Formal map specifications for a National Web Atlas

HELOISA GABRIEL PINHEIRO

March, 2017

SUPERVISORS:

Drs. B.J. Köbben

Dr. Ir. R. A. de By

Formal map specifications for a National Web Atlas

HELOISA GABRIEL PINHEIRO

Enschede, The Netherlands, March, 2017

Thesis submitted to the Faculty of Geo-Information Science and Earth Observation of the University of Twente in partial fulfilment of the requirements for the degree of Master of Science in Geo-information Science and Earth Observation.

Specialization: Geoinformatics

SUPERVISORS:

Drs. B.J. Köbben

Dr. Ir. R. A. de By

THESIS ASSESSMENT BOARD:

Prof. Dr. M. J. Kraak (Chair)

Dr. Ir. R. J. A. van Lammeren (External Examiner, Wageningen University & Research, Laboratory of Geo-Information Science and Remote Sensing)

DISCLAIMER

This document describes work undertaken as part of a programme of study at the Faculty of Geo-Information Science and Earth Observation of the University of Twente. All views and opinions expressed therein remain the sole responsibility of the author, and do not necessarily represent those of the Faculty.

ABSTRACT

For the generation of maps, a cartographer needs to know the data in advance in order to select the best map type to represent the data. The aim of this research was to formally specify of the process of selecting a map type, based on data characteristics to allow the automatization of the process. We studied as other authors described the process of representing data, to determine the necessary steps to generate a map. Then an informal description of these step was performed. The selection of the formal specification language was made through a study of the different types and characteristics of them. And finally, the process of selecting a map type based on data characteristics was formally specified for the study case of the experimental National Dutch Web Atlas. The properties from maps that should be specified are the visual variable and the geometry. The characteristics of the datasets that needs to be specified are the measurement scale, the attributes, and the geometry of the geographical component. Among the user's requirements properties are the necessary for the selective perceptual property, the number of classes and the user's preference for visual variable and geometry that makes the mark visible. The effectiveness and expressiveness of each visual variable and geometry also needs to be specified as a system knowledge. These properties are in Component, Algebra, Scale, Summary, Aesthetic and Geometry steps. The language selected for specifying the process is the Z specification language. And the process of selecting a map type based on the data characteristics was specified to the experimental National Dutch Web Atlas. The process was informally described for the generation of maps with one variable and for discrete data. The formal specification was executed using the Z language. Z is not ease to use, but will allow further specification of the process of generating a map. The process of selecting a map type based on data characteristics, taking into account certain user's requirements was specified for the experimental National Dutch Web Atlas. However, further automation of the process still needs to be done.

Keywords: Formal specification language, z language, data representation

ACKNOWLEDGEMENTS

Eu gostaria de agradecer a Deus por ter me mantido forte durante essa fase que não foi fácil. Aos meus pais por terem sempre me empurrado para frente e acreditado em mim até mesmo quando eu não acreditava. Ao meu marido por nos colocar em primeiro lugar, me mostrando o quanto somos importantes. Ao Exército Brasileiro por me dar essa e tantas outras oportunidades na vida. Aos meus queridos vizinhos que me encorajaram e agora estão "segurando as pontas" da minha vida que ficou no Brasil. Ao Diogo que teve a brilhante idéia de fazermos o curso preparatório para o Toefl. I also would like to thank Drs. B.J. Köbben for "untangling" my thoughts; Dr. de By for helping with Z; Julia Leventi for the support at the end of the core module, it was really important to me; my ITC friends for being so kind, you all made this process unique and really valuable. A todos os meus amigos que sempre torceram por mim, até mesmo sem saber para o quê estavam torcendo. E a Maria, meu lindo presente. Thank you. Obrigada!

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

1. INTRODUCTION

1.1. Motivation and problem statement

In the past years, maps became more popular, due to navigation GPS, and products, such as Google Earth and Google Maps, among other factors. These maps are easily accessible on cell phones, and in a growing number of applications on the internet. As a consequence, along with the free software movement the quantity of free Geographic Information System (GIS) software that also became available increased. And, as a result, people got used to it, and they want to answer "what" and "where" questions and make decision based on spatial data (Basaraner, 2016).

This enhanced the role of spatial data infrastructure (SDI), which was created to "facilitate the availability of and access to spatial data" (Global Spatial Data Infrastructure, 2004). Basaraner (2016) identified three SDI generations. The first two, the data-centric SDI and the process-centric SDI, were concentrated in the data acquisition, modelling, compellation, processing, analyze. In its thirds generation, SDIs are more focused on the user, which implies more importance to visualization, dissemination, access and sharing of data (the user-centric SDI).

A web map service is used since it returns a map representing the spatial data (Open Geospatial Consortium Inc., 2006). These services use a spatial data and a service configuration as input. The latter defines how the data should be portrayed, for example which map type, symbol, colors to be used and what are the data intervals.

However, nowadays the service configuration is set-up by a cartographer, who needs to know the data beforehand to be able to choose the appropriate map to represent the data, taking into account the cartographic design rules, and the correct perception property of the data.

The graphic design rules were introduced by Bertin (1983), who presented how the data should be represented taking into account its measurement level. This theory is used by cartographers to enhance people's perception about the data which is been presented. And the wrong use of the perception property can lead to a false impression of the data and then to an incorrect decision.

The problem is that there is at present no method to automatically generate the mentioned configuration service, a method that depending on the data could set the correct type of map, and create a map with cartographic design decisions included (Köbben, 2013).

This method could also benefit web services applications which use as input different sources and generate a map as output. Some of these services use an automated script to retrieve data from the source, but the cartographic decisions are still being made by a human operator, who configures the service to properly display this data in the browser, and has to rebuild the service whenever there is some important change in the data (Köbben, 2013).

1.2. Research identification

In this section the issues of this research are presented through its objectives and questions.

1.2.1. Research objectives

In order to make possible the automated creation of a map service configuration, we need first to informally describe this process. Even though, it is a detailed decription it is not machine readible. On the other hand formally specifying the process of choosing a map type for a specific data set can lead to an unambiguous description of the process and allow this automation of the process.

Thus, to achieve this main objective of this research, which is the formalization of the process of selecting a map type based on the data characteristics, taking into account some user's requirement, the following objectives have been formulated:

a) Research if, and how, formal specification language is able to describe the process of choosing a map type for a specific data set, taking into account certain user requirements;
b) Select or develop a formal language specification to describe this process;
c) As a proof-of-concept, describe the process of generating the maps from the experimental National Dutch Web Atlas using formal language specification.

1.2.2. Research questions

In order to achieve the objectives mentioned before, the following questions have to be answered:

a) What is formal specification language?
b) What are the types of formal specification language?
c) Can formal specification language describe the process of choosing a map type for a specific data set, taking into account certain user requirements?
d) Is there already a formal specification language that can be used to do this, and if so, which language is this?
e) Which are properties of the maps, data sets and users that should be specified, and how can they be specified in the chosen language?
f) Which are the (transformation) processes from data to maps that should be specified, and how can they be specified in the chosen language?

1.2.3. Innovation aimed at

This research project aims in proving if, and how, formal language can be used to describe the process of choosing a map type for a specific data set, taking into account certain user requirements in order to allow automatic map generation. And in order to specify the process a specification language is chosen, and the process is described using this language.

1.2.4. Related work

Some authors formalized some map characteristics, but not all of them used a formal language specification to do this.

Balley et al. (2014) present three different study case. In the first one, they produce map specifications for generalization based on users' requirements, but they do not use a formal language to do this. In the second one, they present a map specification model for on-demand mapping, based on an already existing data. On the third one, they present a solution to select colors to a map from the users' requirement. However, they do not come up with a formalization of this map specifications, because they were not able to formalize some cartographic knowledge.

Nelson, Alencar, & Cowan (2001) use Z formal specification language to demonstrate that it is possible to specify and verify map-centered applications. They informally describe some elements, properties and relations in order to formalize the geographic space. And then specify using Z a system which is composed of coordinates and resources, and operations as finding a resource, and adding a resource.

Reimer (2015) modelled the design principles of maps from labeling, schematization and generalization for automated mapping. He used a mathematic approach to do it.

This thesis uses formal language specification in order to specify the process of choosing a map type for a specific data set, taking into account certain user requirements to allow automatic web mapping in an SDI environment.

1.3. Project set-up

This section will describe how the goals of this project will be achieved.

1.3.1. Method adopted

The methodology will follow the flow illustrated in Figure 1.

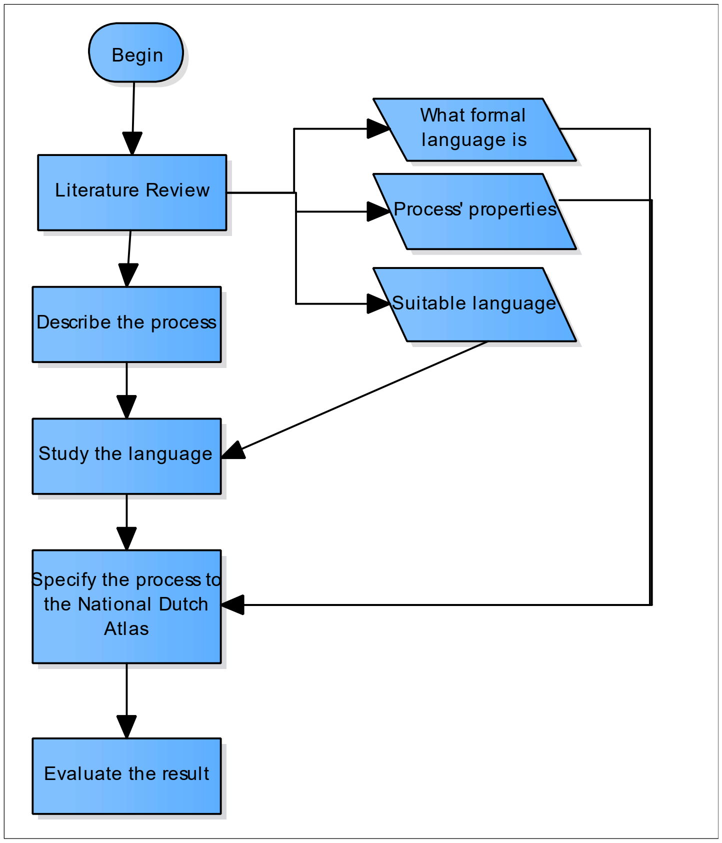


Figure 1: Methodology's flow chart.

The research started with a literature review about the properties of map, data set and users' requirements that should be specified in order to describe the process.

Then another a literature review was performed in order to determine what formal language specification is, what are the available formal specification languages, what are the different types, which of those types are the suitable for the problem, what are the characteristics of the available ones, which language seems to be most suitable to describe the process of choosing a map type for a specific data set, taking into account certain user requirements.

The first literature review allowed us to describe the process of choosing the map type based on data characteristics. An informal description of the process is necessary before formally specifing it.

After choosing one of the suitable languages, a study about it was done, in order to enable its use.

Then using the outcomes from the literature reviews and description, the process of choosing a map type for a specific data set, taking into account certain user' requirements, was specified using the chosen language and was implemented for the use case of the experimental National Dutch Web Atlas.

Finally, the results of this spefication are evaluated and a discussion about the specification of the process is presented.

Since formal specification should lead to an unambiguous specification of the process, this formalization and the formalization of the users' requirement should allow an automatic generation of the service configuration to generate maps from data sets.

1.4. Thesis structure

Chapter 1: The first chapter of this thesis introduces the motivation and problem statement, the research objectives, questions and innovation and also the project set-up.

Chapter 2: The second chapter presents a literature review about the properties of the process of choosing a map type for a specific data set, taking into account certain user requirements and formal specification language in order to make it possible to select one.

Chapter 3: The third chapter describes the process of choosing a map type for a specific data set, taking into account certain user's requirements.

Chatper 4: In this chapter the process of choosing a map type for a specific data set, taking into account certain user's requirements for the experimental National Dutch Atlas is formally specified.

Chapter 5: And the last chapter the conclusions and recommendations are described.

2. LITERATURE REVIEW

To specify the process of choosing a map type based on the data characteristics, the literature review was divided into two parts. The first part of this chapter presents a literature review about the steps that are taken in the process to represent the data graphically, describing the entire process. The second part of this chapter is also a literature review, but about formal specification language, to enable us to make a better selection of a language to specify the mentioned process.

Maps are an interface for the exploration of geospatial data (Kraak & Ormeling, 2010). Cited by Shneiderman (1996) as 2-dimensional data type among the seven data types defined by him, namely 1-dimensional, 2-dimensional, 3-dimensional, temporal, multidimensional, tree and network. Although it could be considered in one of the other categories, as for example, 3-dimensional or even a network user interface or even the ones mentioned but not took into account in his article as 2 ½-dimensional.

And as users interface, maps might be used to accomplish different tasks as for example the ones also categorized by Shneiderman (1996) overview, zoom, filter, details-on-demand, relate, history, and extract.

Overview for the user to have an insight about the data collection. Zoom for focusing in the interested items. Filter for remove unwanted piece of information. Details-on-demand to allow the user to have more information. Relate for enabling comparison. History to trace back the actions. Extract in order to permit the user to take the pieces of information that he/she is interested in. And through his Visual Information Seeking Mantra, he argues that overview should be executed first, so the user can observe the distribution of the entire collection.

In this way, the specification of the process of choosing a map type from the data characteristics presented in this research are the maps with a geographical component and a statistical component and the user task is to have an overview of the data, and an insight into the distribution of the phenomena. Figure 2 displays the distribution of the density population through the Overijssel, this is an example of the maps specified in this study. The geographical component which is the map unit is the municipalities of Overijssel and the statistical component is the population density, which is the measurement that vary depending on the map unit.

