposes of automated data visualization.

Keywords: geographically referenced data, conceptual data characterization, knowledge-based systems, cartographic visualization

1 Motivation

The project CommonGIS (started in November 1998) has the motto "GIS for everyone". The goal is to build a system able to intelligently assist users in exploration of spatially referenced data, the latter being various thematic data associated with objects and locations in space. A key role in the exploration belongs to visual investigation of maps that represent the data. A map serves for a human analyst as a model of reality that preserves spatial relationships and thereby can expose significant spatial patterns and dependencies.

A more exact specification of our goal can be given as follows:

1. Context. A person or organization, further referred to as *data provider*, has a set of spatially referenced data as well as coordinates and geometry of spatial objects the data refer to. S/he wishes to make the data available to a certain circle of *end users*.

[1] URL: http://commongis.jrc.it/. The project is partly funded by the European Commission, DGIII, contract N 28983 (November 1998 – April 2001).

C. Freksa, D.M. Mark (Eds.): COSIT'99, LNCS 1661, pp. 349-366, 1999

The end users need to view and analyze the data in the course of their education or work. This requires the use of maps presenting the data the users are interested in.

2. Users. The users are not supposed to know the principles of graphic and cartographic presentation of information. Therefore the responsibility for selection of an adequate presentation method for a data subset to analyze and for making appropriate settings cannot be assigned to them. A user may also be unaware about possible useful transformations of data, such as converting absolute amounts to relative, or about the ways to perform them.

3. Scenario. The system gives the users convenient tools to select data subsets for analysis from the data set prepared by the data provider. After the subset is selected, the system *automatically* builds a map for data overview employing an adequate presentation technique in accord with data characteristics and relationships among data items. At the same time the system proposes the user a list of possible analysis tasks that could be done with these data. The tasks are formulated using concepts of the application domain the data belong to, for example, "Compare proportions of unemployed in different age groups across the countries". The user chooses a task to pursue, and the system assists her/him in accomplishing it by providing effective visualizations, calculating helpful statistics, and automatically doing necessary data transformations (see [2] for a more detailed view).

4. Maps. The data displays (maps and auxiliary graphics) generated by the system are reactive to user's manipulations and able to change their appearance in real time. We mean here not only such basic operations as zooming, panning, and access to data values through the map. We rather refer to such changes that could increase map expressiveness, expose interesting patterns, or facilitate fulfilling the user's task. Some tools enabling such modifications have been implemented in our Descartes system and described in [3].

To achieve the goal, we decided to elaborate the approach adopted in Descartes [1]. The approach is schematically shown in Fig. 1.

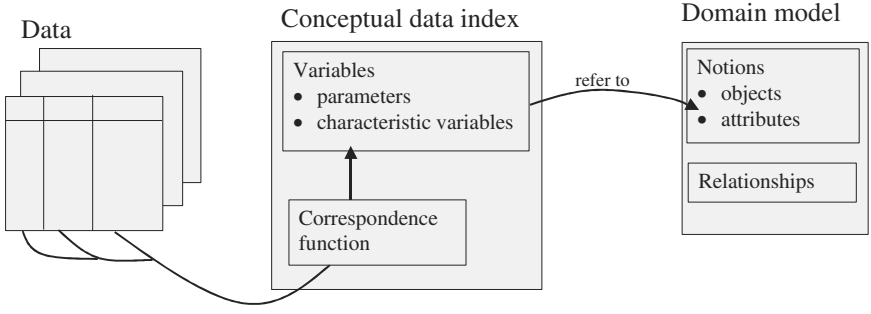

Fig. 1. The approach to data characterization

A domain model is a collection of notions necessary to describe the meaning of a spatially referenced data set. The notions are linked by relationships. Examples of domain notions are "whole population" and "female population" in a demographic application. These two notions are associated by the inclusion relationship. A conceptual data index establishes links between components of the data and notions of

the domain model. These links allow the system to interpret the data and to know relationships among data components.

At the current moment Descartes, on the one hand, does not fully exploit the potential opportunities offered by a domain model. One of the most important of them is a capability to provide a basis, a "common language", for communication between the system and the user about the content of the data and about the user's goals. On the other hand, testing Descartes with various data has exposed incompleteness of the current schema: sometimes the map design was incorrect due to insufficient "understanding" by the system of the nature of data represented. Enhancement of the schema becomes imperative in regard to the more ambitious goal of intelligent support in data exploration, in comparison to automated map generation as a separate task.

The paper describes an advanced data characterization schema we propose for use in the CommonGIS project. The schema defines what knowledge about the data should be provided to a software system for enabling intelligent and user-friendly support in visual data analysis. Prior to the description of the schema we make a survey of relevant literature and compare previous approaches to data characterization with ours.

2 Related Works

2.1 Theoretical Works

Bertin [4] was the first who systematically expounded the principles and rules of presentation design. He taught that in a data set to present a designer should distinguish between the *invariant* (an invariable notion common to all the data) and the *components* (varying concepts)[2]. The invariant is to be reflected in the title of the graphic. The graphical primitives to use should be adequate to the following characteristics of the components:

- *order* of components: which component governs and which is governed. This can be reformulated as independent and dependent variables;
- *number* of components;
- *length* of the components (number of different values or divisions to be distinguished);
- *level of organization*: qualitative, ordered, or quantitative. MacEachren [7] points out that cartographers typically divide the latter category into two ones, *interval* and *ratio*.

While Bertin suggests a general theory to apply to all types of graphics, theorists in cartography focus specifically on the map form of presentation. A study by MacEachren [7] gives a comprehensive overview of contemporary theories and ideas related to maps. Relevant to our work is the view of Nyerges on a map sign as an "entity with an attached bundle of aspects". Its referent can be "a phenomenon[3] with its bundle of properties". In both cases, the bundle includes *space*, *time*, and *theme*.

[2] In practical works based on Bertin's theory components are usually called "variables".

[3] Considering what is shown on maps, cartographers speak about *phenomena* rather than *data*.

Although one of the aspects might dominate in a particular situation, Nyerges argues that a geographic entity cannot be adequately defined without all three.

MacEachren [7] proposes a typology of quantitative data with respect to the properties of their spatial distribution. He introduces 2 orthogonal dimensions: *continuity* and *spatial (in)dependence*. Continuity refers to the spatial completeness of a phenomenon: is it defined everywhere (e.g., population density) or does it occur at distinct separate locations (e.g. number of gas stations per county)? The poles of this dimension are *continuous* and *discrete*. The second dimension refers to the smoothness or abruptness of variation from place to place. If adjacent locations are independent, variation from place to place can be *abrupt*. Spatial variation will be *smooth*, however, if adjacent locations are dependent. The difference between distributions of number of cars and number of cholera cases refers to this second dimension.

To account for Bertin's recommendations, our schema allows for formal representation of a data set invariant as well as expression of the characteristics of components that cannot be automatically retrieved from the data set to present (i.e. order and organization level). The schema reflects the three inherent aspects of geographical entities considered by Nyerges in its semantic categories of attributes. It also includes the MacEachren's typology of phenomena according to their spatial distribution.

2.2 Practical Works: Data Characterization in Software Systems for Computer Aided Graphics Design

Mackinlay [8] was the first who applied Bertin's theory to automated graphic presentation of non-spatial data. The system APT accounted for the data characteristics listed by Bertin. Senay and Ignatius [10] extended the inventory of these characteristics. Regarding quantitative data, they distinguished *scalar* data (single numbers), *vector* data having magnitude and direction, and *tensor* data consisting of several scalar components. Other attended characteristics were *continuity* of data, *functional dependencies* among data variables, s*pacing* between sampling points, *units* of measurement, etc. It can be noted that these characteristics do not capture anything of data semantics.

Roth and Mattis [9] developed a more extended inventory of data characteristics relevant for graphics design:

- Data types:
  - Set *ordering*: quantitative, ordinal, nominal.
  - *Coordinates* vs. *amounts* (capturing the difference between, for example, two o'clock and two hours).
  - *Domain of membership*: some frequently encountered domains such as space, time, temperature, or mass, are predefined. The design software is aware of the stylistic conventions adopted for these domains, e.g. time is shown on a horizontal axis.
- Properties of relational structure:
  - Relational *coverage*: are any values absent? Distinguished are three possible

cases of non-coverage: 1) data are missing, 2) non-applicability, 3) absence of a value is informative, e.g. means absence of the corresponding entity.
  - *Cardinality*: one-to-one correspondence or one-to-many correspondence with a fixed or variable number of corresponding elements.
- *Arity* of relations (unary/binary/N-ary).
- Relationships among relations:
  - *Complex data types* when an N-ary relation cannot be treated as a set of binary relations. The authors stress that such cases require more knowledge of the semantics of data. They propose to incorporate in a graphics design system several predefined complex data types: interval (e.g. start and end dates of a project), statistical abstraction such as mean or standard deviation, and 2-dimensional coordinates.
  - *Algebraic dependencies*, for example, 'total costs' = 'material costs' + 'labor costs'. Such relationships are specified through equations

It may be seen that Roth and Mattis recognized the importance of representation of data semantics. The characteristics "coordinates vs. amounts", "domain of membership", "complex data types", and "algebraic dependencies" are intended to capture certain aspects of meaning of data components and relationships among them.

Jung [5] further developed the data characterization schema proposed by Roth and Mattis [9] for the case of cartographic presentation of territory-related data. He introduced the following geography-specific extensions:

1. *Spatial* type of variables has been added together with relevant characteristics: dimension and meaning of coordinates (*geographic* or *cartesian*), s*cale, projection* etc. Spatial data are classified into *raster*, *vector*, and *geo-reference* data. Vector data, in their turn, may have point, line, or area type.

2. Jung uses the above-described MacEachren's classification of phenomena according to properties of their spatial distribution, i.e. *continuity* and *spatial (in)dependence*.

3. For numeric variables it is indicated whether they are *dependent on area*.

Jung also suggests a more detailed assortment of types of non-geographic variables. In particular, he considers the following categories of numeric variables:

- *amounts*: absolute quantities;
- *measurements*: absolute numbers representing results of measurements (e.g. distance). Along with measurements the corresponding units should be specified;
- *aggregated values*: amounts or measurements summarized by areas. Such variables are always implicitly dependent on the area;
- *proportional values* normalized in the way of division by a fixed value;
- *densities*: amounts or aggregated amounts divided by corresponding areas. As a result, densities do not depend on the area;
- *coordinates* that specify positions in some coordinate system, e.g. on the time axis.