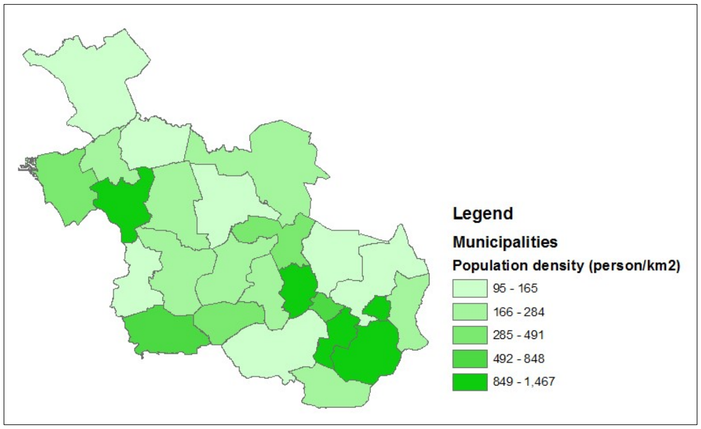


Figure 2: Population density in Overijssel by municipality. Data from Province Overijssel (2016).

The overview task with just one statistical component is also important since maps with more than one statistical variable - or for a different task - can be described as a combination of multiple map types. For example, the map of Figure 3 compares population density of all municipalities against that of a specific municipality (in this case, the municipality of Twenterand), using a bi-polar colour progression map, representing two different forms of information. The first form addresses population density, while the second form differentiates between municipalities with higher or lower densities than that of the chosen municipality.

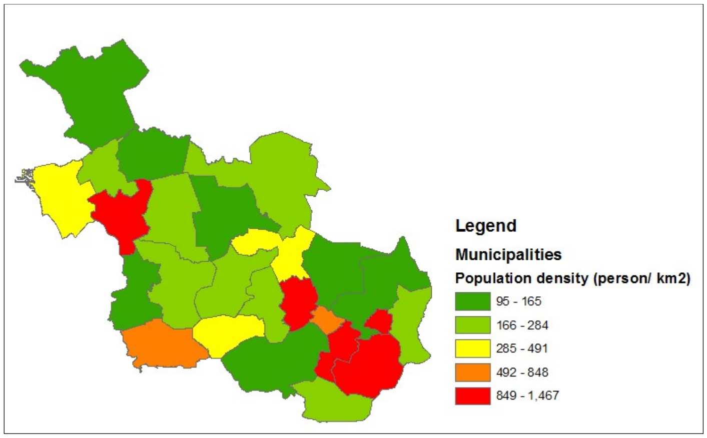


Figure 3: Population density in Overijssel compared to that of a chosen municipality (Twenterand). Data from Province Overijssel (2016).

2.1. The process from data characteristics to map type

In this subsection a literature review of processes to represent data that are based on data characteristics is presented. This results in a selection of steps for the specification of the process of creating a map, that is used latter to the choice of a map type.

Mackinlay (1986) separate the graphical primitives (bar chart, line chart, horizontal axis) by encoding technique (single-position, apposed-position, retinal-list, map, connection and miscellaneous). And then described which type of data each encoding technique could represent. The expressiveness criteria described insures that quantitative data is represented by size, for example.

Then he ranked the perceptual tasks, which says that for example position is the best way to encode nominal data, followed by colour. In order to be effective, one should order the components by importance, and the most important component is represented by the best perceptual task. Which is the called effectiveness criteria.

For the effectiveness criteria to be better applied to the graph as a whole, the importance ordering principle is defined. This principle represents the more important information by the perceptual task that better express it.

And for combining different designs, Mackinlay defined what he called “principle of composition”. This principle arranges together two designs by combining the common parts.

Therefore, following the criteria and principles mentioned, Mackinlay’s process has three steps, namely partition, selection and composition. First, it performs the partitioning step, which consists of ordering the variables by importance and then separating them, and as in Wilkinson (2005), also a set of variables is possible.

Secondly, the selection step, in which, based on the criteria of expressiveness and effectiveness, for each partition, a set of primitive languages possible to represent the variable is generated. The possible values are: single-position, apposed-position, retinal-list, map, connection and miscellaneous. Then the visual variables are ranked for each attribute, based on their measurement scale.

Thirdly, the composition step is performed to check whether the suggested designs per attribute can be applied together. If the principle of composition is not possible to be applied this lead to a backtrack in the primitives and in the visual variables, using the principle of importance ordering. For example, if two variables are suggested to be represented on the horizontal axis, the variable with a lower position in the hierarchy, defined by the ordering partition step, is assigned to the second design choice. Figure 4 gives an overview of the process.

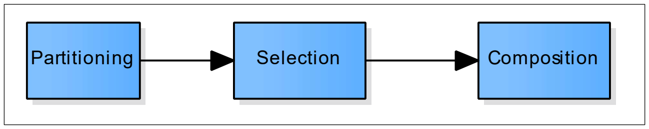


Figure 4: Graph creation process by (Mackinlay, 1986).

Mackinlay also take into account effectiveness of the medium where the graph is going to be displayed. So for example it is a black and white printing, colour is not an effective way to be used and therefore, it also leads to a backtrack in the process and colour is not considered to represent the data.

For Wilkinson (2005), the process of creating a graph from the data follows the flow of Figure 5. In his approach, data representation should start with variables, followed by algebra, scales, statistics, geometry, coordinates and aesthetics.

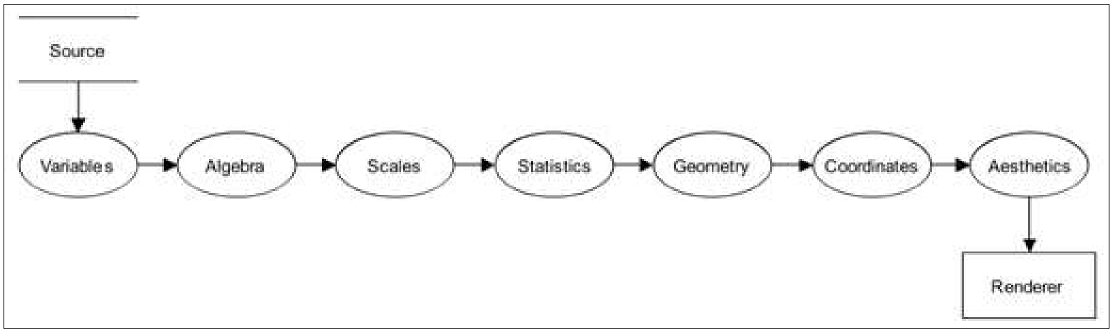


Figure 5: The process of creating a graph by Wilkinson (2005).

The *Variables step* consists of operations applied to data to create a variable set (called *varset*). Wilkinson argues that a column in a dataset could be considered a variable, while a varset is a set of one or more variables. The method allows to create other variables or summarizations, such as calculating a mean.

The *Algebra step* comprises operations applied to retrieve and create a combination of variables. Three operations are mentioned: cross, nest and blend. The cross (Cartesian product) creates tuples of variables. The nest operation relates variables conditionally, which means that the first value in the tuple depends on the value in the second tuple. Blend makes a union of different varsets. This terms are better explained in Subsection 3.3.

In the *Scales step* the scale used in the graph is specified and the required transformations on the data take place, for example, determine the log() of numeric data to use a log scale.

The *Statistics step* creates partitions of the graph space and calculates statistics (like mean, median), for the partition or the object. An example of this step is provided in Subsection 3.5.

Geometry step creates the geometric graphs. These graphs are a subset of all the possible ones, and in this concept points, areas and lines that latter receive one of the aesthetic function in order to become a graphic.

Coordinates step are usually sets of tuples of numbers that locates the objects of the graph in the space. This step is also from transforming from one coordinate system to other.

And finally, the *Aesthetics step* transform a graph in a graphic. It uses Bertin's visual variable as functions in addition to transparency, blur, motion, sound and text.

Kraak & Ormeling (2010) start by the identification of the invariant and components. An *invariant* should be understood as the "common descriptor of all elements" (Kraak & Ormeling, 2010). On the other hand, *components* are the aspects that vary depending on the data element.

Then the components are assessed as which of those is the geographical component, and the measurement level, length and range (if interval or ratio data) of the remaining components are determined. The *length* is the number of distinct values that the attribute has and *range* the set of values an attribute can take.

After this, they select the graphical variable on the basis of the measurement scale. This process is illustrated in Figure 6.

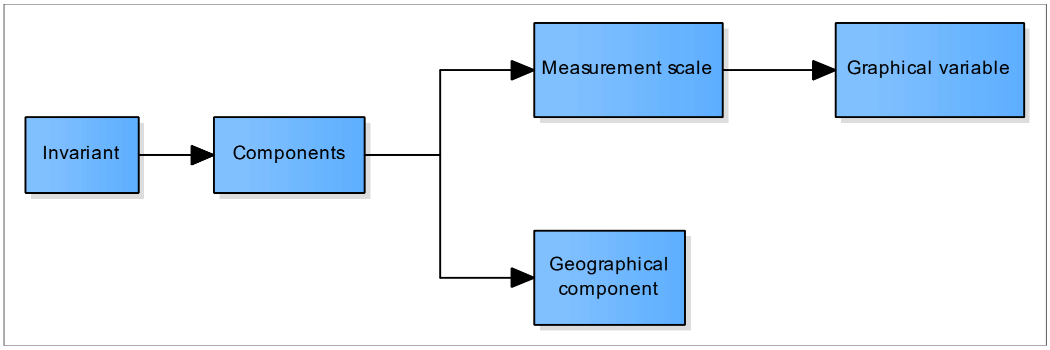


Figure 6: The map creation flow by Kraak & Ormeling (2010).

For the purpose of this research the graphical primitive is the map, since the most important task in a map is to answer the "where" question. Because of that the visual variable position is used to represent the geographical component. Furthermore, Mackinlay's (1986) "principle of importance ordering" should be considered if the maps to be generated have more than one statistical variable, which is out of the scope of this research.

Wilkinson (2005) description of the process from the dataset to a graphic representation of the data is the more detailed. However, his approach is not specific for creating a map, he does not mention classification of the data, for example. So in order to describe the process of creating a map the three approaches are merged.

In maps, it is important to know the *invariant*, because the background information can immediately tell the user where the phenomenon is. Figure 7 illustrates this importance, in the left figure without the background information and on the right with the Netherlands area as background.

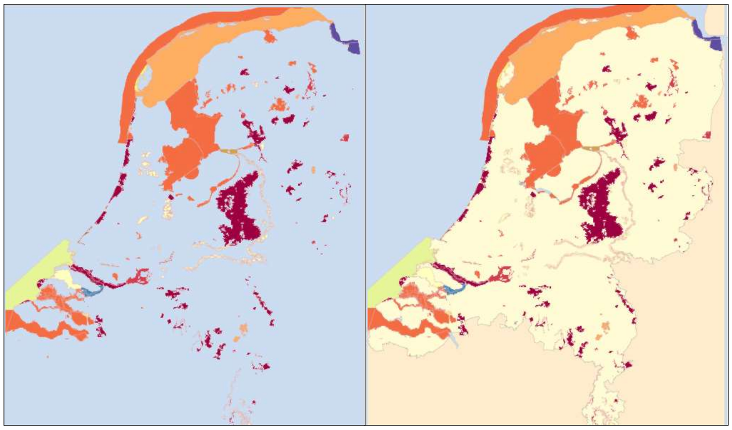


Figure 7: Protected areas in Netherlands, on the right with background information and on the left without. (Stichting Wetenschappelijke Atlas Nederland, 2013)

The step named *Variable step* by Wilkinson is renamed the *Component step*, because it includes the identification of the geographical component, as described by Kraak & Ormeling, which is another specificity of maps, since these geographical components are the map units.

The *Summary step* is included to provide to the system the length and range of the component as mentioned by Kraak & Ormeling (2010). The *Classification step* is also included in our description of the process and should be executed after the *Summary step*, since if classification is necessary, this needs as input the length and range.

Wilkinson used the Geometry to create the geometries and then he used the Coordinate steps to locate the geometries. For maps it is not always that the geometry that makes the mark visible is the map unit's geometry. An example is the proportional symbol map, in which a point mark visible is used to represent the area of the map unit. In this maps the map unit has the role of representing where the phenomenon is occurring, while the mark represents the phenomenon itself. And because of this, we separate these two steps.

The *Topography step* is for the background's and map units' geometry. Thus, using the feature coordinates the shape is created and then located.

The *Geometry step* is for the geometry that makes the mark visible. This step was moved to be performed after the Aesthetic step, because for the selection of the mark's geometry the visual variable needs to be determined before.

Based on the reasons explained above, the steps to describe the process of generating a map based on data characteristics are: Invariant, Component, Algebra, Scale, Statistics, Classification, Summary, Topography, Aesthetics and Geometry. The process flow is illustrated in Figure 8.