Here we see an attempt to elaborate the schema of Roth and Mattis in regard to data semantics. Jung considers also a number of characteristics related to *data quality* such as reliability and exactness.

2.3 Comparison with the Proposed Approach

The schema presented here includes those of the above-mentioned characteristics that are relevant to spatially referenced data and cannot be automatically retrieved from a data set. However, we tried to regard the aspects of data semantics deeper and in a more consistent way. Compare, for example, some parallel features of our model and of the others.

"**Domains of membership**" [9]. Instead of few predefined "domains" we introduce a system of semantic categories of attributes and require each attribute to be assigned some category. The category has an impact on visual presentation of values of the attribute and determines possible transformations of the data and analysis tasks that can be solved with their use. This differs from the role of the "domains of membership" intended only to preserve certain stylistic conventions, the latter being specified implicitly for the regarded "domains". In our model stylistic preferences can be explicitly specified for any attribute.

Complex data types [9]. Again, instead of few predefined types our schema gives a consistent method to define any domain-specific compound attribute.

Algebraic dependencies [9], **categories of numeric variables** [5]. In our schema it is possible to express semantic relationships among data components, in particular, inclusion relationships that, according to [9], are specified through equations like 'total costs' = 'material costs' + 'labor costs'. Note that relationships are applicable even when not all constituents of a sum are present in a database. Still, our schema allows formulas as well. One can define a derived attribute values of which are obtained from values of other attributes through calculation. Such derived attributes may represent, in particular, *aggregated values*, *proportional values*, and *densities* introduced in [5] as categories of numeric variables. Our approach has at least two advantages: it is not limited to few categories, and it gives more information about the nature of attributes. This information is important for designing data transformations and supporting analysis.

The major difference of our schema from the other approaches to data characterization is the involvement of a *domain model* that provides a basis for a semantic description of data. Consider, for example, the data set fragment presented in Table 1. The table would be meaningless for a reader without the caption that associates the columns with some known concepts and thus enables interpretation of the data and their use. In an intelligent system understanding of data semantics needs

Table 1. An example set of spatially referenced data (fragment).

Country	Total population	Number of male from 0 to 14 years	Number of female from 0 to 14 years	Number of male from 15 to 64 years	Number of female from 15 to 64 years	Number of male of 65 years and over	Number of female of 65 years and over
Albania	3413904	563953	520186	1104371	1026321	86821	112252
Austria	7986664	711127	681087	2677100	2672554	453034	791762
...	…	...	...	...	...	...	...

to be simulated. We propose to do this by relating data components to appropriate notions defined formally in a domain model (see Fig. 1).

We recognize that the previous projects on automated visualization have been rather successful without such an extensive representation of data semantics. However, in our project generation of a picture is not the only objective. Thus, referring to the example data set in Table 1, we wish our system to be able to automatically find, formulate, and support the fulfillment of the analysis tasks like "Compare gender structure of population in different age groups" or "Look at the distribution of percentages of a specific age group (0-14 years, 15-64 years, or 65 years and over) across the countries". The approach with formal representation of essential concepts in a domain model and data indexing through reference to these concepts provides an adequate basis for achieving our goal.

3 Domain Model

3.1 Basics

A domain model is a collection of *notions* defined in an object-oriented manner on the basis of certain generic *metaclasses* (a metaclass is a class instances of which are classes). The metaclasses included in the schema are shown in Fig. 2 with the use of the UML (unified modeling language) notation [11]. Below the names of the metaclasses their *properties* are listed. Names of the properties are followed by their types and, possibly, the default values. Defining an instance of a metaclass, it is necessary to instantiate its properties, i.e. indicate the values the properties assume. The result is a domain-specific *class*. Properties of (meta)classes are inherited by descending subclasses.

Each domain-specific notion should be derived from one of the bottom-level metaclasses, i.e. it should be either an instance of a metaclass or a subclass or an instance of an earlier defined class. Further in this text we use the notation *n*:*C* to indicate that notion *n* is an instance of the class *C*.

Typically a domain model does not contain definitions of instances, i.e. notions denoting individual entities. It is assumed that instances are contained in data, and a model supplies general notions (classes) to describe the data. However, a model may contain notions-instances when this is considered useful.

According to the schema, each notion defined receives a unique *identifier* and is always referred to through this identifier. In the following text identifiers of notions cited in examples are enclosed in apostrophes for distinguishing from other words. Identifiers are supposed to be operated by the system internally and hidden from the users. To present a notion to a user, the system uses its *name*. A notion may have more than one name. It is possible to specify names in several languages, or a short code used inside a data set and a meaningful text this code should be replaced by before presenting the data to the user.

Notions may be linked by various semantic *relationships* (the latter are not represented in the diagram in Fig. 2, except for the subclassing relationship *is-a*).

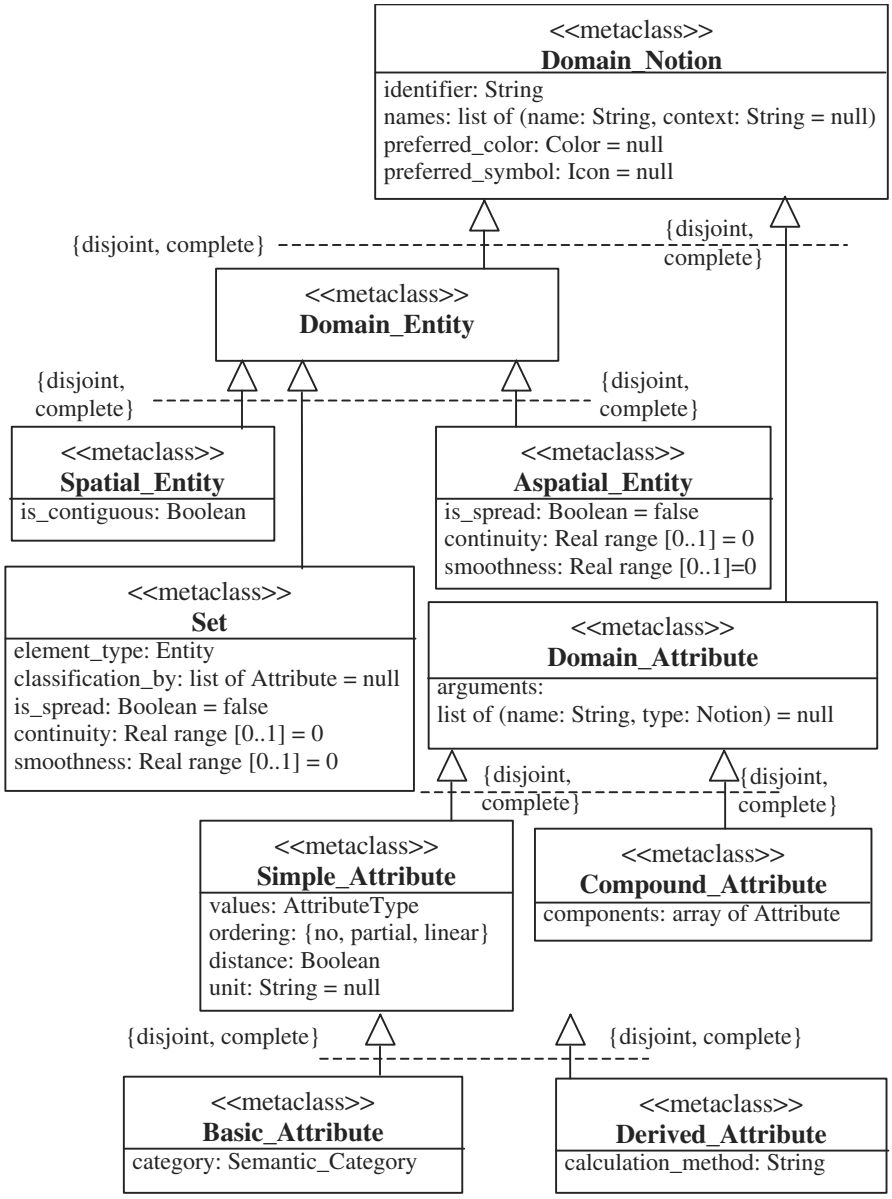

Fig. 2. Generic metaclasses of the schema to be used for definition of domain notions

3.2 Entities

There are two big categories of domain notions, *entities* and *attributes*. Notions-entities denote things, and notions-attributes – properties of the things. We distinguish *spatial* and *aspatial* entities. Spatial entities are pieces of a territory or things having definite locations and shapes in space. A notion to denote a class of spatial entities is defined by subclassing the metaclass Spatial_Entity. The property *is_contiguous* indicates whether the entities of the class completely cover the underlying territory, i.e. belong to some territory division. For example, the classes 'country', 'city district', or 'climatic zone' should be characterized as contiguous whereas 'forest' or 'city' are not contiguous.

Aspatial entities having no exact locations and outlines on a territory may nevertheless be spread over the territory as, for example, climate. Therefore for a notion derived from the metaclass Aspatial_Entity the property *is_spread* should be set. If this property is set to *true*, the properties *continuity* and *smoothness* should be also specified. These two properties reflect the MacEachren's typology of phenomena cited in §2.1.

Notions derived from the metaclass Set represent groups of similar entities, for example, 'population', 'timber', or 'group of countries'. The property *element_type* indicates the class of entities that can be elements of the set. For our example notions-sets the element types are 'person', 'tree', and 'country'. The property *classification_by* specifies attributes that classify set elements into non-overlapping subsets. Thus, 'population' can be divided into subgroups according to 'gender', 'age', 'race' etc. This can be written as 'population' | Set : element_type='person'; classification_by={'gender', 'age', 'race'}. On the basis of this general notion (class) one may define notions-instances denoting various population groups. Defining such an instance, it is necessary to assign values to the classification attributes. For example, 'female children' | 'population' : 'gender'= "female"; 'age'= [0,15]; 'race'= null. The value null means here that the group is not differentiated by race. Another example is 'whole population' | 'population' : 'gender'= null; 'age'= null; 'race'= null.

The properties *is_spead*, *continuity*, and *smoothness* are also relevant to sets.

3.3 Relationships Among Entities

According to the schema, instances of entities may be linked by binary relationships. On the basis of our experience of applying Descartes in various domains we included in the schema a set of predefined relationships important for intelligent presentation and data analysis. Additional domain-specific relationships may also be defined when necessary.

Given below is the list of predefined relationships. We use an infix notation for denoting presence of relationship r between entities e_1 and e_2: $e_1\ r\ e_2$.