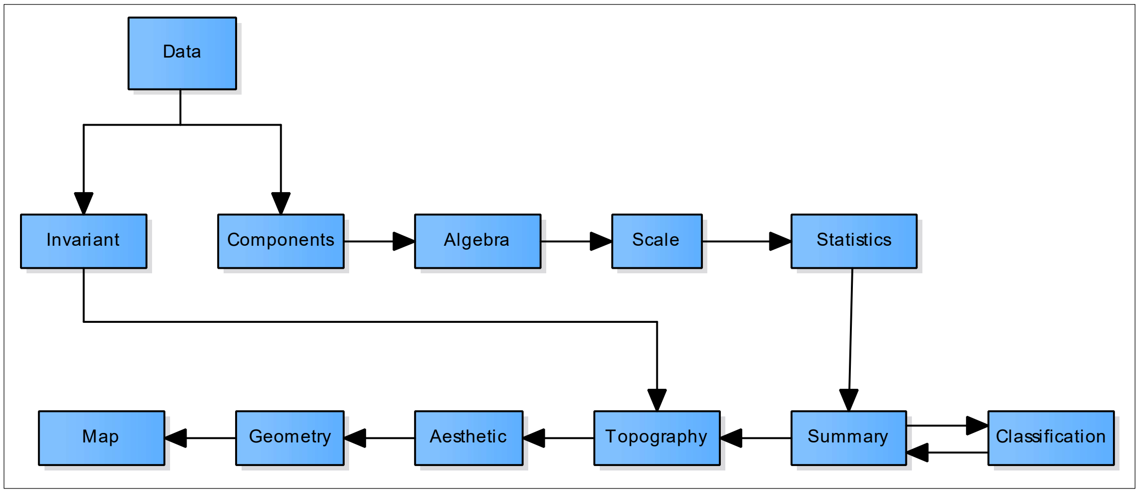


Figure 8: Chosen process to come from data characteristics to a map type.

The *Invariant step* starts the process to determine in which target geography the phenomenon to be represented occurs. Then, the *Component step* separates all components and their identification in the data set, because each phenomenon needs to be represented with the correct selection of the visual variable. The following, *Algebra step* restores the data, since the components have been separated, some relations need to be restored to allow presentation. The *Scale step* serves the purpose of identifying the measurement scale of each component for optimal representation. The *Statistics step* is needed because some maps use statistical graphs to represent their components. The *Summary step* returns information needed for classification and also for the selection of the visual variable. As the length of a variable may change after classification, it will be performed before and after the classification. The *Classification step* is necessary for the user to distinguish between marks represented by the same visual variable. The *Topography step* creates spatial geometries to represent the invariant and the geographic component based on coordinates. The *Aesthetic step* is the method for selection of the visual variable that better represents a specific component. And the *Geometry step* is used to determine how geographical components are made visible.

The steps defined in this research are more detailed and specific to represent the data through a map than processes presented in the literature review, since it takes into account the generation of the background, map units and classification if necessary. They were defined for representing discrete data and, as mentioned before, one statistical variable by time. If two or more depiction of data are necessary, Mackinlay principle of importance ordering should be used. Chapter 3 discusses these steps in more details and Chapter 4 presents the formal specification of properties of the maps, data sets and users.

2.2. Formal specification language

In this subsection a literature review about formal specification language will be performed in order to enable separate them per type, select the most suitable type for describe the process of choosing a map type for a specific data set and then a suitable language.

Woodcock & Loomes (1988) define formal language in terms of alphabet and syntax. The alphabet show which symbols there are in the language and the syntax one how they work together.

They identify four different classes of specification, naming model-oriented, algebraic, process algebra and modal logics. According to them model-oriented or state-based is used to "specify sequential systems" and to model the state of the system, examples are Z and VDM. Algebraic is for specifying large systems and to allow re-use of components, among the languages that follow this approach are Clear, ACT ONE, Larch and OBJ. For concurrency there are LOTOS and occam, which belongs to the groups of process algebra. Modal logics is useful to represent a state which can vary over time, and one example of this class of language is tempura.

Sivey (1998) defines formal specification as the use of some mathematical notation to describe an information system with precision. Formal specification is supposed to say what a system should do instead of how.

For Lamsweerde (2000) "formal specification is the expression, in some formal language and at some level of abstraction, of a collection of properties some system should satisfy". For him a formal a specification should have syntax, semantic and proof of theory, in other words, rules for creating the expressions, for interpreting them correctly and for inferring information. He also stated that formal specification has a declaration part and an assertion part, in the first one, the variables are declared and in the second one the properties of the variable are formalized.

Among the systems he defined that could be specified are a descriptive model of a domain, the user interface or a model of a process. One of the advantages of using formal specification are the reuse of components, while the disadvantage is that it seems to be developed to programmers, instead of specifiers (Lamsweerde, 2000).

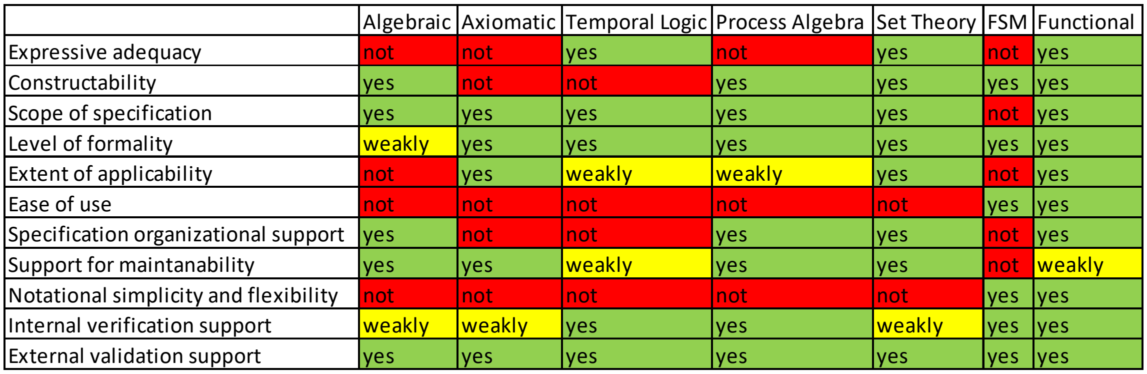
Lamsweerde (2000) classified the languages as history-based, state-based, transition-based, functional based and operational specification. History based specification when is it necessary to specify the different behaviour of the system through the time. State-based when the behaviour of the system in a point in time is to be specified, and the languages in this class are Z, VDM, B and object-oriented. Transition-based when the transition in the behaviour is to be specified, and the languages are statecharts, PROMELA, STeP-SPL, RSML, and SCR. Functional specification for using mathematical for specifying. He divided this group in two, naming algebraic specifications (OBJ, ASL, PLUSS and LARCH languages) and higher-order functions (HOL, PVS). The first one uses algebraic structures and the latter logical theories for specifying. And operational when a system can be described as a collection of processes (Paisley, GIST, Petri nets).

And in order to evaluate the types of specification language Lamsweerde (2000) proposed expressive power and level of coding required; constructability, manageability and evolvability; usability; communicability; powerful and efficient analysis. Expressive power and level of coding required because he defends that specification is not programing, and should be done for people who is aware about the problem, not necessarily a programmer, this language should specify properly the problem without hard coding. And in his conception, algebraic specification is the one that requires more coding and more knowledge. Constructability, manageability and evolvability is the capacity a language should have to be developed and changed in pieces or being possible to modify or increment it. State-based and functional languages are the best ones in this point. Usability concept defines that someone well-trained should be able to develop high quality specification. And this was observed for languages which have mathematical notion. Communicability means that some well-trained person should be able to understand the specification and evaluate it. Powerful and efficient when it is possible to achieve the objectives of formally specify, among them are reusability and deriving consequences.

For Khwaja & Urban (2010), specification is a description of a system's behaviour, while specification language is a method to achieve an objective. They start defining specification properties, specification language and environment properties. And based on these properties, they create some properties to evaluate the specification formalism classification.

They divided the formalisms into algebraic, axiomatic, temporal logic, process algebra, set theory, finite state machine specification and functional. And the properties that define each formalism are expressive adequacy, constructability, scope of specification, level of formality, extent of applicability, ease of use, specification organizational support, support for maintainability, notational simplicity and flexibility, internal verification support and external validation support. Table 1 summarises what are the specifications language properties supported by each specification formalism class.

Table 1: Specification languages properties supported by each specification formalism class (Khwaja & Urban, 2010).



	Algebraic	Axiomatic	Temporal Logic	Process Algebra	Set Theory	FSM	Functional
Expressive adequacy	not	not	yes	not	yes	not	yes
Constructability	yes	not	not	yes	yes	yes	yes
Scope of specification	yes	yes	yes	yes	yes	not	yes
Level of formality	weakly	yes	yes	yes	yes	yes	yes
Extent of applicability	not	yes	weakly	weakly	yes	not	yes
Ease of use	not	not	not	not	not	yes	yes
Specification organizational support	yes	not	not	yes	yes	not	yes
Support for maintanability	yes	yes	weakly	yes	yes	not	weakly
Notational simplicity and flexibility	not	not	not	not	not	yes	yes
Internal verification support	weakly	weakly	yes	yes	weakly	yes	yes
External validation support	yes	yes	yes	yes	yes	yes	yes

In this way, based on the classification given by Khwaja & Urban (2010), the only group that is not suitable for the description of the process of choosing a data type based on data characteristics is the algebraic specification, since it is not suitable for the process' description (Khwaja & Urban, 2010).

Analysing Table 1 the functional formal language group is the most reasonable choice, but the only language cited by the authors is Descartes. But, Khwaja & Urban (2010) mention that it is an executable language and did not clarify if it is an implementation language. As we could not find much material about Descartes to analyse it. Also some authors argue that implementation and specification should be "kept separated" since (as already mentioned before) specification says what a program should do, while implementation says how (Diller, 1990). And therefore we were not sure if it could be suitable for specifying the process.

On the other hand the set theory group defined by Khwaja & Urban (2010) is able to specify different domains, and they cite Z and VDM as languages from this group. And as such it has the characteristics of being "relatively independent from a specific data structure", therefore it can be applied in any domain, and the axioms can be used to represent the constraints.

According to Lamsweerde (2000), Z and VDM belong to the formalisms of the state-based group, and this group is based on pre- and post-conditions. It is used to specify sequential systems (Woodcock & Loomes, 1988).

The Z language is also able to describe the invariants and the changes in the system from state to state, the possible operations and the relationship between inputs and outputs (Sivey, 1998). Since depending on the data characteristics the map will have different states: For example, data will be represented as proportional point symbol if the data is absolute ratio, or as a chorochromatic map if it is a nominal data. The state-based group of languages, seems the more adequate to describe this approach. Also, as the data structure can differ from one map application to another and as the process itself is more relevant to this work than its data structure, a language of the type set theoretic satisfies this requirement.

In addition, in 2002 the Z formal specification language was adopted as a standard by the International Standard Organization (ISO)(Community Z Tools Project, 2016), and therefore has a lot of information. However, the adoption of Z for the specification leads in general to a guideline in how to develop a program instead of an automatic generation of the algorithm (Diller, 1990).

3. DESCRIPTION OF THE PROCESS OF CHOOSING A MAP TYPE FOR A SPECIFIC DATA SET

At the end of chapter 2 the steps of the process of generating a map from a specific data set was introduced. This chapter will present it in more details.

3.1. Invariant

Invariant is defined as the common denominator of all elements in the data (Kraak & Ormeling, 2010) or "the complete and invariable notion common to all the data" (Bertin, 1967/1983). It will be used in the title and also to determine the background of the map. Table 2 is the data from Province Overijssel (2016) used to generate the maps illustrated in Figure 2 and Figure 3. It is the population in the different municipalities of Overijssel in the year 2012. The invariant in this case is the "population in the province of Overijssel in the year 2012".

In this way, the background map should give the reader of the map an entire view of the province of Overijssel.

Table 2: Population in the province of Overijssel in 2012 (Province Overijssel, 2016).

Municipalities	Number of Inhabitants (person)
Almelo	72729
Borne	21770
Dalfsen	27570
Deventer	98581
(…)	
Twenterand	33971
Wierden	23807
Zwartewaterland	22139
Zwolle	122562

3.2. Components

Components are "variational concepts" (Bertin, 1967/1983), the attributes that vary according to the elements (Kraak & Ormeling, 2010). In Table 2 the municipalities and the population per municipality.

If the representation of more than one statistical component is desired, Mackinlay (1986) principle of importance ordering the components, should be used, so the most important attribute will be represented with the best visual variable for it. Likewise, the creation of a new variable using for example, quotient, sum and aggregation should be done in this moment, because the next step already deals with the combination of the variables (Wilkinson, 2010).

Also in this step the geographical component should be identified to allow the association between the component and its geometry.

From Table 2, municipalities are the geographical component and number of inhabitants as the other component. If there were the percentage of man and woman in this population, man's population, woman's population and also the gender would be the components, for example.

3.3. Algebra

Wilkinson (2005) described three operators to produce the combinations of variables, the cross, the nest and the blend. Those operators are used basically to retrieve the different sets of variables.

The cross (represented by * sign) and the nest operator (represented by / sign) creates tuples of variables, the main difference between them is that the first one creates just one domain, while the second one creates two domains (Wilkinson, 2005). The examples below taken from his book show the difference.

In Figure 9, Wilkinson (2010) crossed three variables, naming group (cities in USA or world), cities and population 2000 (city*population*group). As it is possible to observe cities domain contains all the cities in the table. Using the cross operator, it is possible to arrange together the latitude and longitude values creating a tuple, for latter locate this a point in a graph where the horizontal axis corresponds to the latitude values and the vertical axis, to the longitude, for example.