- *element-of* is used to represent inclusion of an entity in a set, for example, 'Albania' *element-of* 'European countries'. Here 'Albania': 'country' and 'European countries' : 'group of countries', where 'group of countries': Set : element_type='country'.

- *part-of* is used to link a subset to a set containing it, or a spatial entity to a larger spatial entity including it as an element of a territory division, for example, 'people born' *part-of* 'whole population', 'Scotland' *part-of* 'United Kingdom'.
  The *part-of* relationship is transitive, that is, from A *part-of* B and B *part-of* C follows that A *part-of* C. If A and B are notions denoting sets, E is an entity, then from A *part-of* B and E *element-of* A it may be automatically concluded that E *element-of* B.
- *is-in* is used to denote that an entity is located within the territory occupied by a spatial entity, for example, 'Bonn' *is-in* 'Germany'.
- *derived-from* is used to express changes of spatial entities in time when an entity is divided into smaller entities (e.g. USSR was split into separate states) or several entities are united (e.g. Western Germany and Eastern Germany were united into one state).

Relationships may link notions-instances defined in a domain model or variables in a data index. The relationships *element-of*, *part-of*, and *is-in* provide a basis for automatic aggregation of data over sets and territories. For example, a data set may contain raw data about houses being sold around a city with indication of the district each house belongs to. On this basis it is possible to calculate and visualize aggregated data for districts such as number of buildings in each district, average prices, sizes, etc.

3.4 Attributes

Attributes are notions derived from descendants of the metaclass Domain_Attribute. They are used to represent properties of entities. An attribute may be treated as a function that matches entities with values from a certain value set. This function has an argument that is substituted by particular entities in actual observations. In general, an attribute may have several *arguments*. For example, to denote various proportions or percentages, we need an attribute with two arguments, one for the whole and another for the part the size of which (relative to the whole) is indicated by values of the attribute.

The schema requires that for each argument the type be specified in order to show what can stand for the argument. This is done through reference to a notion of the domain model. For example, the attribute 'population number' has one argument of the type 'population'. This means that the attribute is defined for instances of the notion 'population' like 'female children' and 'whole population' cited in §3.2.

Simple attributes. Defining a simple attribute, one should describe the set of possible *values*. This is done in one of the following ways, depending on the nature and organization of the elements:

1. Reference to one of the predefined *primitive types*. The primitive types *integer*, *real*, *string*, and *logical* represent the thematic aspect of data. There are also special types representing the spatial and temporal aspects. The spatial types are *point*, *line*, and *polygon*. The temporal types are *century, year, month, day, hour, minute*, and *second*.
2. Specification of a range of integer or real numbers. It is possible to define only the

lower boundary, or only the upper boundary, or both boundaries of the set.
3. Enumeration of all values. In this way only a finite set may be represented.
4. Reference to a set instance defined in the domain model. In this case possible values of the attribute are elements this set, i.e. notions linked to it with the relationship *element-of*. For example, an attribute may refer to the set 'religions' including elements 'catholic', 'protestant', 'orthodox', 'muslim', etc.
5. A definition in the form 1, 2, 3, or 4, as defined above, preceded by a modifier *set-of* or *array-of*. These modifiers indicate that the attribute is multi-valued, i.e. several values may be observed simultaneously. For example, one may introduce an attribute to show which tree species grow in different forests. The modifier *array-of* differs from *set-of* in that it signifies the importance of order in that observed values are listed.

Attributes having the type *integer* or *real* are called *numeric*. For numeric attributes *units* of measurement may be specified. For a *logical* attribute it is necessary to indicate which symbol (character string) represents the value "true" and which represents "false"[4].

The property *ordering* is relevant to attributes with value sets defined through enumeration or reference to a set instance. If an attribute is declared as partially or linearly ordered, the order of values should be specified.

The property *distance* brings two alternatives: not recognized and recognized. The latter option means that there exists a function assigning to each pair of values of the attribute a number expressing how far apart the two elements are with respect to some underlying ordering [6, pp. 46-51]. In our schema *distance* is relevant only to numeric attributes. We assume that distances between values are found as arithmetic differences. For some numeric attributes, as those representing rankings, distance may be not recognized.

Compound attributes. A compound attribute is a structure including two or more other attributes. For example, 'address' is a compound attribute with the following *components*: 'post code', 'country', 'town or village', 'street', and 'house number'. A value of a compound attribute is a tuple made by values of its component attributes.

Semantics of attributes. Simple attributes are divided into *basic* and *derived*. Derived are attributes values of which can be obtained by means of calculations over values of some basic attributes. For example, percentage of a population group in total population is calculated through division of the group size by the total population number and multiplication by 100%. The method of calculation indicates the meaning of such an attribute.

The meaning of a basic attribute is represented through ascribing it to one of the *semantic categories*. The categories can be arranged into 3 groups: time, space, and theme, in accord with Nyerges's "bundle of properties" of geographical phenomena [7].

Time

- *Time moment*. This category has four special subcategories that are essential for dealing with *dynamic* spatial entities, i.e. entities changing their location and/or

[4] In our experience we encountered different ways of signifying these values: T and F, 'yes' and 'no' or Y and N, '+' and '-', and 1 and 0.

shape, *transient* entities existing not always, and *instant* entities existing only at a certain moment:
  - moment of *appearing*;
  - moment of *disappearing*;
  - moment of *existence* (relevant to instant spatial entities);
  - moment of *change*.
- *Time interval*. This category may be assigned to a compound attribute having two time moments as components.