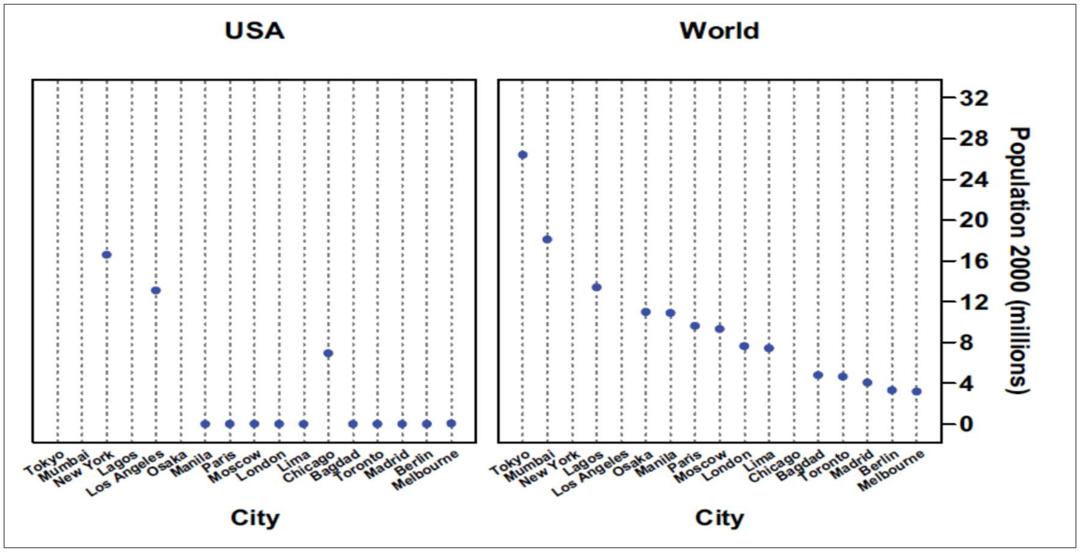


Figure 9: Crossed variables (Wilkinson, 2005).

On the other hand, if one wants to illustrate the graph with different domains, as for example, suppose that there is a data set with the population of different cities, and someone wants to illustrate the municipalities in different provinces in different maps, so the nest operator is useful. An example was taken from the book to illustrate it, first the variable city is nested with group and then the result is crossed with population 2000 (city/group*population). And as can be seen in Figure 10, the two groups have different domains (Wilkinson, 2010).

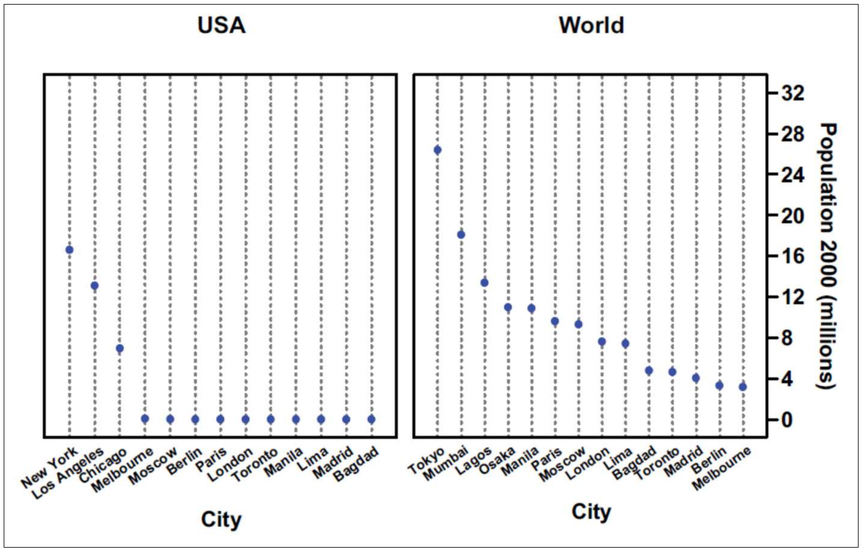


Figure 10: Nest operator (Wilkinson, 2005).

The blend operator (represented by the + sign) makes a union in different components, but tag each variable according to the column it come from. As illustrated in Figure 11, where city is crossed with the blended population 1980 and 2000 (city*(population 1980 + population 2000)). This operator can be useful for someone who wants to represent the population in different years in the same map (Wilkinson, 2010).

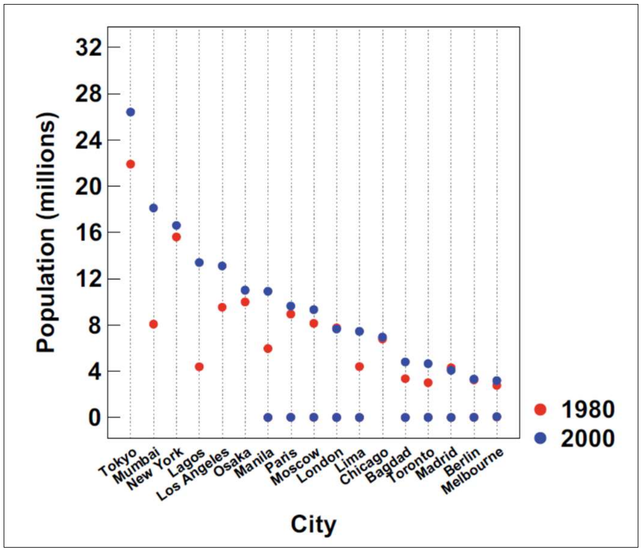


Figure 11: Blend operator (Wilkinson, 2005).

3.4. Scale

Stevens (1946) classified the scales of measurement based on the assignment of numerals to objects, in this way he identified four classes, naming nominal, ordinal, interval and ratio. And for the latter he divided it in two types the fundamental and derived scales.

Nominal scale was defined as the category in which the numerals are just labels, the order of the assignment of the numbers to the objects don't make sense (Stevens, 1946), or as described by Bertin (1967/1983) there is no universal order. And the only statistics possible is the number of occurrences (Stevens, 1946).

Ordinal scale can be ranked, ordered, the elements in this category have a universal order, but the distance between the objects in the scale is not known (Kraak & Ormeling, 2010).

On the other hand the interval scale provides the distance and order, but the zero is just a convention (Kraak & Ormeling, 2010). And statistics as mean, standard deviation can be performed (Stevens, 1946).

Ratio scale is for countable objects, this scale provides distance and order, and all types of statistical can be applied (Stevens, 1946). Fundamental ratio scales are "the result of the direct measurement" (Kraak & Ormeling, 2010), while the derived ratio scale are results of some mathematical function using the fundamental ones (Stevens, 1946). Nowadays they are more commonly known as absolute ratio and relative ratio respectively and these are the terms that will be used from now on. Also the term ordered will be used instead ordinal to avoid confusion.

So in this step, the measurement scale of the component should be assessed based as in the data characteristics as nominal, ordered, interval, absolute ratio and relative ratio. In the example given in the components subsection, population is absolute ratio, man's and woman's population are relative ratio and gender is nominal scale. An example of interval data is the temperature per city and of ordered data the classification of the population income as low, medium and high.

3.5. Statistics

The statistics method partitions the space and calculates statistics for the objects or the partition (Wilkinson, 2010). It can be used for example, when someone has data about the municipalities in the Netherlands, then the space can be partitioned in the different provinces and the statistics for each province calculated or if someone has data for each municipality for different years, then the mean and standard deviation can be calculated for each municipality. Performing statistics after *Scale step* is for additional information or for getting more information in order to perform a classification.

3.6. Summary

This step is necessary to determine the length (the number of classes or objects of a component), the minimum and the maximum of a component, and then derive the range of the component (Kraak & Ormeling, 2010).

The summary phase was included before the classification, because the latter phase needs the information generated in this step. And as the classification will change the length of the component and for the selection of the visual variable with selective property the length is necessary, this phase was also included after the classification.

3.7. Classification

If the measurement scale of a variable is ratio and the selective property is required, classification is necessary. For the classification method, the number of classes, the minimum, maximum and the length are necessary. The result of this method is the intervals in which the attributes of a component are grouped (Kraak & Ormeling, 2010).

There are two most used approaches for classifying the data, naming the graphic and the mathematical approach. The graphic has the break points, the frequency diagram and the cumulative frequency method. The mathematical has the equal steps, quantiles, arithmetic series, geometric series, harmonic series, and the nested means (Kraak & Ormeling, 2010).

For the graphical approach the idea is basically the same for all methods, for the frequency diagram method, the frequency diagram of the data is generated and the discontinues are used as class boundaries. For the frequency diagram method or the cumulative frequency diagram of the data and discontinues assigned as class boundaries. And for the break points method the data is sorted in the ascending order and plotted and again the discontinues are the class boundaries. The problems that can occur of applying the methods from this approach are for example the possibility of the number of discontinues be smaller than the necessary number of classes, or even there is no discontinues (Kraak & Ormeling, 2010).

For the quantiles method, the classes boundaries should be chosen with the purpose that each interval has the same number of observations. And for the nest method, first the average of all observations should be calculated and then average of the first part and the average of the second part and so on (Kraak & Ormeling, 2010).

For the other methods in the mathematical approach described here, the data should also be sorted plotted, as described for the break points method, and its curve compared with the theoretical curve. Figure 12 illustrates the theoretical curves.

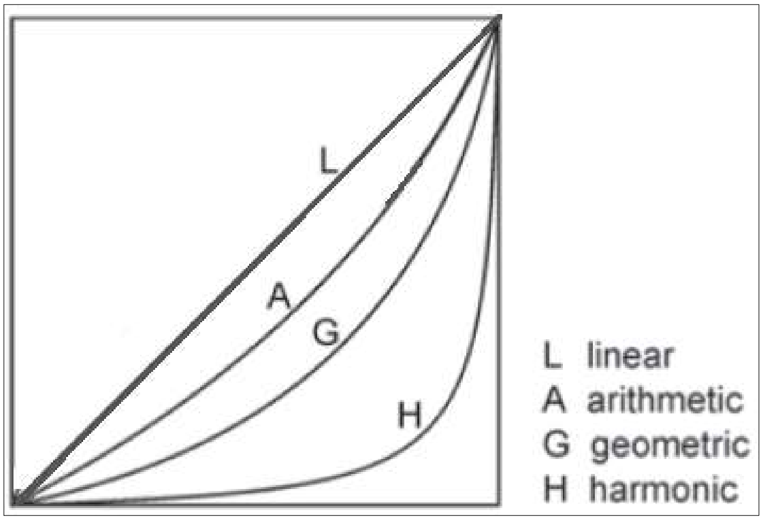


Figure 12: Theoretical curve (Kraak & Ormeling, 2010).

If the curve is similar to the curve represented by the letter L, the class boundaries should be calculated using the formula for equal steps, Equation 1. Using this method means that the intervals of the classes are the same (Kraak & Ormeling, 2010).

Equation 1: Class boundaries for equal step classification method (Knippers & Mank, 2015).

C = (Max – Min) x N
Class n boundaries:
FROM (Min + (n-1) x C) TO (Min + n x C)
	Where:
	C: Constant
	Max: highest value of the component
	Min: Lowest value of the component
	N: Number of classes
	n: number of a class

The approach should be arithmetic series, if the curve is comparable with A, and the class boundaries calculated through Equation 2. For this method the intervals of the classes are in an arithmetic progression (Kraak & Ormeling, 2010).

Equation 2: Class boundaries for the arithmetic series (Knippers & Mank, 2015).

C = ((Max – Min) x 2) / (N x (N+1)) Classes boundaries: Class 1: FROM (Min) TO (Min+C), Interval = C Class 2: FROM (Min+C) TO (Min+3C), Interval = 2C Class 3: FROM (Min+3C) TO (Min+6C), Interval = 3C	Where: C: Constant Max: highest value of the component Min: Lowest value of the component N: Number of classes

If the curve corresponds to the one labelled as G the geometric series method should be used. In this method the intervals are in a geometric progression. The boundaries of the classes should be calculated as Equation 3 (Kraak & Ormeling, 2010).

Equation 3: Class boundaries for the geometric series (Knippers & Mank, 2015).

$c = \sqrt[N]{\frac{Max}{Min}}$ Class boundaries: Class n: FROM (Min x C $^{(n-1)}$) TO (Min x C^{n}), interval = 10^{n}	Where: C: Constant Max: highest value of the component Min: Lowest value of the component N: Number of classes n: number of a class

And finally if the curve resembles the curve H, the data should be classified through the harmonic series method. Equation 4 shows how to calculate the boundaries of the classes (Kraak & Ormeling, 2010).

Equation 4: Class boundaries for the harmonic series (Knippers & Mank, 2015).

$C = \frac{\frac{1}{Min} - \frac{1}{Max}}{N}$ Class boundaries: Class n: FROM $\left(\left(\frac{1}{Min} - (n-1)x\ C\right)^{-1}\right)$ TO $\left(\left(\frac{1}{Min} - n\ x\ C\right)^{-1}\right)$	Where: C: Constant Max: highest value of the component Min: Lowest value of the component N: Number of classes n: number of a class

3.8. Topography

In this step the invariant is generated as an area based on its coordinates and the geographical component is generated as point, multipoint, line, multiline, polygon or multipolygon, also based on its coordinates.

3.9. Aesthetic

Bertin (1967/1983) has identified the perceptual properties, they are selective, associative, ordered and quantitative. The selective perception property enables the reader of a graphic to differentiate the categories. Associative enables grouping the similar objects. Ordered when the variable order is universal. And finally quantitative property is a characteristic of a variable when the distance between two ordered categories are measurable.

He also has identified the 8 visual variables, among those are the two planar dimension and the others are retinal variables, naming size, value, texture, colour, orientation and shape. Size is quantitative and dissociative. Value variation is not quantitative, but it is ordered and dissociative. Texture is ordered. Colour is not ordered, it is selective. Orientation is selective, but not for areas. Shape is only associative; it is not selective. And although almost all the visual variables have selective perception (except shape and area represented as orientation). And based on the perceptual property it is possible to define which visual variable is more suitable to represent a data with a specific measurement scale.

In this way, for representing nominal data, when the selective property is necessary point, line and area mark can vary in colour, also point and line mark can vary in orientation. However, if the selective property is not needed, the mark can vary in shape and area mark can vary in orientation (Bertin, 1983).

Ordered, interval and relative ratio data point, line and area mark can vary in value and texture. For value, the hue remains the same for all marks, but the value changes according to the ordered, interval or quantity in the observations or the classes, in the case of classified relative ratio. For texture this variation in the data, will be proportional to "the number of separable marks in a unitary area" (Bertin, 1983).

And absolute ratio data will be represented by point or line marks varying in size. The size of the mark will be proportional to the quantity of the component or to the average of the class, in the case of classified data (Kraak & Ormeling, 2010).

Mackinlay (1986) ranks the visual variables according to the perceptual property, because he was aiming in representing the most important component with the more effective visual variable, the principle of importance ordering. According to him position is the most effective way to represent the data. Length is the second most effective visual variable to represent quantitative data, but is not much effective to represent nominal data. Shape that is not suitable to represent quantitative or ordinal data, is able to represent nominal data. Figure 13 summarizes this information and present the rank for the other visual variables divided by the perceptual property. Visual variables inside the grey box means that they are not suitable to represent data with that perceptual property.

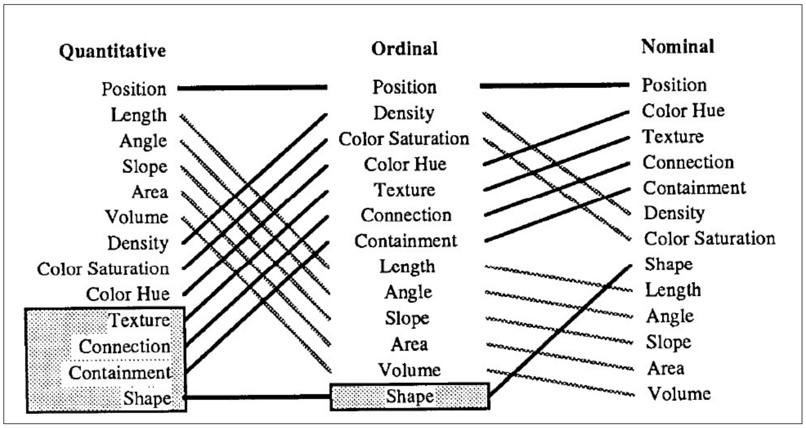
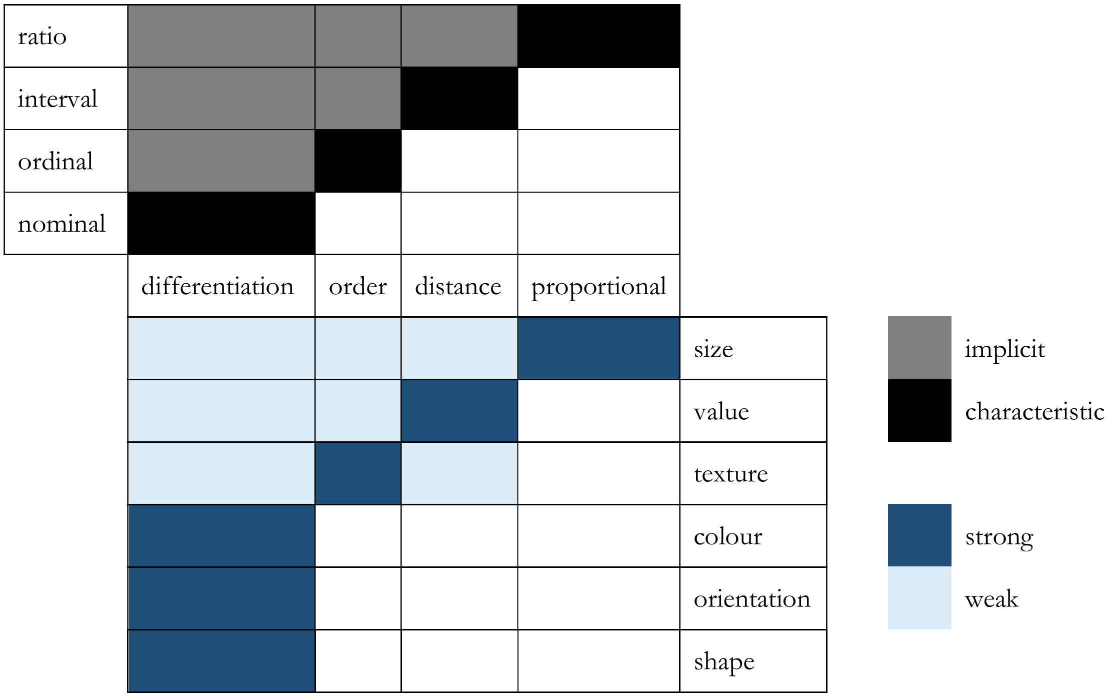


Figure 13: Rank of the visual variables according to the perceptual property (Mackinlay, 1986).

Wilkinson (2005) does not categorize the visual variable per measurement scale, he describes whether they are able to represent categorical or continuous variable. The aesthetic attributes for him are position, size, shape, rotation, resolution, brightness, hue, saturation, granularity, pattern, orientation, blur, transparency, motion, sound and text. Rotation, brightness, granularity are Bertins's orientation, value, texture, respectively. Bertin's texture is divided in granularity, pattern and orientation. Granularity being the number of shapes per space unit. Pattern being a change in the shapes for categorical variable and "increasing the randomness in a uniform spatial distribution" for continuous variables. And orientation is the change of orientation of the shapes. Resolution is defined as "the amount of information contained in its frame". Saturation is the change in the saturation of a colour, and to reflect uncertainty. Blur and transparency for "representing uncertainty".

According to Kraak & Ormeling (2010) there are also blur, transparency and focus. Blur can be applied to visualize uncertainty. Transparency to simulate the third dimension. And focus to attract attention. They also present which visual properties are more strong or weak to represent a certain property. As illustrated in Figure 14.



	differentiation	order	distance	proportional
ratio	▒	▒	▒	■
interval	▒	▒	■	
ordinal	▒	■		
nominal	■			
size	░	░	░	🟦
value	░	░	🟦	
texture	░	🟦	░	
colour	🟦			
orientation	🟦			
shape	🟦			

▒ implicit
■ characteristic
🟦 strong
░ weak

Figure 14: Visual variables characteristics (Kraak & Ormeling, 2010).

For specifying the process of choosing a map type based on the data characteristics, Bertin's concept will be used, but relative ratio components will be represented with ordered perception, since this approach is the most used (Andrienko & Andrienko, 1999).

3.10. Geometry

Bertin (1967/1983) identifies three classes of representation (which he also called implantation), naming the point, the line and the area, because he was working on the plane, but there is also volume that can be included to represent a phenomena (Kraak & Ormeling, 2010). For the specification of the process of choosing a map type based on data characteristics only the classes of representation of the plane will be considered.

Classes of representation make the mark which represents a variable visible. A point does not have an area, but the point mark which represents the point feature has. Marks can vary according to the retinal visual variables. However, area marks cannot vary in size (Bertin, 1967/1983).

However, if the selective perceptual property is needed for the distinguishing, the length of a visual variable is limited depending on the geometry that makes the mark visible. The length is the "number of steps" a mark with a determined visual variable and geometry can represent (Bertin, 1983). Table 3 shows maximum length for maintaining the selective perceptual property.

Table 3: The length of the visual variables for the selective perception (Bertin, 1983).

	Point	Line	Area
Size	4	4	5
Value	3	4	5
Texture	2	4	5
Colour	7	7	8
Orientation	4	2	

4. FORMAL SPECIFICATION OF THE PROCESS OF CHOOSING A MAP TYPE FOR A SPECIFIC DATA SET

In this chapter, first the datasets that are used to generate maps in the use case, named experimental National Dutch Web Atlas, are presented, then the process of choosing a map type for a specific data set for the use case is specified, using the formal specification language Z.

4.1. The experimental National Dutch Web Atlas (the use case)

The experimental National Dutch Web Atlas is an attempt to integrate the Dutch national geodata infrastructure and an interactive web atlas, to provide the NGDI with visualization (Köbben, 2013). Its maps are divided in three main themes: population, economy and nature. Each theme has various subjects, which are divided per map unit, moreover, the user can select the year of interest. The *subjects* are number of inhabitants, number of males, number of females, population density, single person households, CBS code and province name for the population's theme; gross domestic product, total salary and average income per inhabitant for the economy's theme; and protected areas for the nature's theme. The *map unit* is generally municipality or province, except for nature which can only be represented per protected area.

The number of inhabitants, males, or females have integer values and total salary have real values. And they are absolute ratio data.

The population density, the percentage of the single person households, gross domestic product and average income per inhabitant are relative ratio data. The first ones have integer values, while the last has real values.

CBS code, province name, protected areas are nominal data. All of them are represented by string values.

The experimental National Dutch Web Atlas is composed of proportional symbol maps, choropleth maps and chorochromatic maps. It can be accessed through the link http://www.nationaleatlas.nl/.

Table 4 summarizes the information above, showing the statistical component and its measurement scale, the value type and the map type that is used to represent it in the experimental web atlas.

Table 4: Experimental National Dutch Web Atlas data.

Subject	Measurement scale	Value type	Map type
Number of inhabitants	Absolute ratio	integer	Proportional symbol map
Number of males	Absolute ratio	integer	Proportional symbol map
Number of females	Absolute ratio	integer	Proportional symbol map
Population density	Relative ratio	integer	Choropleth map
Number of single person households	Relative ratio	integer	Choropleth map
CBS code	Nominal	string	Chorochromatic map
Total salary	Absolute ratio	real	Proportional symbol map
Gross domestic product	Relative ratio	real	Choropleth map
Average income per inhabitant	Relative ratio	real	Choropleth map
Protected areas	nominal	string	Chorochromatic map

4.2. Concepts needed

As a proof of concept we are going to specify the process of select the map type based on the data characteristics taking into account some user's requirement for the experimental National Dutch Web Atlas. This map type selection task is not the same as actually generating the maps, therefore not all the steps described in Chapter 3 needs to be specified. The necessary inputs, output and process knowledge needed are illustrated in Figure 15.

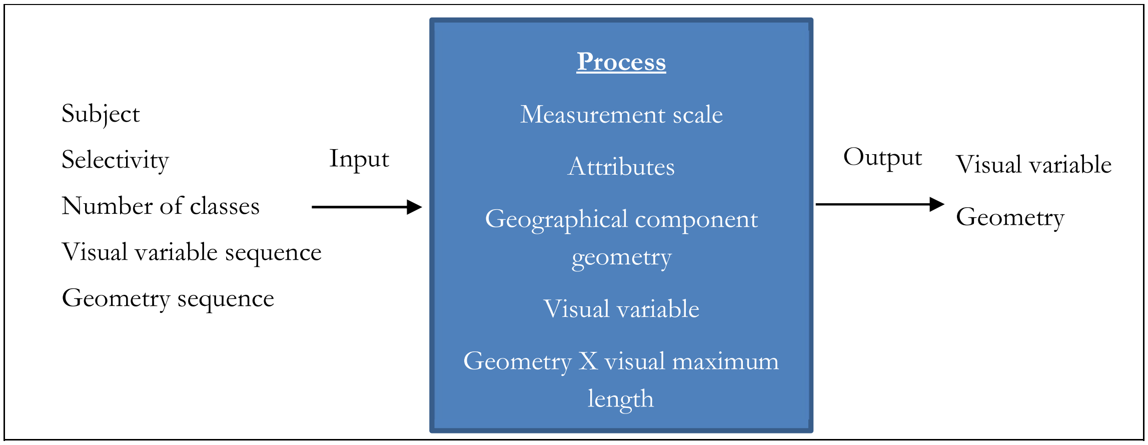


Figure 15: Input/output and process' knowledge.

The necessary inputs to this system are:

- the subject to be represented (one of the components that is not the geographical component)
- if the selective perceptual property is required
- the number of classes
- a sequence (list of) visual variables
- a sequence of geometries.

The subject is an input because it is what we want to represent through the correct map type. It needs to be in the database, because through the subject it is possible to retrieve its attributes and measurement scale, as well as the geographical component's geometry.

The number of classes and the selective perceptual property are modelled as inputs, because the maps in the study case are the maps from the experimental National Dutch Web Atlas and their information should be comparable from map to map. However, there are some maps for which the data is nominal, or where the number of classes is bigger than it is possible to represent preserving selectivity.

In creating a map usually different choices for visual variables and geometry are acceptable. But as we want to automate the process, we do not want to make it interactively, so we store the visual variables and geometries in predetermined preferences lists. One example is the list of visual variables, where we store the order of preference of the visual variables. Thus if colour, orientation and shape are the possible choices among the visual variables and the user list is in order of preference value, orientation, colour and size, then orientation is selected. Likewise, in the same way the mark's geometry is selected.

As briefly mentioned before, given the subject (input) the system knows the subject's measurement scale. Then if selectivity is necessary, which means the differentiation between different categories are necessary and if the data is ratio, the number of classes is used as the component length, otherwise it is the number of different attributes the subject has.

It is also a known to the system which of the possible visual variables are suitable to represent the data with a particular measurement scale. In this way, given the subject, through its measurement scale, the system gives the visual variables that are suitable to represent the input subject. For example, for a subject with nominal measurement scale the system returns colour, shape and orientation as the possible visual variables to represent the subject.

From the literature we know that if the selective perceptual property limits the length of the visual variable and this is also a known to the system. As a consequence, depending on the component length some visual variables are filtered out. The selected visual variable is one in the set generated by the system that is better ranked by the user through its list of visual variables. From the example above of the nominal data, in which colour, shape and orientation are possible, if selectivity is required, we know from the literature that the maximum length for preserving selectivity of the visual variable colour is 8, orientation is 4 and shape is 0. Thus, if the length of the component is 7, and the user's list of visual variable is in order shape, colour and orientation, the selected visual variable is colour.

Using the selected visual variable, and the length (if selectivity is needed) the set of geometries able to make the mark visible is generated. Since the system cannot suggest a line feature to be represented as an area and the system also knows the geographical component's geometry, the geometries that are not able to make the mark visible are filtered out. And as occurred for the visual variable, the set generated is compared to the user sequence of required geometries and the geometry in the set that is better ranked by the user is selected.