Space

A spatial attribute, i.e. an attribute having one of the spatial types, may denote
- *Location;*
- *Shape;*
- *Route;*
- *Direction-source* (initial location);
- *Direction-target*.

Theme
- *Weight*. This category includes attributes denoting absolute numbers, amounts, costs, masses, volumes etc., for example, 'gross national product', 'volume of imports', 'consumed energy', and so on. One subcategory, *size*, is considered separately as having special roles in data interpretation and analysis. Referring to a set, size means number of elements, for example, 'population number'. Referring to a spatial entity, size means area or length.
- *Mark*. This category embraces measurements like 'temperature' that represent marks on some scales rather than amounts, estimations like 'life expectancy', normative values such as 'age of retirement', and all other characteristics that cannot be interpreted as quantities.

The essential difference between "weight" and "mark" is in additivity: if A, B, and C are sets, and $C=A \cup B$, then weight(C)=weight(A)+weight(B), but nothing definite can be said about relationships between mark(C), mark(A), and mark(B). Moreover, the category "mark" may be assigned to a non-numeric attribute what is impossible for "weight".

To represent the meaning of a derived attribute, one should specify its *calculation method*. This should be an arithmetic expression over attributes, argument variables and constants. Examples of useful derivatives are relative sizes or weights, amounts per capita, densities, aggregates over sets, territories, or time intervals such as sum or average, characteristics of dynamics like absolute change, relative change, or rate of change, etc. To define attributes denoting various aggregate characteristics, we suggest for use, in addition to arithmetic operations, the operators *sum*, *average*, *minimum*, *maximum*, and *dominant*.

3.5 Stylistic Preferences

Since one of the main purposes of the proposed schema is to enable good data presentations, it is appropriate to foresee in it the opportunity to express domain-specific stylistic conventions and preferences. Accounting for such conventions in

map design helps users in map interpretation and, consequently, facilitates data analysis.

According to our schema, one can specify presentation preferences for any of defined domain notions. The preferences relevant to cartographic presentation may concern colors and symbols (icons). For an entity it is possible to specify one preferred color or/and one preferred symbol. The system will, whenever possible, use this color or this symbol to represent this entity in a map. For example, in some political application one may associate the ecological party with green color. As a result, building a map with pie charts showing percentages of votes for different parties over electoral districts, the system will paint the sectors corresponding to the ecological party in green.

Defining an attribute, it is possible to specify a preferred color for it. This color will be used to distinguish this attribute from others in diagrams presenting values of several attributes, e.g. in bar charts. Besides, presentation preferences may be attached to values of an attribute. If the attribute has a finite set of values specified through enumeration, one may prescribe a preferred color or/and symbol for each value. For an ordered or numeric attribute a color hue may be selected the shades of which will be used to encode the values. The lowest value will be shown by the lightest shade of this hue, the highest by the darkest one, and the intermediate values will be encoded by proportional degrees of darkness. Another opportunity is to specify two colors, one for the lowest and another for the highest value. In this case the system will encode the values using a double-sided color scheme.

4 Data Index

4.1 Characterization of Data Structure

Considering an item (record) of data, one can distinguish two parts. One part, further called *reference*, defines the context of obtaining the data. The other part, *characteristics,* represents results of measurements, observations, calculations etc. obtained in the given context. The context may include moment of time when the characteristics were obtained, location in space, method of data acquisition, and entity(-ies) the properties of which were measured (observed, calculated, ...). For example, in a data set with economic and demographic characteristics of countries the countries serve for reference whereas population number and gross national product are characteristics. If the data are available for several years, then a year is a reference. If we have population numbers for various groups of population, as in Table 1, these groups form a reference set.

Let a *variable* be a pair (v, V), where v is a unique *label* that distinguishes this variable from all others, and V is the set of possible *states*, or *values*, associated with this variable.

Definition. A *data set* is a function $\mathbf{d: CR \rightarrow CC}$, where CR and CC are Cartesian products $\mathbf{CR=R}_1 \times \mathbf{R}_2 \times \mathbf{...} \times \mathbf{R}_m$, $\mathbf{CC=C}_1 \times \mathbf{C}_2 \times \mathbf{...} \times \mathbf{C}_n$ of value sets of variables r_1,

r_2, ..., r_m and c_1, c_2, ..., c_n, respectively. The variables r_1, r_2, ..., r_m are called *reference variables*, or *parameters*; the variables c_1, c_2, ..., c_n are called *characteristic variables*.

The presented view on data was influenced by Klir [6]. Klir uses the terms "attribute" and "backdrop" that roughly correspond to our notions of "characteristics" and "reference".

In an ideal case a data set **d: CR→CC** is a collection of tuples of the length m+n, where the first m elements of each tuple are values of the reference variables r_1, r_2, ..., r_m, and the remaining n elements are values of the corresponding characteristic variables c_1, c_2, ..., c_n. However, in many cases some reference variables are implied rather than explicitly represented through their values in data tuples. Such was our example data set in Table 1. Presenting it as a table, we provided a caption explaining its meaning. The values of the parameters 'gender' and 'age group' are present only in this caption.