The output of the system is the map type most suitable to represent the data, taking into account certain user's requirements. For this model, "map type" is a combination of a visual variable and a geometry to represent the data. As for example, the combination of point and size, which is the proportional symbol map or area and value, which is the choropleth map.

4.3. Formalization problem statement

In literature we often find the problem statement, to exactly describe what we are going to formalize.

To represent a subject in a map, it is necessary to know its measurement scale, because based on it is possible to determine which visual variables should vary in order to better represent this data. Then the information about selective perception property is necessary to be known, since some visual variable cannot represent this type of data anymore, as for example, is the case of the shape visual variable, which can represent nominal data, except if the selective property is necessary. If the selective property is necessary, the length of the statistical component influences the selection of the geometry to make the mark visible, and so if it is a ratio data, this needs to be classified and number of classes selected becomes the component length. And then the visual variable and the geometry for the data is selected. Figure 16 summarizes the explanation above.

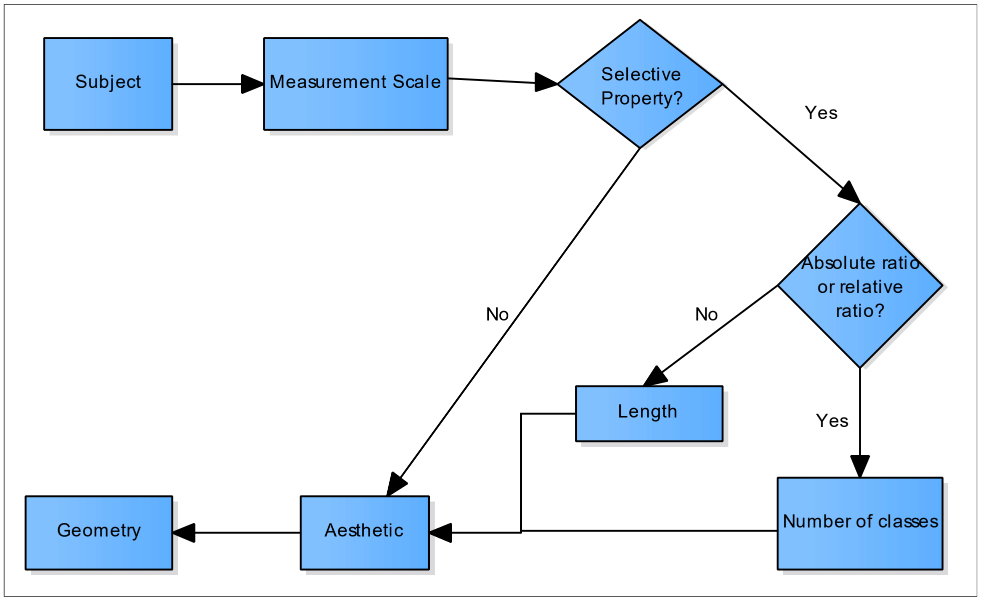


Figure 16: Prcess' flow from a subject to a map type.

4.3.1. Entities and types

To specify the necessary input, output and process variables, some data types need to be defined. A single subject is captured as an element under the enumerated SUBJECT type. The subjects for the national web atlas are ones listed in Table 4. For the sake of a complete example, a number of values and objects are included in the specification here. The true way to do this is to include all objects and values, but this is obviously highly unpractical in this text here.

SUBJECT::= *nr_inhabitants*_2011_*municipality* | *nr_inhabitants*_2013_*municipality* | *nr_inhabitants*_2011_*province* | *protected_areas*_2012

A subject has contents, which are the measurements itself, and we denote the measurement type as MEASUREMENT. The measurements of the use case are 157050, 72600, 80770, which are the values of the number of inhabitants for different municipalities for year 2011 in Netherlands, for example.

MEASUREMENT::= *v157050* | *v72600* | *v80770* | *v307080* | *v593050* | *v1130345* | *v158625* | *v72730* | *v80950* | *v321915* | *v616295* | *ONTWERP_2007* | *Natura_2000_besluit_2010*

These values are related to a municipality, for example 157050 is the number of males in the municipality of Enschede. These are the geographical components their type is OBJECT.

OBJECT::= *Enschede*| *Almelo* | *Hengelo*| *Utrecht*| *Rotterdam*| *Overijssel*| *Aamsveen*| *Broekvelden*| *Brunssummerheide*

To express the selectivity property, BOOLEAN is defined as enumerated type, and *yes* and *no* are its instances.

BOOLEAN::=*yes* | *no*

VISUAL is an enumerated type, and *shape*, *colour*, *orientation*, *size*, *value* and *texture* are the instances of this type. It is the type of one of the outputs and also the type of the elements of the sequence the user gives as an input to the system. It represents the set of values that the visual variable of a map can take.

VISUAL::=*shape* | *colour* | *orientation* | *size* | *value* | *texture*

GEOMETRY is an enumerated type which has *point*, *multipoint*, *line*, *multiline*, *polygon* and *multipolygon* as instances. The mark geometry of a map is the symbol used in the map to represent the variation in the measurements. Thus GEOMETRY is the type of the mark geometry and map unit geometry. One input to the decision procedure is a sequence of geometries with which the user expresses her/his preferences for map's mark geometry.

GEOMETRY::=*point* | *multipoint* | *line* | *multiline* |*polygon* | *multipolygon*

The SCALE is the type of all the possible subject's measurement scale it takes *nominal*, *interval*, *ordered, absolute ratio* and *relative ratio* as instances.

SCALE::=*nominal* | *interval* | *ordered* | *relativeRatio* | *absoluteRatio*

Finally, REPORT is an enumerated type of different output messages of the decision procedure that report the success or error of the procedure. The message *ok* signals that the procedure successfully ended. The error *big_number_of_classes* means that the user needs to select a smaller number of classes. Message *unknown_subject* signals a subject that is unknown to the system. Message *user_amplify_geometry* signals that the user selected a geometry that cannot represent the data because of its maximum length, while there is another geometry that can represent the data. Message *user_amplify_visual* signals that a visual variable can represent the data, but was not listed as preferred by the user.

REPORT::= *ok* | *big_number_of_classes* | *unknown_subject*| *user_amplify_geometry*
| *user_amplify_visual*

4.3.2. State Space

MapState describes the database system. It holds a set of *subjects,* and a set of *objects*. The *subject function* records the *objects* and its respective *measurements* of a *subject.* For every so referenced *subject,* the system also registers the subject's *unit geometry* and its measurement *scale*. For every scale, a set of *possible visuals* is known. For each implied combination of *visual* variable and *geometry* the *maximum length* that it can vary and maintain the selective perceptual property is known. These information is declared in the upper part of the *MapState* schema.

The lower part of the *MapState* schema describes the relationship among the variables declared in the upper part. The *subjectFunction* just registers the *objects* and *measurement* of the *subjects* that are known to the system (*dom subjectFunction = subjects*). The same holds for the *unitGeometry* (dom *unitGeometry = subjects*) and *measurementScale* (*dom measurementScale = subjects*) function. The *objects* from the *subject* that have a *measurement* are the ones in the database ($\forall s$: *subjects* • *dom* (*subjectFunction* (*s*)) ⊆ *objects*). For every measurement scale a subject has, the system knows the visual variables that are able to represent it (*ran measurementScale* ⊆ *dom possibleVisuals*). The measurement *scales* a *subject* can have are the ones that the system knows the *visual* variables possible to represent the *subject* (*ran measurementScale* ⊆ *dom possibleVisuals*). The set of *possible visual* variables that the system is able to suggest to a given measurement *scale*, are the ones that the system knows its *maximum lengths* ($\forall s$: *dom possibleVisuals* • *possibleVisuals s* ⊆ {*p: dom maximumLengths*• *first p*}).

$$
\begin{array}{l}
\_\mathit{MapState}\rule{20em}{0.4pt} \\
\mathit{subjects}: \mathbb{P}\ \mathit{SUBJECT} \\
\mathit{objects}: \mathbb{P}\ \mathit{OBJECT} \\
\mathit{subjectFunction}: \mathit{SUBJECT} \nrightarrow \mathit{OBJECT} \nrightarrow \mathit{MEASUREMENT} \\
\mathit{unitGeometry}: \mathit{SUBJECT} \rightarrow \mathit{GEOMETRY} \\
\mathit{measurementScale}: \mathit{SUBJECT} \rightarrow \mathit{SCALE} \\
\mathit{possibleVisuals}: \mathit{SCALE} \nrightarrow \mathbb{P}\ \mathit{VISUAL} \\
\mathit{maximumLengths}: \mathit{VISUAL} \times \mathit{GEOMETRY} \nrightarrow \mathbb{N} \\
\hline
\forall s: \mathit{subjects} \bullet \mathit{dom}\ (\mathit{subjectFunction}\ (s)) \subseteq \mathit{objects} \\
\mathit{dom\ subjectFunction} = \mathit{subjects} \\
\mathrm{dom}\ \mathit{unitGeometry} = \mathit{subjects} \\
\mathrm{dom}\ \mathit{measurementScale} = \mathit{subjects} \\
\mathit{ran\ measurementScale} \subseteq \mathit{dom\ possibleVisuals} \\
\forall s: \mathit{dom\ possibleVisuals} \bullet \mathit{possibleVisuals}\ s \subseteq \{p: \mathit{dom\ maximumLengths} \bullet \mathit{first}\ p\} \\
\hline
\end{array}
$$

4.3.3. Initial State of the System

For the start of the system, all subjects of the atlas, measurements, municipalities, provinces, protected areas, and years should be added. However, to demonstrate this start of the system just some of this information is actually added to the system, not all of them. For example, we added to the system the subjects number of inhabitants, and protected areas. We also entered the years 2011, 2012, 2013. The municipalities of Enschede, Almelo, Hengelo, Utrecht, Rotterdam; the province of Overijssel; and protected areas Aamsveen, Broekvelden, Brunssummerheide and its geometry. These values that initialize the system are used through the specification as an example.

Also, for launching the system the *visual* variables that can possibly represent a data with a determined measurement *scale* and also the *maximum length* for a given *visual* variable and *geometry*. This possible visual variable and the maximum length are specified here as described in Chapter 3.

_InitMapState

MapState'

subjects' = {*nr_inhabitants_2011_municipality*, *nr_inhabitants_2013_municipality*,
nr_inhabitants_2011_province, *protected_areas_2012*}
objects'={*Enschede*, *Almelo*, *Hengelo*, *Utrecht*, *Rotterdam*, *Overijssel*, *Aamsveen,*
Broekvelden, Brunssummerheide}
subjectFunction'= {*nr_inhabitants_2011_municipality* ↦ {(*Enschede*, *v*157050),
(*Almelo*, *v*72600), (*Hengelo*, *v*80770), (*Utrecht*,*v*307080), (*Rotterdam*, *v*593050)},
nr_inhabitants_2011_province ↦ {(*Overijssel*, *v*1130345)},
nr_inhabitants_2013_municipality ↦ {(*Enschede*,*v*158625), (*Almelo*, *v*72730),
(*Hengelo*, *v*80950), (*Utrecht*,*v321915*), (*Rotterdam*, *v*616295)},
protected_areas_2012↦ {(*Aamsveen*, *ONTWERP_2007*),
(*Broekvelden*, *Natura_2000_besluit_2010*),
(*Brunssummerheide*, *ONTWERP_2007*)}}
unitGeometry'= { *nr_inhabitants_2011_municipality* ↦ *multipolygon*,
nr_inhabitants_2013_municipality ↦ *multipolygon*,
nr_inhabitants_2011_province↦ *multipolygon*, *protected_areas_2012*↦ *multipolygon*}
measurementScale'= { *nr_inhabitants_2011_municipality* ↦ *absoluteRatio*,
nr_inhabitants_2013_municipality ↦ *absoluteRatio*,
nr_inhabitants_2011_province ↦ *absoluteRatio*, *protected_areas_*2012↦ *nominal*}
possibleVisuals'={*nominal*↦ {*shape*,*orientation*,*colour*}, *interval*↦ {*value*, *texture*},
ordered↦ {*value*, *texture*}, *relativeRatio*↦ {*value*, *texture*},*absoluteRatio*↦ {*size*}}
maximumLengths'={(*size*,*point*)↦ 4,(*size*,*line*)↦ 4,(*size*,*polygon*)↦ 5,(*value*,*point*)↦ 3,
(*value*,*line*)↦ 4,(*value*, *polygon*)↦ 5,(*texture*,*point*)↦ 2,(*texture*,*line*)↦ 4,
(*texture*,*polygon*)↦ 5, (*colour*,*point*)↦ 7,(*colour*,*line*)↦ 7,(*colour*, *polygon*)↦ 8,
(*orientation*,*point*)↦ 4, (*orientation*, *line*)↦ 2,(*orientation*,*polygon*)↦ 0,(*shape*,*point*)↦ 0,
(*shape*,*line*)↦ 0,(*shape*,*polygon*)↦ 0}

4.3.4. Operation

The operation *DetermineMapType* returns the best visual variable and geometry to represent a subject known to the system, taking into account some user's requirements. This operation never changes the *MapState*. And some errors might occur during its execution.

The first error that might occur is in the case of *unknown subject* to the system (*s?* ∉ *subjects*). This might occur if the user inputs a *subject* that is not known to the system. For example, density population per municipality for the year 2000, in this way the system signals known subject.

$\_DetermineMapTypeUnknownSubject$

$\Xi MapState$
$s?: SUBJECT$
$result!: REPORT$

$s? \notin subjects$
$result! = unknown\_subject$

In the case that the *subject* is known to the database (*s? ∈ subjects*). The system needs to know if the *selective* perceptual property is required and the *number of classes*. For example, for the atlas the number of inhabitants per municipality for the year 2011 map, the *selective* property is required (*yes*) and the *number of classes* is 3.

Then it decides whether or not to use the *number of classes* or the attribute length. For the example mentioned (number of inhabitants per municipality in year 2011), the variable length is 5 (*(enschede, 157050), (almelo, 72600), (hengelo, 80770), (utrecht, 307080), (rotterdam, 593050)*), and the number of classes is 3, so the *length* is 3. Because the data is absolute ratio, selectivity is a requirement and the variable length is greater than the number of classes (*if measurementScale*(*s*?) in {*absoluteRatio*, *relativeRatio*}*)∧ selective?=yes* ∧ (# (*ran* (*subjectFunction*(*s*?))) > *numberOfClasses*?)). On the other hand, for the protected areas per natura 2000 areas for the year 2012, the data is nominal, and selectivity is not a requirement, then the *length* is 3 the same as the variable length ((Aamsveen, ONTWERP_2007), (Broekvelden, Natura_2000_besluit_2010), (Brunssummerheide, ONTWERP_2007)).

The next step, the system records the geometries able to *represent* the map *unit*, since a line feature cannot be represented through an area mark (*r prefix ⟨point,multipoint,line,multiline, polygon,multipolygon⟩*; *last r=unitGeometry(s?)*; *representUnit= (ran r)* ∩ {*point, line, polygon*}). For both examples, protect areas and number of inhabitants, point, line and polygon are able to represent, since according to the initialization of the system they are polygon.

If selectivity is needed (*selective*?=*yes*), and the variable length is bigger than the maximum length of the combination geometries and visual variable able to represent the data (*if selective? then* {*v: possibleVisuals(measurementScale(s?)); g: representUnit | maximumLengths(v,g)≥length • v*} *else possibleVisuals(measurementScale(s?))* = ∅), then a message error is outputted signalizing to the user that the number of classes should be smaller (*result!= big_number_of_classes*).

```
_DetermineMapTypeTooManyClasses_________________________________
| ΞMapState
| s?: SUBJECT
| selective?: BOOLEAN
| numberOfClasses?: ℕ
| result!: REPORT
|_______________________________
| s? ∈ subjects
| ∃ length: ℕ; representUnit: ℙ GEOMETRY; r: seq GEOMETRY |
| length =
|        if (measurementScale(s?) in {absoluteRatio, relativeRatio}) ∧
|               selective?=yes ∧ (# ( ran (subjectFunction(s?))) > numberOfClasses?)
|        then numberOfClasses?
|        else # ( ran (subjectFunction(s?)))
| r prefix ⟨point,multipoint,line,multiline, polygon,multipolygon⟩
| last r=unitGeometry(s?)
| representUnit= (ran r) ∩ {point, line, polygon}
| if selective?
| then {v: possibleVisuals(measurementScale(s?)); g: representUnit |
|               maximumLengths(v,g)≥length• v}
| else possibleVisuals(measurementScale(s?)) = ∅
| result!= big_number_of_classes
|_________________________________________________________________
```

If there is at least one *visual* variable that can represent the data (*{v: possibleVisuals(measurementScale(s?)); g: representUnit |maximumLengths(v,g)≥length• v} ≠ Ø*). Then the system selects the visual variables possible to represent the data based on its measurement scale (*possibleVisuals(measurementScale(s?))*) and the maximum length (*v: possibleVisuals(measurementScale(s?)); g: representUnit | maximumLengths(v,g)≥length• v}*), if selectivity is required. For the examples, the system should generate a set with the elements colour, shape, and orientation for the protected areas and size for the number of inhabitants.

Then this set of visual variables is compared with the sequence given by the user preference. So, the sequence of the user preferences for the visual variable is required. In the case that there is no visual variable in common between the sequence and the set of *possible visual* variables to represent the subject, an error *user_amplify_visual* is returned. This indicates that there are available visual variables to represent the data, but none of them was selected by the user.

For the experimental National Dutch Web Atlas, let us suppose that the mentioned sequence of user preferences for visual variable is [size, colour, value, orientation, texture, shape] for both subjects. Then, this step would produce the sequence [colour, orientation, shape] for the protected areas and [size] for the number of inhabitants (*userVisual? ↾ if selective?=yes ∧{v: possibleVisuals(measurementScale(s?)); g: representUnit |maximumLengths(v,g)≥length• v}≠Ø then {v: possibleVisuals(measurementScale(s?)); g: representUnit |maximumLengths(v,g)≥length• v} else possibleVisuals(measurementScale(s?))*).

```
_DetermineMapTypeUserAmplifyVisual_______________________
| $\Xi MapState$
| $s?: SUBJECT$
| $selective?: BOOLEAN$
| $userVisual?: \text{seq } VISUAL$
| $numberOfClasses?: \mathbb{N}$
| $result!: REPORT$
|_____________________________
| $s? \in subjects$
| $\exists\ length: \mathbb{N}; representUnit: \mathbb{P} GEOMETRY; r: \text{seq } GEOMETRY \mid$
| $length =$
|     $\textbf{if } (measurementScale(s?) \text{ in } \{absoluteRatio, relativeRatio\}) \wedge$
|         $selective? = yes \wedge (\# (ran (subjectFunction(s?))) > numberOfClasses?)$
|     $\textbf{then } numberOfClasses?$
|     $\textbf{else } \# (ran\ subjectFunction(s?))$
| $r \text{ prefix } \langle point, multipoint, line, multiline, polygon, multipolygon \rangle$
| $last\ r = unitGeometry(s?)$
| $representUnit = (\text{ran } r) \cap \{point, line, polygon\}$
| $userVisual? \upharpoonright \textbf{if } selective? = yes \wedge \{v: possibleVisuals(measurementScale(s?)); g: representUnit \mid$
|                 $maximumLengths(v,g) \geq length \bullet v\} \neq \varnothing$
|               $\textbf{then } \{v: possibleVisuals(measurementScale(s?)); g: representUnit \mid$
|                 $maximumLengths(v,g) \geq length \bullet v\}$
|               $\textbf{else } possibleVisuals(measurementScale(s?)) = \varnothing$
| $result! = user\_amplify\_visual$
|__________________________________________________________
```

On the other hand, if there is at least one among the user preference's that can represent the data (*userVisual?* ↾ *if selective?=yes* $\wedge$*{v: possibleVisuals(measurementScale(s?)); g: representUnit |maximumLengths(v,g)≥length • v}≠Ø then {v: possibleVisuals(measurementScale(s?)); g: representUnit |maximumLengths(v,g)≥length • v} else possibleVisuals(measurementScale(s?))* ≠ *Ø*). Then, if selectivity is required, the *maximum length* of the combination *geometries* and *visual* variable best ranked are again compared with *length* (*{v: possibleVisuals(measurementScale(s?)); g: representUnit |maximumLengths(v,g)≥length • v}*). If the *maximum length* is bigger than the *length*, the *geometry* is selected. For the example of the number of inhabitants, as the *length* is 3, point, line and polygon are selected (*(size,point)↦4,(size,line)↦4,(size,polygon)↦5*). And these selected geometries are compared with the set of geometries that are able to *represent* the map *unit* and the common geometries are then compared with the user's preference for geometry and the common ones sorted in the user's preference (*userGeometry*? ↾ {*g*: *representUnit* | *maximumLengths* (*head visualSeq,g*)≥*length*}). For the atlas, this sequence of the user's preference for the number of inhabitants' subject is point and polygon in this order. As mentioned before the geometries able to *represent* this map *unit* are point, line and polygon. So the sequence of geometries resulting from the comparison with the user's preference of geometry is point and polygon in this order.

If selectivity is not required, then just the geometries able to *represent* the map *unit* are sorted according to the user's preference for geometries (*userGeometry*?↾ *representUnit*). In the example of the protected areas, the user's preference sequence for geometry is polygon, point and line. And so the geometry sequence resulting of this comparison is polygon, point and line.

If there is no geometry in common among user's preference for geometry, the set of geometries able to *represent* the map *unit* and geometries able to make the mark visible, the system signals the *user_amplify_geometry*.

```
_DetermineMapTypeUserAmplifyGeometry________________________
$\Xi MapState$
$s?: SUBJECT$
$selective?: BOOLEAN$
$userVisual?: \mathrm{seq}\ VISUAL$
$userGeometry?: seq\ GEOMETRY$
$numberOfClasses?: \mathbb{N}$
$result!: REPORT$
_____________________________
$s? \in subjects$
$\exists\ length: \mathbb{N}; representUnit: \mathbb{P}\ GEOMETRY; r: \mathrm{seq}\ GEOMETRY \mid$
$length =$
    $\textbf{if}\ (measurementScale(s?)\ \mathrm{in}\ \{absoluteRatio, relativeRatio\}) \wedge$
        $selective?=yes \wedge (\#(ran\ (subjectFunction(s?))) > numberOfClasses?)$
    $\textbf{then}\ numberOfClasses?$
    $\textbf{else}\ \#\ (ran\ subjectFunction(s?))$
$r\ \mathrm{prefix}\ \langle point, multipoint, line, multiline, polygon, multipolygon\rangle$
$last\ r = unitGeometry(s?)$
$representUnit = (\mathrm{ran}\ r) \cap \{point, line, polygon\}$
$userVisual? \upharpoonright\ \textbf{if}\ selective?=yes \wedge \{v: possibleVisuals(measurementScale(s?)); g: representUnit \mid$
                $maximumLengths(v,g) \geq length \bullet v\} \neq \emptyset$
            $\textbf{then}\ \{v: possibleVisuals(measurementScale(s?)); g: representUnit \mid$
                $maximumLengths(v,g) \geq length \bullet v\}$
            $\textbf{else}\ possibleVisuals(measurementScale(s?)) \neq \emptyset$
$\textbf{if}\ selective?=yes$
$\textbf{then}\ userGeometry? \upharpoonright \{g: representUnit \mid maximumLengths\ (head\ visualSeq, g) \geq length\}$
$\textbf{else}\ userGeometry? \upharpoonright representUnit = \emptyset$
$result! = user\_amplify\_geometry$
_____________________________________________________________
```

In the case, there is at least one geometry in common, the first geometry in the sequence is retuned as the *geometry* to make the mark visible (*geometry!= head (if selective?=yes then userGeometry?* ↾ {*g: representUnit* | *maximumLengths* (*head visualSeq,g*) ≥ *length*} *else userGeometry?*↾*representUnit* ≠ ∅)). Also the first *visual* variable in the sequence generated from the comparison with the user's prefenece is outputted (*visual!* = *head* (*userVisual?* ↾ *if selective?=yes* ∧{*v: possibleVisuals(measurementScale(s?)); g: representUnit* | *maximumLengths(v,g)*≥*length* • *v*}≠∅ *then* {*v: possibleVisuals(measurementScale(s?)); g: representUnit*

| maximumLengths(v,g)≥length• v} else possibleVisuals(measurementScale(s?)) ≠ Ø)). The combination of this two outputs is the map type.

```
_DetermineMapTypeOk_______________________________________________
| ΞMapState
| s?: SUBJECT
| selective?: BOOLEAN
| userVisual?: seq VISUAL
| userGeometry?: seq GEOMETRY
| numberOfClasses?: ℕ
| visual!: VISUAL
| geometry!: GEOMETRY
| result!: REPORT
|______________________________
| s? ∈ subjects
| ∃ length: ℕ; representUnit: ℙ GEOMETRY; r: seq GEOMETRY |
| length =
|       if (measurementScale(s?) in {absoluteRatio, relativeRatio})∧
|             selective?=yes ∧ (# ( ran (subjectFunction(s?))) > numberOfClasses?)
|       then numberOfClasses?
|       else # (ran subjectFunction(s?))
| r prefix ⟨point,multipoint,line,multiline, polygon,multipolygon⟩
| last r=unitGeometry(s?)
| representUnit= (ran r) ∩ {point, line, polygon}
| visual! = head (userVisual? ↾ if selective?=yes ∧{v: possibleVisuals(measurementScale(s?));
|             g: representUnit | maximumLengths(v,g)≥length• v}≠∅
|       then {v: possibleVisuals(measurementScale(s?)); g: representUnit |
|             maximumLengths(v,g)≥length• v}
|       else possibleVisuals(measurementScale(s?)) ≠ ∅)
| geometry!= head (if selective?=yes
|       then {g: representUnit | maximumLengths (head visualSeq,g)≥length}
|       else userGeometry?↾representUnit ≠ ∅)
| result!=ok
|__________________________________________________________________
```

In the case of the protected areas described as an example through this specification the combination is colour and area and for the number of inhabitants is size and point.

Thus, the specification of the operation of determining the visual variable and geometry that is the most suitable to represent a subject is given in the schema below.

$$\begin{aligned} \mathit{DetermineMapType} \triangleq\ & \mathit{DetermineMapTypeUnknownSubject} \\ & \vee\ \mathit{DetermineMapTypeTooManyClasses} \\ & \vee\ \mathit{DetermineMapTypeUserAmplifyVisual} \\ & \vee\ \mathit{DetermineMapTypeUserAmplifyGeometry} \\ & \vee\ \mathit{DetermineMapTypeOk} \end{aligned}$$

As explained before, the system specified here only includes partial data included through the initial state of the system. A complete system would have all the data for the state variables *subjects*, *objects*, *subjectFunction*, *unitGeometry*, and *measurementScale* included.

For example, the density population year 2011 per province map from the atlas, this subject needs to be included in the initial state. So the density_population_2011_province would also be in *subject'*.

The provinces of Netherlands should be input in the object state, except the province of Overijssel that was already included as partial data. Thus, *object'* would have also Gelderland, Noord-Holland, Flevoland, Groningen, Friesland, Zeeland, Noord-Brabant, Drente, Utrecht, Zuid-Holland, and Limburg.