So, to describe a data set, one should explicitly establish a **correspondence** between positions in data tuples (if the data are stored in the table format, these will be table columns) and variables and values of reference variables:

$$\mathbf{corr:}\ \mathbf{P} \rightarrow \mathbf{R} \cup \mathbf{C} \cup \{(c_i, \{(r_j, v_{j,k})\}, 1 \le j \le m, v_{j,k} \in R_j), 1 \le i \le n\},$$

where **P** is the set of numbers of positions in data tuples [1, *l*] (we assume that all data tuples have the same length *l*).

In other words, for each specific number *p*, 1<=*p*<=*l*, **corr**(*p*) is one of the following:

- c_i. This means that the position *p* contains values of the characteristic variable c_i.
- r_j. This means that the position *p* contains values of the parameter r_j.
- a pair (c_i, {(r_j, $v_{j,k}$)}). The first component is always one of the characteristic variables, the second component is one or more pairs (<parameter>, <value>). This means that the position *p* contains values of this characteristic variable referring to the specified values of the specified parameters.

Thus, our example data set may be described by the following correspondence function:

corr(1)="country";
corr(2)=("population number", {("gender", "both"), ("age", "all ages")});
corr(3)=("population number", {("gender", "male"), ("age", "0-14 years")});
corr(4)=("population number", {("gender", "female"), ("age", "0-14 years")});
corr(5)=("population number", {("gender", "male"), ("age", "15-64 years")});
corr(6)=("population number", {("gender", "female"), ("age", "15-64 years")});
corr(7)=("population number", {("gender", "male"), ("age", "> 65 years")});
corr(8)=("population number", {("gender", "female"), ("age","> 65 years")}),

where "country", "gender", and "age" are labels of parameters and "population number" is a label of a characteristic variable, all other texts in quotation marks are values.

Definition. A structural characterization of a data set **d** is a system (**C**, **R**, **corr**), where **C** is the set of characteristic variables, **R** is the set of reference variables (parameters), and **corr** is the correspondence function.

It is possible that a variable in a characterization has only one specified value. Such a variable is called *invariant* (see §2.1 and Bertin [4]). Through an invariant variable it is possible to indicate, for example, that all the data in a data set were collected in 1998.

4.2 Characterization of Data Semantics

In the examples given in the previous subsections we used meaningful texts as labels of variables for better understanding. To enable interpretation of a data characterization by a software system, one must instead explicitly relate variables to notions defined in the appropriate domain model, e.g. r_1:'country', r_2:'gender', r_3:'age', c_1:'population number', where 'country', 'gender', 'age', and 'population number' are identifiers of notions. Usually such a link automatically determines the set of possible values of the variable: the notation *v*:*n* means that variable *v* represents instances of the notion *n*, and these instances constitute the value set. Note that instances of attributes are their values.

A variable may also refer to a notion-instance defined in the domain model. For such references other notations are used: *v*=*n* if *n* represents a standalone entity or *v element-of n* if *n* represents a set. The first case may be used for specifying invariant variables.

If a variable instantiates an attribute, it is necessary to specify the value(s) assumed by the arguments of the attribute. The value assigned to an argument should be an instance of the type specified for this argument. It is not required that this instance should be previously defined in the domain model. The definition may be done directly in the data index, for example, c_1: 'population number' | arg_1=*x*: 'population' | 'gender'="female", 'age'=[0,14], where arg_1 is the name of the argument. This example demonstrates how an argument is instantiated with a constant expression. However, when an argument of a characteristic variable varies through the data set, as in our example in table 1, an expression with reference variables should be specified: c_1: 'population number' | arg_1=*x*: 'population' | 'gender'= r_2, 'age'= r_3. This scheme of argument specification does not depend on whether values of the parameters are contained in data tuples or the characteristic variable corresponds to several positions in the tuples implicitly referring to different combination of parameter values.

If a characteristic variable refers to an entity, it should be linked to at least one of the parameters with a predefined relationship of the schema or a domain-specific relationship defined in the domain model. For example: r_1: 'district'; c_1: 'sector' | *part-of*(r_1,c_1); c_2: 'borough' | *part-of*(c_1,c_2). Note that the characteristic variable c_2 is not directly related to the parameter r_1. However, due to the transitivity of the relationship *part-of*, it may be inferred that *part-of*(r_1,c_2).

When data are described by reference to **compound attributes**, two variants of representation of values of a compound attribute in data should be distinguished:

1. Each component of a value is contained in a separate position of a data tuple. Thus, for the attribute ‘date’ with components ‘year’, ‘month’, and ‘day’ a possible situation is when years are stored in the n-th position of each tuple, months in the (n+1)-th, and days in (n+2)-th position.
2. All components of a value are “packed” in a single string having a certain format, and only one position is needed to contain the whole value. Thus, values of the attribute ‘date’ may be represented as strings xx.yy.zzzz, where xx are two digits showing the day, yy encodes the number of the month, and zzzz are four digits representing the year.

The correspondence function is specified differently in these two cases, for example:
1. corr(n)=v.‘year’, corr(n+1)=v.‘month’, corr(n+2)=v.‘day’ and
2. corr(n)=“xx.yy.zzzz” | x= v.‘day’, y= v.‘month’, z= v.‘year’,

where v is a variable related to the compound attribute ‘date’.