The attributes of the population density year 2011 for all the provinces should be entered in the *subjectFunction'* with its respective subject and province. So *subjectFunction'* would have the following elements besides the ones that it already has: density_population_2011_province ↦{(Groningen, 576665), (Flevoland, 387880), (Friesland, 646320), (Drente, 490980), (Overijssel, 1130345), (Gelderland, 1998940), (Utrecht, 1220900), (Noord-Holland, 2669090), (Zuid-Holland, 3505600), (Zeeland, 381410), (Noord-Brabant, 2444155), (Limburg, 1122705)}.

The *unitGeometry* should also be updated with the geometry of the map unit to the subject. Thus the *unitGeometry'* should has in addition that multipolygon is the map unit for the subject density population year 2011 per province.

And the final update to the initial state of the map, for this example should be adding the density_population_2011_province ↦ relativeRatio to the variable *measurementScale'*, which is the measurement scale of the subject density population 2011 per province.

With this enters in the initial state of the system, it is possible to determine a map type to represent the population density year 2011 per province subject using the operation specified above. For this the subject, the necessity for selectivity property, the number of classes and the user's preferences for visual variable in order, and user's preference for geometry in order are necessary.

For this example and taking into account the population density map in the national web atlas, the density_population_2011_province as a subject, the selective as yes, number of classes as 5, and suppose the user's preference for visual variable as the sequence [value, size, colour, texture, orientation, shape] and the user's preference for geometry in sequence as [polygon, point] need to be input in the operation.

As this subject is now a known subject to the system (*s* ∈ *subjects*), there is not the error *unkown_subject*. In the next step the operation defines the *length*. As the measurement scale is relative ratio (*density_population_2011_province ↦ relativeRatio*), selectivity is necessary and the variable length (12) is bigger than the number of classes (5), length is the number of classes, which is 5 (*length = if (measurementScale(s?) in {absoluteRatio, relativeRatio}) ∧ selective?=yes ∧ (# (ran (subjectFunction(s?))) > numberOfClasses?) then numberOfClasses? else # (ran (subjectFunction(s?)))*).

Then the system generates *r*, which is a sequence of geometries being the last geometry the same geometry of the map unit. In the population density example, *r* is the sequence [point, multipoint, line, multiline, polygon, multipolygon]. The elements of this sequence are compared with the elements of the set with elements point, line and polygon and the common elements are recorded in *representUnit*, in our example point, line and polygon. Those are the geometries of the mark able to represent the map unit, not taking into account the selectivity property.

As the selectivity is necessary, the system test gets each visual variable possible to represent the data, based on it measurement scale (*possibleVisuals*(*measurementScale*(*s*?))), for the example value and texture, and the geometries in *representUnit*, and for each combination of these visual variable and geometries, the system tests its maximum length and the *length* and return the visual variables able to represent the data. For the example the visual variables able to represent the data are value and texture, because for both of them, the combination with polygon has maximum length 5. And as this is not an empty set, selectivity is possible.

Then the user's preference for visual variable are filtered by the value and texture, which leads to the sequence [value, texture]. As this is not an empty sequence, the error *user_amplify_visual* is not returned. Following the first visual variable in the sequence generated (value) is combined with the geometries able to represent the data and the maximum length of each combination is compared with the *length*. The user's preference for geometry is filtered out by the geometries of the combination which are bigger than *length*. In the example, [polygon, point] is filtered out by polygon, because the combination value and point is 3 and value and line is 4. Thus the resulting geometry is polygon. As this is not an empty sequence the error *user_amplify_geometry* is not returned. In this way, as value and polygon are the first visual variable and geometry in each of the last generated sequence, the map type is a combination of value and polygon. Which is the choropleth map used to represent the population density map in the experimental National Dutch Web Atlas.

The specification above shows that it is possible to formalize the process to determine a map type based on data characteristics, taking into account some user's requirements.

While specifying this process, the user's requirement for visual variable and geometry is required every time that this operation is performed. As the user (the one who is looking for information about Netherlands) could select the visual variable and geometry. It could be useful to the atlas case, that these two sequences are knowns to the system.

As stated by Khwaja & Urban (2010) the languages in the set theory group are not ease to use, and going through the details of the set theory required by Z, requires training. However, as also pointed by them, the abstract structure of the languages in this group makes it possible to be used in a varied of domains.

In the operation described, the Z language allowed the error messages which are necessary in the specification of the process. However, Z was selected also because it is able to describe different states of the system. For example, one possible operation could be adding a subject or a new visual variable not known to the system, which could have changed the *MapState*.

We have formalized the process of choosing a map type. And for this the specification of the measurements was done using a non-numerical type, named MEASUREMENT. However, if someone wants to formalize the process of creating a map, then a classification method might be necessary. Thus measurement should be defined using a numerical type, to allow mathematical operations.

The formal specification of the process of selecting a map type based on data characteristics was done for discrete data, to display one information at a time and for an "overview" of the user's task. In case where there is more than one subject the user should use Mackinlay's (1986) principle of importance ordering and give as input a *sequence* of subjects. However, this would require more modifications on this specification, since some visual variables are dissociative.

5. CONCLUSION

Formal specification language uses mathematical notation to precisely specify a system. The use of formal specification language leads to unambiguous description and a possible automation of the process.

Thus, this research was conducted in order to prove if it is possible to formally specify the process of choosing a map type, based on the data characteristics.

For achieving the formalization of the process we performed a literature review that enabled us to define the step of the process of generating a map. Those steps are: Invariant, Component, Algebra, Scale, Statistics, Summary, Classification, Topography, Aesthetic and Geometry. To be able to formally describe this process, first an informal description was executed in Chapter 3. This informal description of the process is for creating a statistical map from discrete data and one variable.

We also reasoned about the existing types of formal language specification, based on the work of different authors, which lead to the selection of the state-based group of formal languages. The languages in this group are based on pre- and post-conditions and can describe different states of the system. Among the languages in this group we selected the Z specification language for the specification of the process. Z language has a lot of manuals, case studies and is an ISO standard. However, the Z language is not easy to use, and requires training.

The knowledge obtained from the literature review allowed us to determine that for selecting the map type based on the data characteristics the selective perceptual property, the number of classes, the length of a variable, the measurement scale, the geometry of the geographical component the visual variable, the geometry, and the user's preference among visual variable and geometry should be specified. And more important the dependencies among all those items, as explained in Chapter 3 and 4.

Finally, the research specified as a proof of concept the process of selecting a map type based on the data characteristics for the experimental National Dutch Web Atlas, in Chapter 4. The atlas has as characteristic that is composed of maps with one statistical component.

Below the research question and their answers:

a) What is formal specification language?
Formal specification language uses mathematical notation to precisely specify a system. Formal specification leads to an unambiguous description.

b) What are the types of formal specification language?
There is not an agreement for a formal specification language types, Woodcock & Loomes divide in model-oriented (state-based), algebraic, process algebra and modal logics; Lamsweerde as history-based, state-based, trasition-based, functional-based and operational and Khwaja & Urban in algebraic, axiomatic, temporal logic, process algebra, set theory, finite state machine specification and functional. In this work, we usef the types of the latter authors.

c) Can formal specification language describe the process of choosing a map type for a specific data set, taking into account certain user requirements?
Yes, it is possible to specify the process of choosing a map type for a specific data set, taking into account certain users requirements, as we did in Chapter 4. In the literature review, we found the set theory group which is a type of languages able to specify any domain; and the state-based types describes changes from state to state and are based on pre- and post-conditions.

d) Is there already a formal specification language that can be used to do this, and if so, which language is this?

In the state-based group and in the set theory group there is the Z language. And using Z we were able to formally specify the process of selecting a map type based on the data characteristics for the expertimental National Dutch Web Atlas.

e) Which are properties of the maps, data sets and users that should be specified, and how can they be specified in the chosen language?
For the experimental National Dutch Web Atlas the properties of maps, data sets and users that should be specified for describing the process of selecting a map type based on the data characterisitics and it formal specification, can be found in Chapter 4. These characteristics for maps are the combination of visual variable and geometry. For data sets we should specify the *variable length*, its *measurement scale*, and the *geometry* of the geographical component. For the user specify his/her preference for *selective* perceptual property, *visual variable*, *geometry* to make the mark visible and the *number of classes*. The specification of number of classes is specific for our use case.

f) Which are the (transformation) processes from data to maps that should be specified, and how can they be specified in the chosen language?
The processes that should be specified are Component, Algebra, Scale, Summary, Aesthetic and Geometry and in Chapter 4 presents the specification.

By answering this research questions, we have reached our research objectives.

This research informally describes the generation of maps only for one variable and discrete data. In the literature it was possible to find an ordering of the visual variables by effectiveness in representing a quantitative attribute for graphs in general. Therefore, the next step before going to the description of the process of map generation representing more than one variable, should be a study to rank the visual variables for each measurement scale, but now specially for maps.

We also formally specified the process of generating a map type based on the data characteristics. The next step is the formal specification of the process creation of the map informally specified in Chapter 3.

The next step towards the automation of the process of creating a map should be a research about the possibility of automatic generating the algorithm from the formal specification of the process of map generation.

LIST OF REFERENCES

Andrienko, G. L., & Andrienko, N. V. (1999). Interactive maps for visual data exploration. *International Journal of Geographical Information Science*, *13*(4), 355–374. http://doi.org/10.1080/136588199241247

Balley, S., Baella, B., Christophe, S., Pla, M., Regnauld, N., & Stoter, J. (2014). Map Specifications and User Requirements. In D. Burghardt, W. Mackaness, & C. Duchêne (Eds.), *Abstracting Geographic Information in a Data Rich World* (pp. 17–52). Cham, Switzerland: Springer International Publishing. http://doi.org/10.1007/978-3-319-00203-3_2

Basaraner, M. (2016). Revisiting cartography: towards identifying and developing a modern and comprehensive framework. *Geocarto International*, *31*(1), 71–91. http://doi.org/10.1080/10106049.2015.1041560

Bertin, J. (1983). *Semiology of graphics: diagrams, networks, maps*. (W. J. Berg, trans.). Madison, Wisconsin: The University of Wisconsin Press.

Community Z Tools Project. (2016). CZT: Community Z Tools. Retrieved February 7, 2017, from http://czt.sourceforge.net/

Diller, A. (1990). *Z an introduction to formal methods* (1st ed.). Chichester, England: John Wiley & Sons.

Global Spatial Data Infrastructure. (2004). *Developing Spatial Data Infrastructures: The SDI Cookbook*. (Douglas D. Nebert, Ed.). Retrieved from http://gsdiassociation.org/index.php/publications/sdi-cookbooks.html

Khwaja, A. A., & Urban, J. E. (2010). A property based specification formalism classification. *Journal of Systems and Software*, *83*(11), 2344–2362. http://doi.org/10.1016/j.jss.2010.07.031

Knippers, R., & Mank, T. (2015). *Visualization of thematic data*. Retrieved from https://blackboard.utwente.nl/bbcswebdav/pid-918532-dt-content-rid-1940775_2/courses/M16-GFM-103/Exercises/Cartography/2. Choropleth_Piegraph_2015_exercise2.pdf

Köbben, B. (2013). Towards a National Atlas of the Netherlands as part of the National Spatial Data Infrastructure. *The Cartographic Journal*, *50*(3), 225–231. http://doi.org/10.1179/1743277413Y.0000000056

Kraak, M.-J., & Ormeling, F. (2010). *Cartography: Visualization of Geospatial Data* (3rd ed.). Edinburgh, Scotland: Person Education Limited.

Lamsweerde, A. Van. (2000). Formal specification: a roadmap. In *Proceedings of the conference on The future of Software engineering - ICSE '00* (pp. 147–159). New York, USA. http://doi.org/10.1145/336512.336546

Mackinlay, J. (1986). Automating the design of graphical presentations of relational information. *ACM Transactions on Graphics*, *5*(2), 110–141. http://doi.org/10.1145/22949.22950

Nelson, M. A. V, Alencar, P. S. C., & Cowan, D. D. (2001). An approach to formal specification and verification of map-centered applications. *Environmental Modelling & Software*, *16*(5), 459–465. http://doi.org/10.1016/S1364-8152(01)00017-2

Open Geospatial Consortium Inc. (2006). *Open Geospatial Consortium Inc . OpenGIS ® Web Map Server Implementation Specification*. (J. de la Beaujardiere, Ed.) (1.3.0 ed.). Open Geospatial Consortium Inc.

Retrieved from http://www.opengeospatial.org/ogc/Document

Province Overijssel. (2016). Data Overijssel. Retrieved October 13, 2016, from https://overijssel.databank.nl/jive/jivereportcontents.ashx?report=home

Reimer, A. (2015). *Cartographic Modelling for Automated Map Generation*. Eindhoven University of Technology.

Shneiderman, B. (1996). The eyes have it: a task by data type taxonomy for information visualizations. *Proceedings 1996 IEEE Symposium on Visual Languages*, 336–343. http://doi.org/10.1109/VL.1996.545307

Sivey, J. M. (1998). *The Z Notation: A Reference Manual* (2nd ed.). London, England: Oriel College, Oxford.

Stevens, S. S. (1946). On the Theory of Scales of Measurement, *103*(2684), 677–680.

Stichting Wetenschappelijke Atlas Nederland. (2013). De Nationale Atlas Van Nederland. Retrieved February 6, 2017, from http://www.nationaleatlas.nl/

Wilkinson, L. (2005). *The Grammar of Graphics* (2nd ed.). New York, USA: Springer-Verlag. http://doi.org/10.1007/0-387-28695-0

Wilkinson, L. (2010). The grammar of graphics. *Wiley Interdisciplinary Reviews: Computational Statistics*, *2*(6), 673–677. http://doi.org/10.1002/wics.118

Woodcock, J., & Loomes, M. (1988). *Software Engineering Mathematics* (1st ed.). London, England: Pitman Publishing.