Fig. 3 shows schematically an example data set characterization.

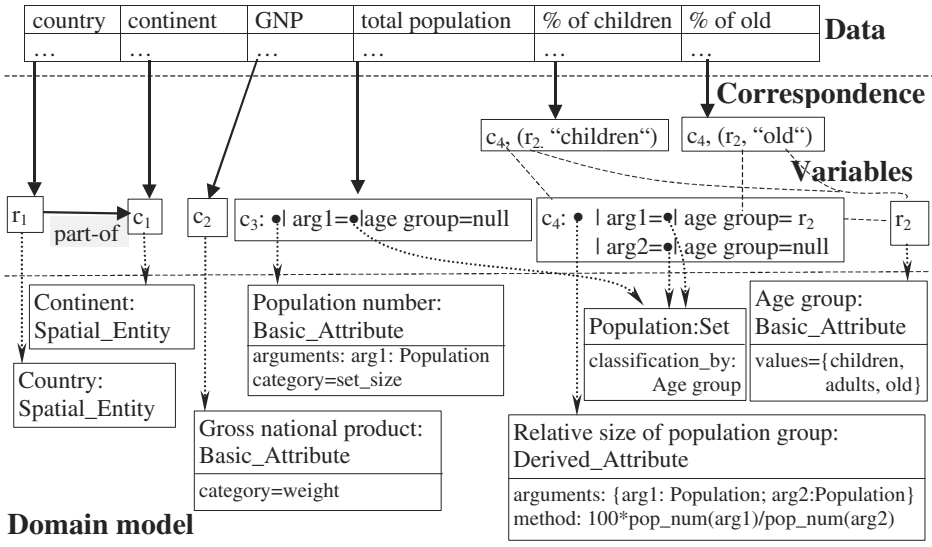

Fig. 3. A graphical representation of an example data characterization. Columns of the table are referred to parameters r_1 and r_2 and characteristic variables c_1, c_2, c_3, and c_4. The variables, in their turn, refer to appropriate domain notions

5 Conclusion

The presented data characterization schema is proposed as an appropriate and sufficiently powerful mechanism of formal description of data sets for the purposes of intelligent software support in visual exploration of spatially referenced thematic data. Extended representation of data semantics not only supports valid map design

solutions but also enables an intelligent assistance to the user in data analysis. This assistance includes communication with the user about her/his analytic goals, appropriate data transformations, calculation of useful statistics, and building visual displays well suited to the goals.

We realize that data description according to the schema is not an easy task for a data provider. Therefore it is foreseen that an interactive tool will be developed that acquires the domain model and data index by interviewing a person who knows the meaning of the data. The tool hides from the user all the formalisms and interacts with her/him using usual lexicon and convenient graphic interface. A prototype of such a tool already exists as a module of Descartes called Application Builder. This program proposes to the user to group data components with similar meanings, and to indicate the common concept and the differences. To express the commonality or the difference, the user introduces a new notion or reuses a previously defined notion. In this way the domain model and the data index are built in parallel.

Acknowledgements

We thank our CommonGIS partners, in particular Frank Tuijnman (Professional Geo Systems, the Netherlands), Ursula Kretschmer and Uwe Jasnoch (Fraunhofer IGD, Germany), and Hans Voss (GMD, Germany) for fruitful discussions.

References

1. Andrienko, G. and Andrienko, N. (1998) Intelligent Visualization and Dynamic Manipulation: Two Complementary Instruments to Support Data Exploration with GIS. In *Proceedings of AVI'98: Advanced Visual Interfaces Int. Working Conference* (L'Aquila – Italy, May 24-27, 1998), ACM Press, pp.66-75.
2. Andrienko, G. and Andrienko, N. (1999a) Making a GIS Intelligent: CommonGIS Project View, In *Proc. AGILE'99 Conference* (Rome, April 15-17, 1999), pp.19-24.
3. Andrienko, G. and Andrienko, N. (1999b) Interactive Maps for Visual Data Exploration. *International Journal Geographical Information Science*, **13** (4), accepted.
4. Bertin, J. *Semiology of Graphics. Diagrams, Networks, Maps*. The University of Wisconsin Press, Madison, 1967/1983.
5. Jung, V. (1995) Knowledge-based Visualization Design for Geographic Information Systems, in *Proc. 3rd ACM Int. Workshop on Advances in GIS* (Baltimor), ACM Press, pp.101-108.
6. Klir, G.J. (1985) *Architecture of systems problem solving*. Plenum Press, NY.
7. MacEachren, A.M. (1995) *How Maps Work: Representation, Visualization, and Design* (NY: The Guilford Press)
8. Mackinlay, J. (1986) Automating the Design of Graphical Presentation of Relational Information. *ACM Transactions on Graphics*, **5** (2), 110-141.
9. Roth, S.M. and Mattis, J. (1990) Data Characterization for Intelligent Graphics Presentation, in *Proc. SIGCHI'90: Human Factors in Computing Systems*, (Seattle), ACM Press, pp.193-200.
10. Senay, H. and Ignatius, E. (1994) A knowledge-based system for visualization design. *IEEE Computer Graphics and Applications*. **14** (6), 36-47.
11. *Unified Modeling Language Notation Guide.* URL http://www.rational.com/uml/resources/